HEALTHIFY DOCUMENTATION

SADDIQ ABDUL QADIR

Saddiq Abdul Qadir
2024

Abstract

E-commerce has emerged as a thriving force in an internet-dominated society, outpacing the expansion of traditional brick-and-mortar enterprises. This initiative sets out to address a critical issue in the world of health and wellness products: accessibility. In Trinidad, the availability of locally sourced health and wellness products is restricted, and people's busy lives frequently prevent them from doing lengthy searches for these essential products.

The primary purpose of the project is to develop an innovative e-commerce web platform concentrating on health and wellness products. This application will provide an extensive variety of health supplements, organic products, and specialized wellness items, redefining access for Caribbean citizens. However, the ultimate objective of this initiative is to enhance the overall customer experience by utilizing cutting-edge components like gamification and artificial intelligence, going beyond basic functionality.

The key objectives of this project concentrate on the creation of a health and wellness e-commerce platform that meets the needs and expectations of users. The project starts by conducting thorough background research, delving into the e-commerce industry to study market trends and contemporary norms. Following that, a thorough requirements analysis phase outlines the project's scope, objectives, and functional/non-functional requirements, as well as personas and use cases. Database and application development is the centre of design, with an emphasis on data organization, user experience, and interactive prototypes. Implementing software brings ideas to life, including features like recommender systems, gamification, and responsive design. Extensive testing ensures that functionality, performance, and issue resolution meet user expectations. Software evaluation validates project effectiveness by assessing real-world efficiency, strengths, and opportunities for improvement. The conclusion highlights the project's path, emphasizing learning through experience, accomplishments, and important lessons learned.

This project successfully tackles a major issue in Trinidad and demonstrates the capacity of technology to improve accessibility, convenience, and trust within the health and wellness industry by developing this e-commerce platform for health and wellness. Artificial intelligence, user-centric design, and gamification techniques come together to provide a dynamic and entertaining shopping experience that meets the needs of a wide range of customers.

Keywords

E-commerce. Artificial Intelligence (AI). Digital. Entity Relationship Diagram (ERD). Data Dictionary. Agile. Responsive Design. User Experience (UX). User Interface (UI). Version Control. Chatbot. JavaScript. Analytics. Data Visualization. Gamification

Table of Contents

Abstract 1

Keywords 2

Introduction 7

Introduction 7

The Subject 7

Project Aim 9

Project Objectives 9

Summary of Chapters 10

Project Management 12

Introduction 12

Importance of Project Management 12

Work Breakdown Structure 13

Gantt Chart 14

Background Research 17

Introduction 17

E-commerce Systems 17

Websites Reviewed 23

Ecommerce Functionalities 26

Artificial Intelligence 26

Chatbot 27

Gamification 33

Recommender System 36

Responsive Web Design 40

Project Related Choices 43

Software Development Tools 43

Version Control System 44

Software Development Methodology ... 45
Summary ... 46
Recommendations ... 46
Requirements Analysis ... 48
Introduction ... 48
The Process of Producing the Requirements ... 48
Personas ... 49
The Requirements ... 52
Functional Requirements ... 53
Non-Functional Requirements ... 55
UML Diagrams ... 56
Information Architecture ... 61
Activity Diagram ... 63
Design ... 66
Introduction ... 66
Database Design (Back End Design) ... 66
Entity Relationship Diagram (ERD) ... 69
Normalization Process ... 69
Logical Model ... 71
Data Dictionary ... 72
Application Design (Front End Design) ... 73
Nielsen’s Heuristics ... 73
Sketches ... 75
Wireframes ... 75
Interactive Prototype ... 76
Implementation ... 79
Introduction ... 79
Database Implementation ... 79

Connection between Database and Front End 80
User Interface Implementation 81
Core Features Implemented 83
FR1 – Registration 83
FR2 – Login 90
FR3 – User Profile Management 94
FR4 – Product Catalogue 99
FR5 – Search Functionality 102
FR6 – Product Filtering 106
FR7 – Product Detail Page 108
FR8 – Add to Cart 111
FR9 – Shopping Cart Management 112
FR10 – Checkout Process 119
FR11 – Payment Integration 122
FR12 – Order Confirmation 124
FR13 – Order History 126
FR14 – User Reviews and Ratings 129
FR15 – Account Management 133
FR16 – Admin Dashboard 136
Advanced Features Implemented 142
FR17 – Chatbot Integration 142
FR18 – Recommendation Engine 145
FR19 – Gamification Features 146
FR20 – Advanced Analytics and Data Visualization 153
FR21 – Responsive Web Design 163
Security Consideration 164
Error Messages 168
Version Control 169

Features Not Implemented 170
Additional Features Implemented 170
Testing 173
Introduction 173
Test Plan 173
Functionality Testing 174
Non-Functionality Testing 175
Defects Log 176
Analysis of Test Results 177
Evaluation 178
Introduction 178
Method of Evaluation 178
The Heuristics 178
Summary of Findings 180
Analysis of Results 181
References 182
Appendices 191
Appendix A: Requirements Analysis – Functional and Non-Functional Requirements 191
Appendix B: Data Dictionary 199
Appendix C: Project Progress Reports 206
Appendix D: Testing of Functional Requirements 220
Appendix E: Poster 223
Appendix F: Sketches 224
Appendix G: Stripe API 225

Introduction

Introduction

The initial section delves into the project's origins, articulating its purpose and underlying motivations. It defines the project's goal and objectives, providing readers with a foundational understanding and serving as an intro to the subsequent chapters, guiding the narrative flow of the report.

The Subject

In this internet era, e-commerce is booming, far exceeding the development of brick-and-mortar enterprises. Many brick-and-mortar businesses are turning to an online or e-commerce-driven alternative. People in the developed world, as well as a rising number of people in the developing world, are increasingly using e-commerce websites to make purchases on a regular basis (Ullah, et al., 2016).

Locally available health and wellness products are limited in Trinidad and there are few physical storefronts. Furthermore, locating health and wellness products in Trinidad, especially those that are niche or specialised, could prove challenging. In Trinidad, many people lead busy lives and may not have the time to physically visit many stores in search of the health and wellness products they require. This project aims to develop a health and wellness e-commerce web application that provides health supplements, organic products, and specialised wellness products. It will utilize cutting-edge features such as the application of artificial intelligence and gamification techniques to encourage user engagement and participation and enhance the user experience.

The development of this e-commerce application would be supported and enhanced by:

- Synthesizing and analysing existing related websites to gain a deeper understanding of the market landscape, potential target segments, etc.
- Academic journals, industry publications, and reputable online sources.
- Academic articles addressing the effects of e-commerce on the health and wellness sector, online purchasing habits, marketing tactics, and developing modern technology.
- Assessing industry publications and online sources in search of innovative concepts, best practices, and case studies in the health and wellness e-commerce space.

Any gamified domain that captures how a user feels whilst interacting with the system must initially prioritize the user experience. Gamification is increasingly being utilized in e-commerce as a design strategy for enhancing various behavioural outcomes. According to studies, gamification enhances the e-commerce user experience. Users are satisfied with the gamified user experience, which increases their probability of purchasing. Users will become fond of a particular brand and become dedicated to the products. Additionally, they will aid in business expansion by spreading positive word of mouth (Mominzada, et al., 2022).

Users can be persuaded to use a website using gamification techniques like rewards, challenges, and badges subsequently leading to higher website engagement and activity. Artificial intelligence in online retail apps can boost user satisfaction and engagement by providing personalised product recommendations, chatbot support, and seamless navigation.

Rewards and Points Systems are gamification techniques used in e-commerce systems that allow users to earn points for actions such as purchases or referrals, which may then be redeemed for discounts or perks, thereby encouraging user engagement.

Chatbots and Virtual Assistants are two instances of artificial intelligence in the e-commerce market. Chatbots and virtual assistants powered by AI provide real-time customer care, offering rapid responses and assisting with inquiries, improving the customer experience, and decreasing the need for human intervention.

The establishment of a health and wellness e-commerce website in Trinidad would solve significant issues that affect individuals seeking access to a diverse range of products. Limited availability of comprehensive health and wellness options locally can impede individuals’ ability to access specific products. The website would bridge this gap, offering an extensive range of products that are challenging to attain in physical stores locally. Furthermore, acquiring specific health and wellness products in Trinidad would be considerably less complicated, reducing the need for prolonged searches or depending on irregular local availability. In addition, the health and wellness market frequently lacks trustworthy information and sources, triggering client scepticism. The website would provide extensive product descriptions, usage instructions, and credible customer testimonials, thereby resulting in trust and allowing customers to make rational decisions. Ultimately, the e-commerce platform would improve accessibility, convenience, and trust, resulting in increased customer satisfaction and overall health in Trinidad.

Gamification techniques, such as prizes, badges, and interactive challenges, will be incorporated into the website to stimulate user participation and engagement. Likewise, artificial intelligence will be leveraged to enhance the user experience by providing personalised product recommendations, chatbot support, and seamless navigation. The features mentioned above will be effortlessly incorporated into the online commerce application, offering customers a dynamic and immersive purchasing experience.

Project Aim

This project aims to utilize the SDLC framework to analyse, design, implement, and evaluate an online e-commerce store for health and wellness products. The project will leverage gamification techniques to increase user engagement and participation, while artificial intelligence will be leveraged to improve the overall user experience.

Project Objectives

Project objectives are focused and goal-oriented, accompanied by sub-objectives encompassing tasks, risk and fund management, and monitoring processes. Achieving these objectives is crucial for attaining the goal while ensuring effective control and monitoring to mitigate any adverse factors (Project-Management. pm, 2017).

The project's objectives are as follows:

1. Background Research: A variety of resources, including industry reports, academic journals, market studies, and books on health and wellness will be utilized to conduct background research for this project. These resources will offer valuable insights regarding industry best practices, market trends, customer behaviour, and competitor analysis. The information obtained from these sources will be leveraged to help design, execute, and differentiate the e-commerce platform for health and wellness products.
2. Requirements Analysis: The requirements analysis for this project will consist of establishing and documenting the specific demands, features, and objectives of the online e-commerce store for health and wellness products, as well as linking them to user expectations and corporate goals.
3. Design: The project's general design approach will include the development of a prototype alongside an entity-relationship diagram that will be utilized as a blueprint to connect backend and frontend components. These important components are necessary to ensure that the application satisfies all requirements and functions appropriately.
4. Implementation: To ensure that the project achieves the required functionality, appropriate programming languages and technologies like HTML, CSS, JavaScript, and PHP would be used to ensure that the application's features are implemented.
5. Testing: To test the application by developing a detailed test strategy and including relevant test data to ensure that it functions as expected.

6. Evaluation: To evaluate the program's software potential, assess its usability using heuristic evaluation, and adjust based on the results.

Summary of Chapters

The following is a summary of the chapters of the final report:

Table 1: Displaying the summary of chapters.

Chapter	Description
Project Management	The section on project management will emphasize the critical importance of structured project oversight in ensuring success. It will look at specific project aspects that have a direct connection to project management practices. The role of tools such as the Gantt Chart and the Work Breakdown Structure in effective project planning and execution will be central to the discussion. The chapter will also emphasize the transformative impact of strong project management on project management, from providing a clear roadmap to ensuring efficient time management and task prioritization.
Background Research	The background research section delves into a thorough examination of the e-commerce landscape, examining current market trends, advances in technology, and common functional requirements. The chapter aims to gain insights into the latest methodologies and tools shaping the industry by studying existing e-commerce platforms. This extensive research serves as a foundation providing the project with the knowledge required to produce a cutting-edge artifact following modern e-commerce standards.
Requirements Analysis	The first phase of the software development life cycle, requirement analysis, focuses on meticulous documentation and comprehension of client needs. Its essence can be summed up by Stephen Covey's saying, "Begin with the end in mind." This chapter delves into defining the project's scope, objectives, and functional and non-functional requirements. To ensure clarity in subsequent stages, personas are created to represent typical users, with use cases detailing interactions and information architecture displayed through sitemaps and activity diagrams.
Design	The design chapter, often regarded as the most imaginative phase, bridges the gap between specifications and actionable strategy. This stage, in the words of Steve Jobs, "design is not just what it looks like... design is how it works," goes beyond aesthetics. The database (back end) and application (front end) are both intricately designed. The focus of database design is on data structuring, normalization, and entity-relationship mapping. In contrast, application design revolves around user experience, which is aided by Nielsen's Heuristics, wireframes, and interactive prototypes.

Software Implementation	The implementation chapter documents the transformation of the Software Design Document into executable code. It emphasizes the importance of harmoniously marrying the back end and front end when transforming plans into reality. The recommender system, gamification, and responsive design all come to life. There is a review of how tools and languages such as HTML, CSS, JavaScript, PHP, and others play a critical role in realizing the vision. In addition, the incorporation of security measures and version control systems ensures that development is streamlined.
Testing	Following the coding process, this critical phase is dedicated to evaluating the software's efficacy. Testing compares the software to its expected outcomes to ensure that its functionality and performance meet the desired standards. The chapter describes the systematic approach to testing, from planning to evaluating functional and non-functional aspects and tracking and resolving issues. The ultimate goal is to strengthen the software's robustness, ensuring that it meets user expectations and project objectives.
Software Evaluation	Software evaluation is an important step that goes beyond simply determining whether or not the application works, delving into its overall performance in real-world settings. The implementation phase focuses on developing the application's components, whereas the evaluation phase evaluates its efficiency and functionality. This phase aims to identify the application's strengths, areas for improvement, and alignment with intended standards by using structured methodologies and gathering feedback from a variety of parties. The project supervisor and the university coordinator will lead the evaluation in this context, ensuring a thorough review.
Conclusion	The conclusive section summarizes the entire project's journey, shedding light on the intricate processes, technical theses, and artefact development. As we progress through this story, the student's experiential learning and growth come to the fore. An open self-reflection reveals obstacles overcome, victories attained, and lessons learned for future endeavours. Finally, a critical analysis of the final software artefact is presented, emphasizing its strengths and areas for potential improvement.

Project Management

Introduction

Our endeavour requires project management because it provides systematic planning, coordination, and control. It ensures that resources are optimized, risks are minimized, schedules are met, and project objectives are met. This systematic approach improves project efficiency, quality, and overall success. In this section, we underline the critical role of project management in guaranteeing the project's success. It discusses the importance of project management, including project evaluation and future plans. The section goes into detail on developing a Work Breakdown Structure (WBS) to divide the project into manageable pieces. It also offers Gantt charts for visualizing the timing and sequence of future project operations. This structured strategy promotes effective project coordination, tracking, and achievement of project objectives.

Importance of Project Management

Project management (PM) stands as a pivotal organizational function, offering a structured approach to effectively plan, execute, and control projects. It ensures that projects are completed within a defined scope, time, and budget, fostering efficiency and resource optimization. PM methodologies, encompassing tools like Gantt charts and Work Breakdown Structures (WBS), provide clear project visualization and task allocation. Additionally, PM enables risk identification and mitigation, enhancing project resilience. Ultimately, PM plays a crucial role in aligning projects with organizational goals, ensuring successful outcomes and enhanced productivity (Mir & Pinnington, 2014). Project management is very important in the context of the Undergraduate Project. Due to the complexity of our Undergraduate Project, effective project management is critical. The presence of several project deliverables increases the probability of missing deadlines, resulting in penalties. With diverse skill requirements for distinct project aspects and the need to grasp critical technologies, all within a 12-week timeframe, effective project management becomes the linchpin for methodical task allocation, timely execution, and optimal resource utilization, ensuring our project's success. The Work Breakdown Structure (WBS) will be strategically used in this project to deconstruct complex activities into manageable sub-tasks, providing orderly task management. The Gantt chart will facilitate timeline visualization, aiding task scheduling, resource allocation, and progress monitoring throughout the 12-week project duration.

Work Breakdown Structure

According to Schwalbe (2013), “A work breakdown structure (WBS) is a deliverable-oriented grouping of the work involved in a project that defines its total scope”. A WBS divides projects into stages, deliverables, and work packages, assisting with project planning. It has an impact on project processes such as scheduling, risk analysis, and quality compliance (Pasaribu, et al., 2019). A WBS is essential because it divides large projects into smaller tasks, thus ensuring a defined scope and assisting with project planning. It outlines deliverables, improves stakeholder communication, reduces the risk of omitting important deliverables, and provides a plan for effective project control (Hans, 2021).

Guided by the Project Handbook, the project's scope has been systematically divided into six key deliverables, ensuring alignment with project requirements:

- Reflective Report
- Software Artefact
- Final Report
- Poster
- Project Viva
- Project Supervisor Meeting

Breaking down a project into distinct key areas is a fundamental approach that supports the successful organization of tasks. The 6 Undergraduate Deliverables listed above serve as the key divisions for this project, marking critical milestones. Notably, four of these major areas have been further subdivided for further depth.

The Reflective Report is divided into two tasks: Self Reflection and a List of Final Report Chapters. Refining the Final Report entails finishing individual chapter write-ups and ending with a comprehensive list of chapters. Similarly, the Software Artefact has been deconstructed into core features and advanced features development, each with its list of features. This well-organized framework, as shown in Figure 1, streamlines the project execution, ensuring that each activity is completed efficiently and successfully.

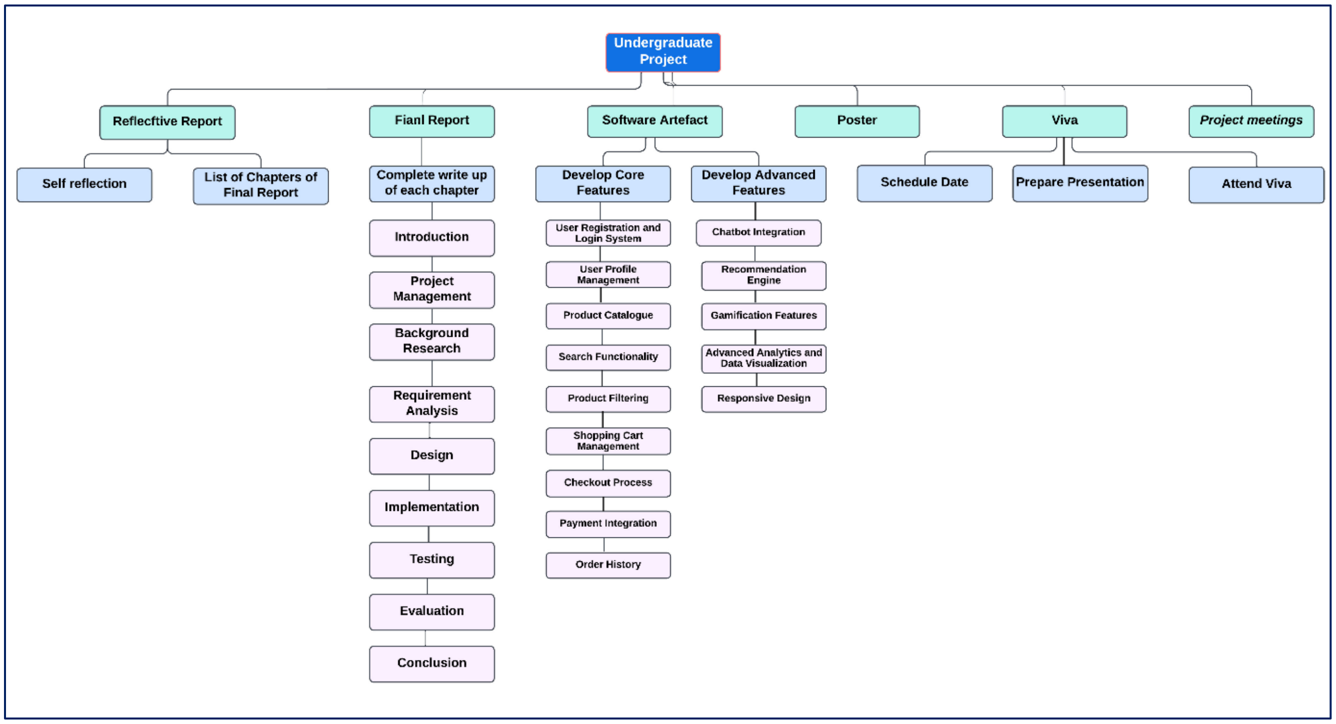


Figure 1: Displaying the Work Break Down Structure.

Gantt Chart

A Gantt chart is a visual representation of the progress of tasks or activities across time. It presents tasks as horizontal bars along a timeline, with start and end dates indicated. Gantt charts are very useful in project management since they display task dependencies, durations, and project scheduling (Scully-Allison & Isaacs, 2021). In the realm of project management, Gantt charts demonstrated remarkable durability, withstanding the test of time despite changing management trends. Gantt charts continued to be useful even when more sophisticated scheduling methods like PERT and CPM appeared. These diagrams provide an easy-to-understand method for visualizing project activities, coordinating tasks, keeping track of progress, and improving team communication. Gantt charts have become generally important for effectively managing projects across a range of contexts and complexities as a result of their historical significance and effectiveness (Geraldi & Lechter, 2012).

The Work Breakdown Structure (WBS) developed earlier was utilized as a reference when developing the Gantt Chart for the project. The main tasks and supporting tasks for the project were comprehensively broken down in the WBS. Following with the six project deliverables that had been determined, additional decomposition was done to divide each task into easier-to-manage subtasks. For instance, the Reflective Report was further divided into "Self-Reflection" and "List of Chapters of the Final Report," while the Software Artefact was further split into

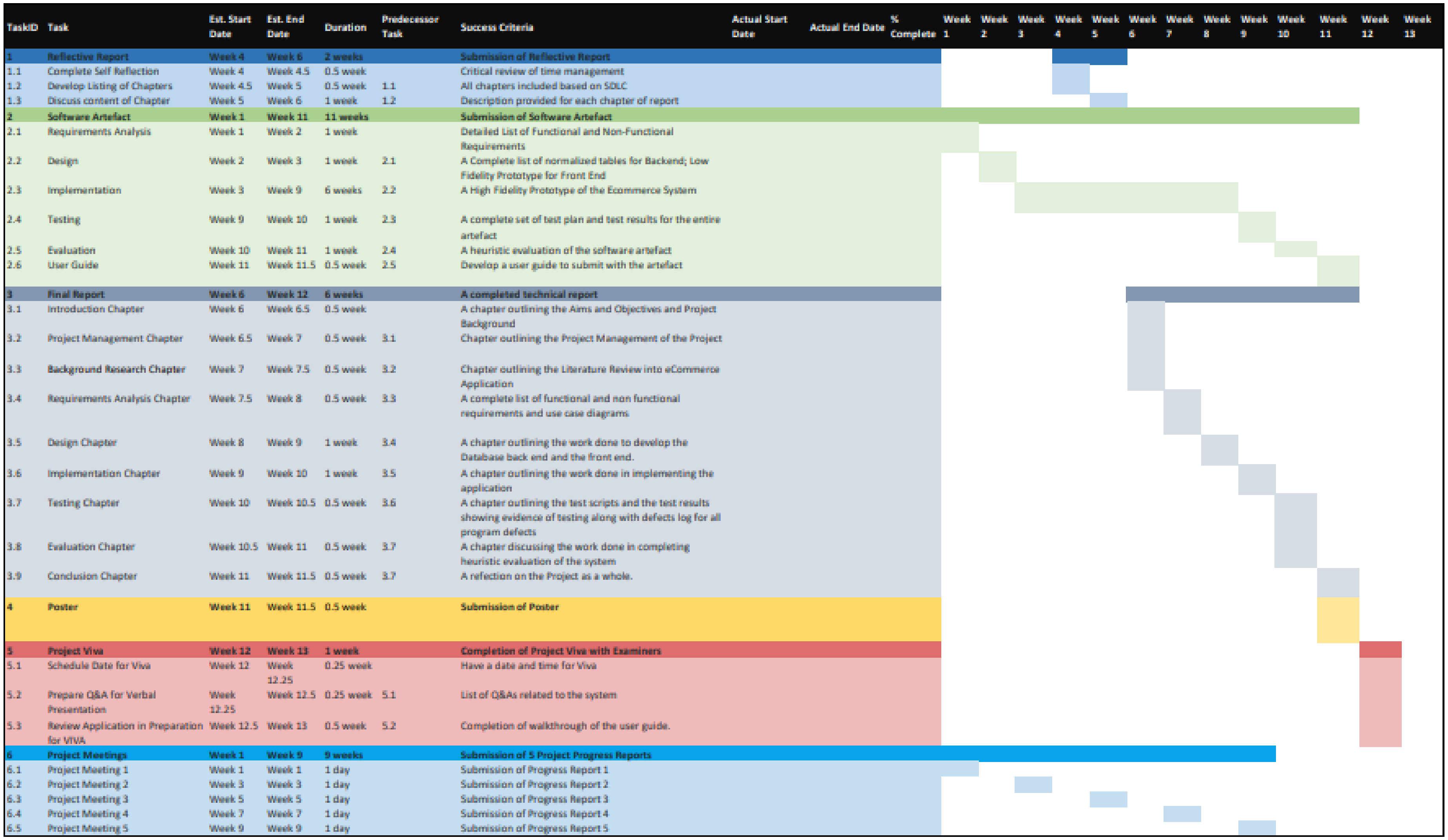
"Develop Core Features" and "Develop Advanced Features." This breakdown allowed for a clear understanding of the various steps required to complete each deliverable. These subtasks were subsequently organized in chronological sequence, taking dependencies and prerequisites into account, to construct a timeline that serves as the foundation for the Gantt Chart. This approach ensured that tasks were structured and logically organized, allowing for efficient project execution and monitoring.

In the development of the Gantt Chart, several columns were integrated to enhance the project management and monitoring process. These columns included:

- % Complete: The percentage of work performed for each task was tracked in this column. It enabled real-time project status evaluation and gave a visual depiction of work progression.
- Actual Start Date and End Date: These columns indicated the precise dates that tasks started and ended. With the use of this data, it was possible to compare the actual schedule with the one that had been intended to recognize any delays or early completions.
- Predecessor Tasks: The Predecessor Tasks column included any tasks that had to be accomplished before starting a new activity. This dependency information ensured that tasks were completed in an appropriate order, avoiding unnecessary delays.
- Success Criteria: This column specifies the requirements that must be met for each activity to be regarded as accomplished effectively. It guided project execution by providing a precise definition of work outcomes.
- Duration: The Duration column stated how long each job was expected to take. Providing an overview of the time commitment for each task, aided in resource allocation and time management.

The inclusion of these columns in the Gantt Chart increased project transparency, accuracy, and overall management. It enabled better work progress tracking, early detection of potential delays, and more informed decision-making. Ultimately, adding to the project’s prosperity.

Microsoft Excel was used for developing the Gantt Chart. Excel's wide accessibility, intuitive interface, and ability to efficiently organize and present project timelines, task progress, dependencies, and additional columns for greater project tracking and management influenced this decision. The software's familiarity and versatility were crucial in making it the favoured option for constructing the Gantt Chart. The Gantt Chart is displayed in Figure 2 below.



TaskID	Task	Est. Start Date	Est. End Date	Duration	Predecessor Task	Success Criteria	Actual Start Date	Actual End Date	% Complete	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13
1	Reflective Report	Week 4	Week 6	2 weeks		Submission of Reflective Report							■	■								
1.1	Complete Self Reflection	Week 4	Week 4.5	0.5 week		Critical review of time management							■									
1.2	Develop Listing of Chapters	Week 4.5	Week 5	0.5 week	1.1	All chapters included based on SDLC							■									
1.3	Discuss content of Chapter	Week 5	Week 6	1 week	1.2	Description provided for each chapter of report								■								
2	Software Artefact	Week 1	Week 11	11 weeks		Submission of Software Artefact				■	■	■	■	■	■	■	■	■	■	■		
2.1	Requirements Analysis	Week 1	Week 2	1 week		Detailed List of Functional and Non-Functional Requirements				■												
2.2	Design	Week 2	Week 3	1 week	2.1	A Complete list of normalized tables for Backend; Low Fidelity Prototype for Front End					■											
2.3	Implementation	Week 3	Week 9	6 weeks	2.2	A High Fidelity Prototype of the Ecommerce System						■	■	■	■	■	■					
2.4	Testing	Week 9	Week 10	1 week	2.3	A complete set of test plan and test results for the entire artefact												■				
2.5	Evaluation	Week 10	Week 11	1 week	2.4	A heuristic evaluation of the software artefact													■			
2.6	User Guide	Week 11	Week 11.5	0.5 week	2.5	Develop a user guide to submit with the artefact														■		
3	Final Report	Week 6	Week 12	6 weeks		A completed technical report									■	■	■	■	■	■		
3.1	Introduction Chapter	Week 6	Week 6.5	0.5 week		A chapter outlining the Aims and Objectives and Project Background									■							
3.2	Project Management Chapter	Week 6.5	Week 7	0.5 week	3.1	Chapter outlining the Project Management of the Project									■							
3.3	Background Research Chapter	Week 7	Week 7.5	0.5 week	3.2	Chapter outlining the Literature Review into eCommerce Application									■							
3.4	Requirements Analysis Chapter	Week 7.5	Week 8	0.5 week	3.3	A complete list of functional and non functional requirements and use case diagrams										■						
3.5	Design Chapter	Week 8	Week 9	1 week	3.4	A chapter outlining the work done to develop the Database back end and the front end.											■					
3.6	Implementation Chapter	Week 9	Week 10	1 week	3.5	A chapter outlining the work done in implementing the application												■				
3.7	Testing Chapter	Week 10	Week 10.5	0.5 week	3.6	A chapter outlining the test scripts and the test results showing evidence of testing along with defects log for all program defects													■			
3.8	Evaluation Chapter	Week 10.5	Week 11	0.5 week	3.7	A chapter discussing the work done in completing heuristic evaluation of the system													■			
3.9	Conclusion Chapter	Week 11	Week 11.5	0.5 week	3.7	A refection on the Project as a whole.														■		
4	Poster	Week 11	Week 11.5	0.5 week		Submission of Poster														■		
5	Project Viva	Week 12	Week 13	1 week		Completion of Project Viva with Examiners															■	
5.1	Schedule Date for Viva	Week 12	Week 12.25	0.25 week		Have a date and time for Viva															■	
5.2	Prepare Q&A for Verbal Presentation	Week 12.25	Week 12.5	0.25 week	5.1	List of Q&As related to the system															■	
5.3	Review Application in Preparation for VIVA	Week 12.5	Week 13	0.5 week	5.2	Completion of walkthrough of the user guide.															■	
6	Project Meetings	Week 1	Week 9	9 weeks		Submission of 5 Project Progress Reports				■	■	■	■	■	■	■	■	■				
6.1	Project Meeting 1	Week 1	Week 1	1 day		Submission of Progress Report 1				■												
6.2	Project Meeting 2	Week 3	Week 3	1 day		Submission of Progress Report 2						■										
6.3	Project Meeting 3	Week 5	Week 5	1 day		Submission of Progress Report 3								■								
6.4	Project Meeting 4	Week 7	Week 7	1 day		Submission of Progress Report 4										■						
6.5	Project Meeting 5	Week 9	Week 9	1 day		Submission of Progress Report 5												■				

Figure 2: Displaying the Gantt Chart.

Background Research

Introduction

A literature review is a methodical analysis of prior research on a specific subject. It is crucial for contextualizing research, identifying trends, avoiding duplication, generating research questions, guiding methodology, building theoretical frameworks, assessing quality, informing policy, contributing to theory, and ensuring that new research is informed, relevant, and advances knowledge within a given field (Snyder, 2019).

This project's literature review will systematically compile and synthesize existing research regarding developing an online store for health and wellness products that uses gamification and artificial intelligence. Using databases such as Google Scholar and Emerald Insights Library, suitable articles will be selected based on predefined criteria, using qualitative and quantitative synthesis approaches (Xiao & Watson, 2017). The assessment will shape the project's theoretical framework and design by critically examining approaches and spotting trends, ensuring a strong and evidence-based approach to development and evaluation.

E-commerce Systems

E-Commerce

In this section, we'll look at the evolution of e-commerce, its impact on daily life, successful strategies, problems encountered, current trends, and future possibilities. Furthermore, we will examine how developing technologies affect e-commerce and address the corresponding challenges.

What exactly is e-commerce?

The term "e-commerce," short for electronic commerce, refers to the online exchange of goods and services. This encompasses various models like B2B, B2C, and C2C, all facilitated by digital platforms that cater to evolving consumer trends and market dynamics. By harnessing the power of the internet and computer networks, e-commerce blurs the lines between traditional and global markets, driving a transformative shift (Gupta, 2014).

The growth and evolution of e-commerce

The evolution of e-commerce has been closely intertwined with the rapid advancement of information and communication technology (ICT) over the last 15 years (Silva, et al., 2010). In the early stages, particularly during the 1980s, the internet user base remained limited, with gradual growth attributed to emerging text-based functionalities like email and file transfers. However, the pivotal turning point was marked by the introduction of the World Wide Web, ushering in an extraordinary surge in the number of Internet users (Kansana, et al., 2016). This growth has set an unprecedented benchmark in the history of communication media.

Notably, the International Telecommunication Union (ITU), a United Nations agency, forecasted that by 2015, an estimated 3.2 billion individuals would have internet connectivity, an astonishing statistic considering the global population was approximately 7.2 billion at that time. In contrast, a mere 400 million people were online in the year 2000 (ITU, 2015), showcasing the remarkable trajectory of digital adoption.

Furthermore, UNCTAD's (2021) research highlights that the e-commerce sector experienced a significant uptick in its share of total retail sales, rising from 16% to 19% in 2020. This shift is particularly evident in the Republic of Korea, which witnessed internet sales escalating from one in five transactions in 2019 to more than one in four the following year. These developments underscore the increasing significance of online transactions and emphasize the critical role of such data, especially for developing nations navigating post-COVID-19 economic recovery.

This trend was echoed in several other countries, including the UK, China, the US, Australia, Singapore, and Canada, which observed substantial spikes in online transactions during the same period. Additionally, UNCTAD's report highlighted that the top 13 global companies registered a remarkable $2.9 trillion in online business-to-consumer (B2C) sales in 2020. *Figure 3* below displays the online retail sales based on selected economies for the period 2018-2022.

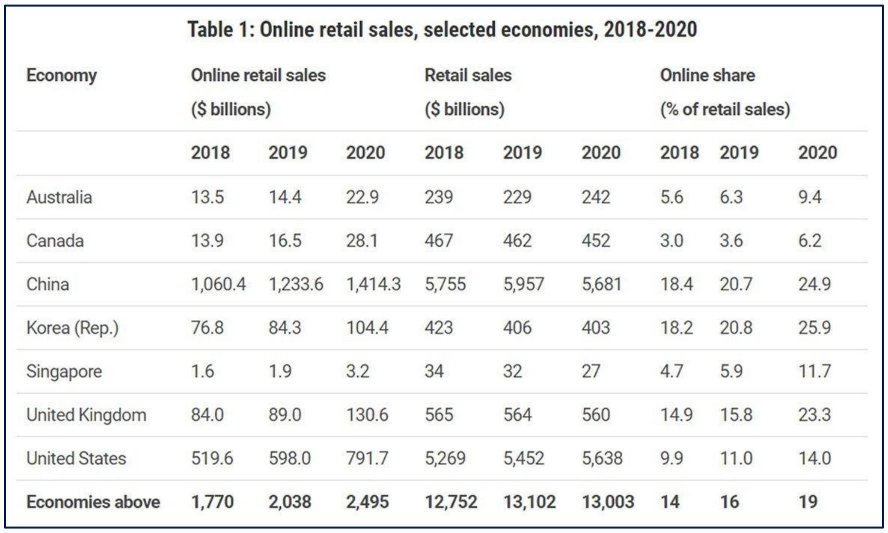


Table 1: Online retail sales, selected economies, 2018-2020

Economy	Online retail sales ($ billions)			Retail sales ($ billions)			Online share (% of retail sales)		
	2018	**2019**	**2020**	**2018**	**2019**	**2020**	**2018**	**2019**	**2020**
Australia	13.5	14.4	22.9	239	229	242	5.6	6.3	9.4
Canada	13.9	16.5	28.1	467	462	452	3.0	3.6	6.2
China	1,060.4	1,233.6	1,414.3	5,755	5,957	5,681	18.4	20.7	24.9
Korea (Rep.)	76.8	84.3	104.4	423	406	403	18.2	20.8	25.9
Singapore	1.6	1.9	3.2	34	32	27	4.7	5.9	11.7
United Kingdom	84.0	89.0	130.6	565	564	560	14.9	15.8	23.3
United States	519.6	598.0	791.7	5,269	5,452	5,638	9.9	11.0	14.0
Economies above	**1,770**	**2,038**	**2,495**	**12,752**	**13,102**	**13,003**	**14**	**16**	**19**

Figure 3: Displaying online retail sales based on selected economies for the period 2018-2022.

Source: (UNCTAD, 2021).

Looking ahead, the trajectory of e-commerce growth remains promising. Forecasts suggest that the global e-commerce landscape is set to witness a noteworthy expansion in 2023, with a projected growth rate of 8.9%, propelling worldwide e-commerce sales to a remarkable $5.8 trillion. This growth momentum is anticipated to continue at a steady pace, as a growth rate of 9.4% is predicted for 2024, followed by a slightly moderated growth of 8.6% in 2025. The subsequent year, 2026, is expected to maintain this consistent growth, reaching e-commerce sales of approximately $7.5 trillion. Further, into the future, 2027 is anticipated to bring a growth rate of 7.6%, propelling total e-commerce sales beyond the remarkable $8 trillion threshold for the very first time.

During the period between 2023 and 2027, the most significant annual growth in e-commerce sales is expected to occur in 2024, while the lowest growth is projected for 2027. This projection indicates a substantial increase in total sales of $2.3 trillion, reflecting an impressive overall growth of 38.9% and an average annual growth rate of 8.6%. These estimations underscore the resilience and expanding potential of the e-commerce sector on a global scale. *Figure 4* below displays the global e-commerce growth rate (2023-2027).

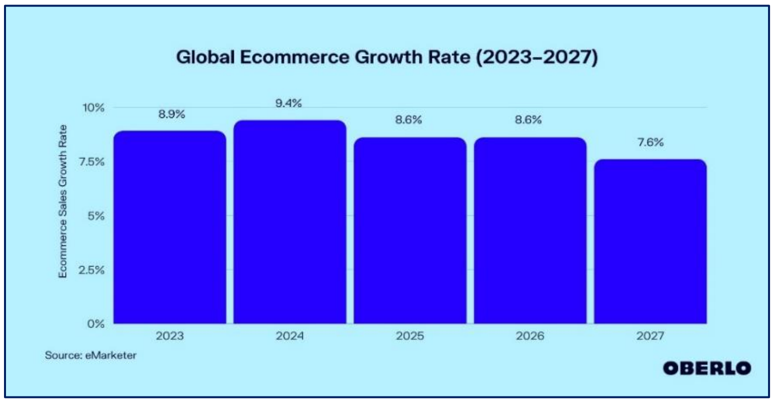


Figure 4: Displaying the global e-commerce growth rate (2023-2027).

Source: (eMarketer, 2023)

Looking globally, retail and e-commerce sales are poised for a notable rebound this year, characterized by accelerated digital sales growth and a gradual deceleration in overall retail (Anonymous, 2023). While most national markets are regaining a sense of normalcy, variations across countries persist. E-commerce sales growth worldwide is anticipated to regain momentum after a dip in 2022, with projections indicating a growth rate surpassing 6.5%. As we approach the next year, total sales are expected to cross the $6 trillion mark, with a significant contribution from China, cementing its role as a global e-commerce leader.

In this progressive e-commerce landscape, the intricate dance between technology, consumer behaviour, and market dynamics is shaping an ever-evolving digital future. The fusion of technology and commerce continues to revolutionize the way businesses operate and consumers engage, laying the foundation for a dynamic and promising e-commerce era. *Figure 5* below displays the retail e-commerce sales worldwide for the period 2021 to2027.

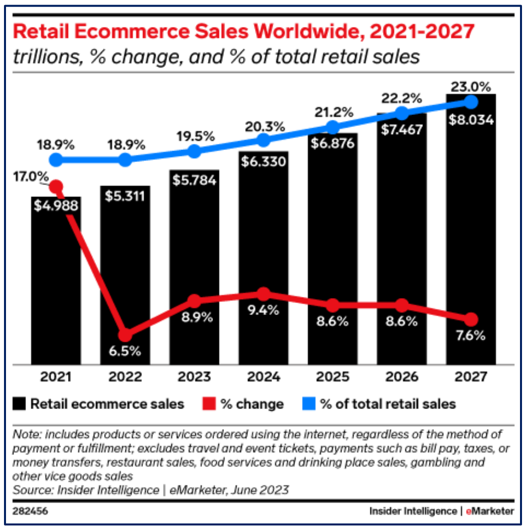


Figure 5: Displaying retail e-commerce sales worldwide, 2021-2027.

Source: (eMarketer, 2023)

<u>The impact of e-commerce</u>

The evolving relationship between commerce and technology has significantly impacted how we engage in various activities (Rahayu & Rahayu, 2015). With the rapid advancements in information and communication technology (ICT) in recent decades, traditional commerce methods have struggled to meet modern demands. This has given rise to new approaches, like e-commerce, which are essential for the success of small and developing businesses in today's global economy (Cegarra-Navarro, et al., 2007). The ubiquity of the internet has transformed our lives, creating job opportunities and reshaping business practices (Jai, et al., 2013). The integration of electronic technologies, particularly IT, has led to e-commerce's emergence, altering the way we conduct economic transactions (Feizollahi, et al., 2014). As a result, optimizing e-commerce efficiency relies heavily on effectively using IT resources. This transition also involves breaking information monopolies, introducing novel commercial methods, and improving overall efficiency (Salehi, et al., 2012). The impact of e-commerce on society is profound, shifting the dynamics of buying and selling, improving customer-supplier communication, and fostering economic growth (Yang, et al., 2015). As information and communication technologies continue to advance, organizations must embrace them for competitive advantages, propelling development and prosperity (Borges, et al., 2009). Indeed, e-commerce's impact has revolutionized our approach to business and daily activities (MacGregor, 2005).

Advantages and Disadvantages of E-Commerce

The surge in online shopping has become increasingly evident, offering numerous benefits to customers, businesses, and society. E-commerce's transformative impact has revolutionized traditional shopping practices, saving time and energy for customers who previously faced the hassle of physical store visits (Taher, 2021). This trend is driven by a multitude of reasons, leading to a mutually beneficial situation for consumers and merchants alike (Taher, 2021). *Table 2* displays the advantages of e-commerce for Customers, Businesses and Society.

Table 2: Displaying the advantages of e-commerce for Customers, Businesses and Society.

Groups	Advantages	Description
Customers	24/7 Accessibility	E-commerce platforms are open around the clock, enabling customers to shop at their convenience, overcoming time constraints and providing ease of use through well-designed websites.
	Convenience	E-commerce has emerged as the most convenient way to shop, allowing consumers to make purchases using internet-connected devices from any location globally.
	Time Savings	E-commerce accelerates buying and selling processes, significantly saving customers' time by enabling quick transactions and delivery within a week.
	Access to Information	E-commerce provides customers with comprehensive product information, purchase history, and the ability to assess value before making a purchase.
	Price Comparison	Online platforms offer a wider array of product options and prices, facilitating easy comparison, and leading to informed purchasing decisions.
Businesses	Higher Advertising Returns	Well-managed e-commerce campaigns yield higher returns on advertising investments.
	No Geographical Constraints	E-commerce eliminates geographical barriers, enabling businesses to expand to national and international markets with minimal investment.
	Precise Targeting	E-commerce platforms collect consumer data for precise audience targeting, optimizing marketing efforts and driving higher engagement and sales.
	Cost Savings	E-commerce incurs lower operational costs compared to physical stores, as there's no need for staff, rent, or substantial fixed expenses.

Society	Reduced Pollution	Online shopping minimizes physical transportation, decreasing street congestion and air pollution.
	Improved Government Services	E-commerce aids in offering communal services, such as healthcare and education, at reduced costs with enhanced accessibility (Clarke, 1999).
	Affordability	Lower operational expenses in e-commerce result in reduced product prices, making items accessible to a broader range of income groups.
	Accessibility to Remote Areas	Previously inaccessible rural areas can now access goods and services through e-commerce.

E-commerce's advantages are evident in its convenience, cost savings, improved efficiency, and positive impact on society. This digital transformation offers a win-win scenario for all stakeholders involved.

E-commerce, while offering numerous benefits, also comes with its own set of challenges and drawbacks, falling into two main categories: technical and non-technical. *Table 3* below displays the technical and non-technical disadvantages of e-commerce.

Table 3: Displaying technical and non-technical disadvantages of e-commerce.

Category	Disadvantages	Descriptions
Non-Technical	Inability to Test Products	Online shopping prevents customers from physically testing items before purchasing, leading to uncertainty about product quality.
	Lack of Personal Interaction	E-commerce lacks the personal touch and interaction found in physical stores, inhibiting the development of relationships between customers and sales assistants.
	Delivery Delays	Online shopping involves waiting for product delivery, delaying the gratification that comes with immediate in-store purchases.
	Damage during delivery	Products purchased online can be damaged during transportation and delivery.
	Restricted Customer Services	Online customer service can be limited in terms of availability and responsiveness compared to assistance provided in physical stores.
Technical	Security Concerns	Cybersecurity is a global issue, and inadequate security systems pose risks to customer data and erode trust between customers and providers.

	Dependency on Internet	Online shopping requires continuous internet access and compatible devices for customer participation.
	Credit Card Fraud	The rising threat of credit card fraud impacts both customers and businesses, potentially leading to financial losses and reputational damage.
	Software Development Challenges	Online companies must constantly update and maintain software and hardware, presenting ongoing challenges for development and compatibility.

In conclusion, e-commerce offers numerous benefits such as 24/7 shopping convenience, cost savings, efficient business operations, and improved societal access to services. However, challenges include the inability to physically test products, lack of personal interaction, delivery delays, limited customer service, and technical issues like security concerns and software development. Balancing these advantages and disadvantages is essential for optimizing the e-commerce experience. As technology advances, addressing these drawbacks becomes crucial to ensuring a seamless and secure online shopping environment for customers and businesses alike.

Websites Reviewed

The dynamic landscape of e-commerce is continuously evolving with the integration of cutting-edge technologies. To stay competitive and cater to the ever-changing preferences of consumers, businesses are adopting innovative strategies that enhance user experiences and drive sales. From AI-driven personalization to sustainable practices, these trends are reshaping the way e-commerce operates. *Table 4* below displays some of the current trends in e-commerce.

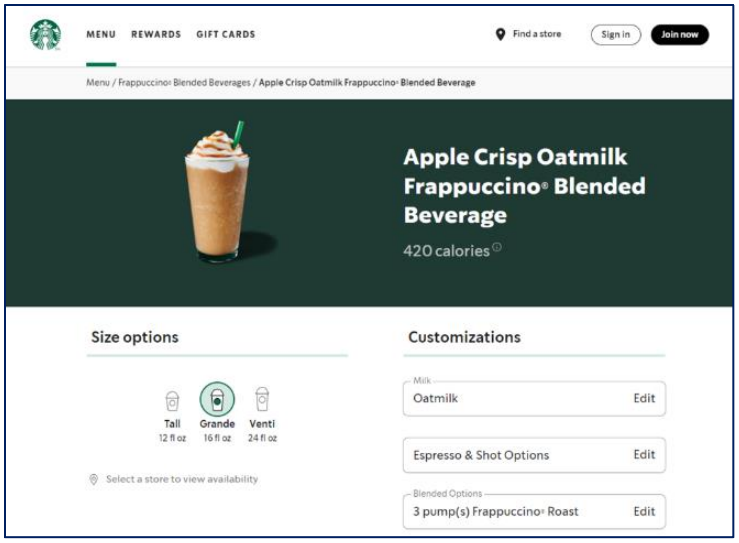
Table 4: Displaying Current trends in e-commerce.

No	**Trend**	**Description**
1	AI-Enabled Personalization	AI technology is revolutionizing e-commerce by tailoring user experiences based on preferences and behaviours, leading to a more engaging and relevant shopping journey.
2	Hybrid Commerce Integration	Connecting online and physical store experiences seamlessly improves customer engagement and boosts purchase rates, offering a cohesive shopping journey.

3	Voice and Visual Search Optimization	Optimizing for voice and visual search accommodates changing user behaviours, enhancing the e-commerce site's accessibility and user-friendliness.
4	Diverse Payment Options	Providing various payment methods caters to different customer preferences, including traditional methods and newer options like digital wallets and BNPL.
5	Augmented and Virtual Reality	Immersive experiences through AR and VR empower customers to inspect products closely, enhancing their confidence in making purchases.
6	Social Selling Strategies	Leveraging social commerce, influencer marketing, and live shopping harnesses the power of social media platforms for cost-effective and in-app purchases.
7	Marketing Automation	Automating marketing processes streamlines workflows, boosts productivity, and enhances conversions, making campaigns more efficient.
8	Subscription Models	Offering subscription-based services not only ensures customer convenience for recurring expenses but also strengthens loyalty and profitability.

These trends epitomize the intersection of cutting-edge technology and customer-centric approaches, redefining how e-commerce enterprises create immersive experiences and cater to the ever-changing demands of their customers. Two eye-catching trends which are AI-Enabled Personalization and Voice and Visual Search Optimization will be analyzed in depth to understand how they are reshaping the e-commerce landscape.

The fusion of Artificial Intelligence (AI) with e-commerce ushers in a transformative era, leveraging data analysis and AI's human-like intelligence for profound consumer insights. A pivotal application is personalized shopping experiences, fostering customer loyalty. Forecasts predict the global personalization software market to reach over $5 billion by 2030 from $943.25 million in 2022. Starbucks exemplifies this trend, customizing 60% of cold beverage orders using real-time customer data via their mobile app. As the global AI market surges to $407 billion by 2027 from $86.9 billion in 2022, 91.5% of leading brands consistently invest in AI. This AI and machine learning synergy, epitomized by Starbucks, embodies the future of personalized e-commerce. Refer to *Figure 6* below to observe how Starbucks integrates AI-driven personalization into their website.


MENU
REWARDS
GIFT CARDS
Find a store
Sign in
Join now
Menu / Frappuccino® Blended Beverages / Apple Crisp Oatmilk Frappuccino® Blended Beverage

Apple Crisp Oatmilk Frappuccino® Blended Beverage

420 calories

Size options

Tall 12 fl oz
Grande 16 fl oz
Venti 24 fl oz

Select a store to view availability

Customizations

Milk: Oatmilk — Edit

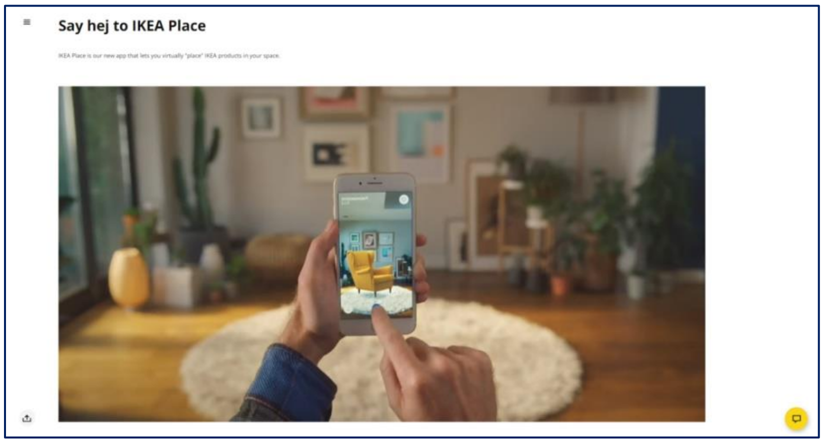
Espresso & Shot Options — Edit

Blended Options: 3 pump(s) Frappuccino® Roast — Edit

Figure 6: Displaying AI-Enabled Personalization used by Starbucks.

Source: (Starbucks, 2023)

Projected to hit $856.2 billion by 2031, the AR/VR market boasts a 41.1% ten-year CAGR. Augmented Reality (AR) merges reality with computer-generated elements, while Virtual Reality (VR) offers simulated environments through specialized headsets. Many eCommerce brands are embracing these technologies for immersive shopping. Positive consumer response is evident, with 71% expressing increased shopping intent due to AR, and companies with AR/VR content showing a 94% higher conversion rate. IKEA, a pioneering example, employs Apple's ARKit platform in their IKEA Place app. Refer to *Figure 7* to observe how Apple's ARKit utilizes Visual Search Optimization, allowing users to virtually position 3D products in their environments, thus enriching their decision-making process.


Say hej to IKEA Place

IKEA Place is our new app that lets you virtually "place" IKEA products in your space.

Figure 7: Displaying use case of Visual Search Optimization.

Source: (Apple, 2023)

The future of e-commerce is shaped by emerging technologies that are set to revolutionize the industry. Artificial Intelligence (AI) and Machine Learning (ML) are key players, enhancing personalized customer experiences by providing tailored product recommendations based on extensive data analysis (Edelman & Abraham, 2023). Voice and Visual Search Optimization is another pivotal trend, as e-commerce platforms are optimized for voice assistants and image recognition, streamlining user interactions.

Augmented Reality (AR) and Virtual Reality (VR) are transforming online shopping with immersive experiences that allow customers to virtually try products (Xiong, et al., 2021). Blockchain technology ensures secure and transparent transactions, boosting consumer trust (Chen, 2020). Sustainability practices are gaining prominence, with eco-conscious consumers favouring brands with environmentally responsible practices, such as eco-friendly packaging and sustainable sourcing.

In conclusion, the future of e-commerce lies in AI-driven personalization, voice and visual search, AR/VR integration, blockchain security, and sustainability practices. These innovations are reshaping the industry, providing enhanced shopping experiences and redefining customer interactions with online stores.

Limitations in integrating emerging technologies into e-commerce.

The integration of emerging technologies into e-commerce presents several limitations and challenges. Firstly, concerns about data privacy and security arise due to the increased collection of customer information for personalization. Secondly, the cost of implementing and maintaining advanced technologies can be prohibitive for small businesses. Thirdly, the digital divide could exclude certain demographics from benefiting fully. Fourthly, the rapid pace of technological change may result in compatibility issues and the need for constant updates. Lastly, the potential for job displacement and ethical concerns regarding AI-driven decision-making are also points of contention (Martens, 2022).

Ecommerce Functionalities

Artificial Intelligence

AI, or artificial intelligence, refers to the simulation of human-like cognitive processes and decision-making capabilities in machines. It involves the development of computer systems that can perform tasks that typically require human intelligence. These tasks include understanding natural language, recognizing patterns, solving complex problems, learning from experience, and making informed decisions. AI systems can analyze large amounts of data, extract meaningful insights, and adapt their behaviour based on the information they process (Suleiman, et al., 2021). Some specific applications of **AI include:**

- Eye-Recognition and Face Recognition Software: These are examples of AI-powered technologies that can identify and verify individuals based on their unique facial features or eye patterns.
- 3D and 4D Apps for Product Testing and Virtual Tours: These apps use AI to create interactive and immersive experiences, allowing users to virtually explore products, apartments, or travel destinations.
- Machine Learning: Machine learning is a subset of AI that involves training algorithms to learn from data and improve their performance over time. It's used for tasks like making predictions, identifying patterns, and automating decision-making.
- Digital Marketing: AI is utilized in digital marketing to analyze consumer behaviour, personalize advertisements, optimize marketing campaigns, and enhance customer engagement.
- Automation: AI-powered systems can automate repetitive tasks, which increases efficiency and reduces the need for manual labour in various industries, including manufacturing and services.

In summary, AI encompasses a range of technologies and techniques that enable machines to perform tasks that were traditionally associated with human intelligence. These technologies have far-reaching implications across industries and are driving advancements in areas such as automation, data analysis, and decision-making.

Chatbot

Certainly, one noteworthy implementation of artificial intelligence is the development and utilization of chatbots. Chatbots represent a compelling example of how AI can seamlessly integrate into our daily lives, enhancing human-computer interaction and offering a range of practical applications across various domains (Bansal & Khan, 2018).

A chatbot is essentially an AI-driven computer program designed to simulate conversation with human users. By leveraging Natural Language Processing (NLP), chatbots can understand and respond to text or voice-based inputs in one or more human languages (Khanna, et al., 2015). They serve as intelligent agents that can perform a wide spectrum of tasks, from simple conversational interactions to more complex operations.

In the context of Human-Computer Interaction (HCI), chatbots serve as intelligent entities that engage with users in a manner akin to human conversation. They can be integrated into messaging apps or platforms, enabling users to interact with them without the need for additional installations (Klopfenstein, et al., 2017). This seamless accessibility has contributed to the widespread adoption and popularity of chatbots.

The advantages of chatbots extend to both users and developers, making them a valuable tool in various applications. For users, chatbots offer:

- Ease of Access: Chatbots are instantly available within messaging apps, making them easily accessible without requiring users to install separate applications.
- Social Integration: Chatbots can be shared and interacted with within a user's existing social network or messaging platform, ensuring the user's identity and facilitating engagement.
- Integrated Services: Payment services and other functionalities can be seamlessly integrated into chatbots, offering users a safe and reliable way to conduct transactions.
- Re-Engagement: Chatbots can use notification systems to re-engage with users who may have become inactive, enhancing user retention.
- Parallel Conversations: Users can engage in multiple conversations with chatbots simultaneously, enabling efficient multitasking.
- Transferable Knowledge: Skills and interactions learned from using one chatbot can often be transferred to the use of other chatbots, reducing the learning curve for new applications.

In summary, chatbots exemplify how AI technologies like Natural Language Processing can transform human-computer interaction. They serve as versatile tools that not only entertain but also find utility in education, information retrieval, business, e-commerce, and more (Shawar & Atwell, 2007). As AI technology continues to evolve, chatbots are likely to become even more sophisticated, enhancing their ability to provide meaningful and seamless interactions for users across various contexts.

History of Chatbot

The history of chatbots has spanned decades, showcasing the progression of artificial intelligence and human-computer interaction. It all began in 1950 when Alan Turing proposed the Turing Test to explore whether machines could emulate human-like intelligence through conversation (Turing, 2009). A significant milestone arrived in 1966 with the development of Eliza, the first known chatbot, designed as a psychotherapist that engaged users in dialogue with question-like responses (Weizenbaum, 1996). Subsequent years brought forth advancements, such as PARRY in 1972, which simulated a persona with paranoid schizophrenia (Colby, et al., 1971). The mid-1990s introduced ALICE, a chatbot leveraging pattern matching and the Artificial Intelligence Markup Language (AIML), and subsequently won the Loebner Prize (Wallace, 2009). In the early 2000s, chatbots like SmarterChild (Molnár & Szüts, 2018) emerged on messenger platforms, leading to the creation of virtual personal assistants like Apple's Siri (Siri, 2023), Microsoft's Cortana (Cortana Home Assistant – Microsoft., 2023), Amazon's Alexa (Amazon Alexa, 2023), Google Assistant (Google Assistant, 2023), and IBM Watson (IBM Watson, 2023). These sophisticated systems marked a transition from basic chatbots to versatile assistants capable of task execution and information provision through natural language interactions. This historical journey underscores the evolution and increasing integration of chatbots and AI into our daily lives.

Illustrated in *Figure 8*, as depicted by Scopus (2022), there has been a notable surge in the interest surrounding chatbots, particularly post-2016. This heightened attention has led to the development of numerous chatbots catering to industrial needs. Simultaneously, a diverse array of lesser-known chatbots has emerged, with significant relevance to research and its various applications (Colace, et al., 2018).

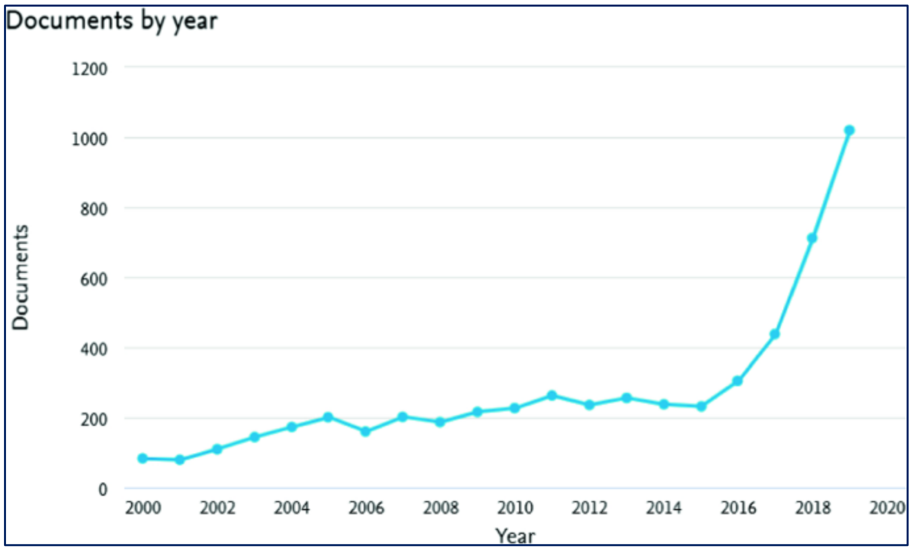


Figure 8: Displaying notable surge in the interest surrounding chatbots between 2000 and 2020.

__Fundamental concepts related to Chatbot.__

Pattern Matching: Pattern Matching relies on distinctive stimulus-response segments. This involves inputting a sentence (stimuli) and generating an output (response) that aligns with the user's input (Marietto, et al., 2013). Eliza and ALICE, the pioneering chatbots, were constructed using pattern recognition algorithms. However, this approach bears a drawback: the responses tend to be entirely foreseeable, monotonous, and devoid of human-like nuance. Additionally, since past responses aren't stored, it can result in repetitive loops in conversations (Ramesh, et al., 2017).

Artificial Intelligence Markup Language (AIML): Developed between 1995 and 2000, the Artificial Intelligence Markup Language (AIML) draws its origins from Pattern Recognition, specifically the Pattern Matching technique. AIML finds applications in modelling natural language for human-chatbot interactions that adhere to the stimulus-response method. This XML-based markup language employs tags and is exemplified in Figure 2. AIML revolves around fundamental dialogue units labelled as categories (within the <category> tag), comprising user input patterns (<pattern> tag) and chatbot responses (<template> tag) (Marietto, et al., 2013). *Figure 9* represents an example of AIML code.

```
<aiml version="1.0.1" encoding="UTF-8"?>
   <category>
      <pattern> My name is * and I am * years old </pattern>
      <template> Hello <star/>. I am also <star index="2"/> years old!</template>
   </category>
</aiml>
```

Figure 9: Displaying an example of AIML code.

Latent Semantic Analysis (LSA): The integration of Latent Semantic Analysis (LSA) with AIML holds potential in chatbot advancement. LSA serves to unveil semantic connections among words through vector representation (Ahmad, et al., 2018). While AIML can effectively handle template-based inquiries like greetings and common questions, the introduction of LSA allows for responses to more intricate and unanswered queries (Thomas, 2016).

Chatscript: Chatscript, the successor to AIML, embodies an expert system encompassing an open-source scripting language and its corresponding engine. Operating via rules linked to topics, Chatscript identifies the most fitting item aligning with the user's query and then executes a rule within that specific topic. This system incorporates $ variables, functioning as long-term memory to retain user details such as names or ages. Notably case sensitive, Chatscript expands the spectrum of responses feasible for a given user input, capitalizing on uppercase to convey emphasis, a common conversational practice (Ramesh, et al., 2017).

RiverScript: RiverScript stands as a text-based scripting language employed in crafting chatbots and similar conversational entities. This open-source tool offers interfaces for various programming languages including Go, Java, JavaScript, Perl, and Python (RiveScript, 2023).

Natural Language Processing (NLP): Natural Language Processing (NLP), a branch of artificial intelligence, delves into the computer-driven handling of natural language text or speech. It harnesses insights into human language comprehension and usage to create methodologies that enable computers to grasp and manipulate natural expressions to accomplish specific objectives (Jung, 2019). A substantial portion of NLP techniques rests on the foundation of machine learning.

Natural Language Understanding (NLU): At the heart of various NLP endeavours lies Natural Language Understanding (NLU), a cornerstone technique essential for crafting natural user interfaces like chatbots. NLU's purpose revolves around deciphering contextual cues and nuances embedded in unstructured natural language inputs, thus enabling suitable responses aligned with user intentions (Jung, 2019). This process involves identifying user intents and extracting pertinent entities linked to specific domains. More precisely, intents act as bridges between user utterances and corresponding chatbot actions. These actions encompass the steps to be taken when particular intents are triggered by user inputs, potentially accompanied by parameters furnishing intricate details (Ramesh, et al., 2017). Intent detection typically takes the form of sentence classification, predicting one or multiple intent labels for each sentence (Jung, 2019).

Design and Development of Chatbot

The creation and development of a chatbot encompasses a diverse array of techniques (Ahmad, et al., 2018). This process involves comprehending the chatbot's intended offerings and categorization, which aids developers in selecting suitable algorithms, platforms, and tools for its construction. Simultaneously, this clarity benefits end-users in understanding the chatbot's capabilities (Nimavat & Champaneria, 2017).

Essential requisites for chatbot design encompass precise knowledge representation, a strategy for generating responses, and a predefined set of generic replies for instances when user input isn't comprehended (Augello, et al., 2018). Initiating the design of any system involves segmenting it into constituent parts, following a standardized approach that enables modular development (Ramesh, et al., 2017). *Figure 10* presents a comprehensive illustration of a general chatbot architecture.

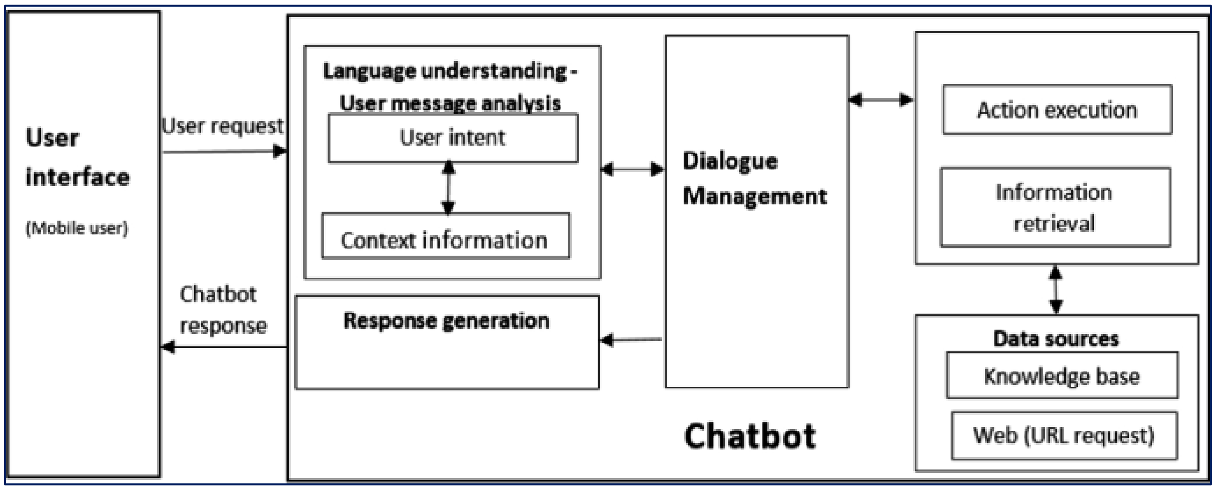


Figure 10: Displaying a comprehensive illustration of a general chatbot architecture.

The chatbot development process involves user requests via messaging apps or platforms like Amazon Echo (Zumstein & Hundertmark, 2017). The Language Understanding Component interprets intent and information from user inputs (Kucherbaev, et al., 2018). Once understood, the chatbot decides its next step, which can include immediate action, retention of information, seeking context, or seeking clarification (Verloop.io, 2023). It then executes actions or retrieves data from sources like databases or external resources through API calls (Kucherbaev, et al., 2018). The Response Generation Component crafts human-like replies using Natural Language Generation (Singh, et al., 2016). Dialogue Management sustains context, prompts for missing details, and poses follow-up questions (Kucherbaev, et al., 2018). Development options range from programming languages to NLU platforms like DialogFlow, wit.ai, LUIS, Watson Conversation, Amazon Lex, and SAP Conversation AI. These platforms leverage machine learning and differ in features. Other platforms include RASA, Botsify, Chatfuel, Manychat, Flow XO, Chatterbot, Pandorabots, Botkit, and Botlytics.

SWOT Analysis of Chatbot

Table 5 displays a SWOT analysis of the chatbot. This SWOT analysis provides a comprehensive overview of the strengths, weaknesses, opportunities, and threats associated with chatbots.

Table 5: Displaying a SWOT analysis of chatbot.

Positive		**Negative**	
Strengths	**Description**	**Weaknesses**	**Description**
24/7 Availability	Customer service via chatbots is round-the-clock, enhancing satisfaction.	Job Displacement	Human roles in customer service may diminish.
Cost Savings:	Companies reduce HR costs as chatbots handle more customers.	Limited Problem Solving	Chatbots can't address all issues, leading to customer dissatisfaction.
Quick Responses	Chatbots swiftly address queries, minimizing wait times.	User Familiarity	Users may struggle with appropriate communication.
Expanded Reach	Chatbots interact with multiple customers simultaneously at, anytime.	Low AI Acceptance	Some remain sceptical of AI's usefulness.
Enhanced Interaction	Chatbots foster personal connections and ease of communication.	Rising Competition	Advancements by other developers may outpace current technology.
Error Reduction	Chatbots minimize human errors.	System Vulnerabilities	Chatbots are susceptible to system crashes.

Opportunities	Threats
Trends in digital conversations, long-term investment, customer satisfaction acceleration.	Low technology acceptance, competition from evolving AI technology.
Increased chatbot usage, service personalization, and urban living standards.	Low technology acceptance, competition from evolving AI technology.
Demand for faster and more sophisticated solutions.	

Chatbots exemplify streamlined technology use while surpassing human effectiveness, potentially transforming into adept information tools. They yield significant customer service savings and, as AI advances, could mirror human interactions. This research enhances understanding for users and developers, with potential future work exploring platform specifics and ethical considerations, navigating the evolving landscape of chatbot-human interaction.

Gamification

Gamification emerged in the early 2000s and, after a decade of being less prominent, regained traction around 2010, attracting attention from researchers across various fields (Sardi, et al., 2017). The core concept of gamification involves transplanting elements of game design from gaming to non-gaming contexts, aiming to replicate the motivation found in video games (Deterding, et al., 2011). It entails applying game elements like points, scoring, competition, and rules to different areas to encourage engagement and interaction (Hsu & Chen, 2018).

Gamification is essentially an online marketing strategy aimed at fostering user engagement with products or services (Deterding, et al., 2011). Yudhoatmojo and Ramadana (2016) elaborate that its purpose is to enhance customer satisfaction, motivation, and attachment to a product or service by incorporating game-like elements. Bakker and Demerouti (2007) define gamification as implementing gaming mechanics in non-game activities to influence behavioural changes. Hsu and Chen (2018) further describe it as the integration of game dynamics, and potentially new game mechanics, into websites, business services, online communities, or marketing campaigns to drive participation and engagement within a business context.

Gamification serves to support and inspire users to complete specific tasks, transforming these activities into more engaging experiences (Koivisto & Hamari, 2014). Ultimately, it seeks to infuse a sense of playfulness into participants' interactions with a given activity, creating an environment conducive to increased engagement (Koivisto & Hamari, 2014).

Examples of Gamification Techniques in E-commerce

Incorporating gamification techniques into e-commerce platforms has become an innovative approach to enhancing user engagement and driving desirable behaviours. This section highlights various real-world examples where gamification has been effectively employed by e-commerce businesses to create immersive experiences, foster customer loyalty, and boost sales. These instances illustrate the diverse ways in which gamification can be harnessed to achieve tangible outcomes within the digital retail environment. *Table 6* displays various common examples of gamification techniques used by various e-commerce websites to engage users and encourage specific behaviours.

Table 6: Displaying examples of gamification techniques used by various e-commerce websites.

Website	Gamification Example	Purpose
Amazon	"Amazon Prime" loyalty program with badges and rewards for frequent purchases.	Encourages customer loyalty and repeat purchases.
Nike	"Nike Run Club" app tracking runs and rewarding users with achievements and badges.	Enhances brand loyalty and motivates users to stay active.
eBay	Auction-style listings with countdown timers, creating a sense of urgency for bids.	Increases competition and urgency to boost bidding activity.
Swiggy	"Swiggy Super" subscription with discounts and free deliveries, promoting loyalty.	Encourages customers to use the platform more frequently.
McDonald's	McDonald's Monopoly" game offers chances to win prizes with food purchases.	Increases sales and excitement among customers.
Sephora	"Beauty Insider" program with tiers and points for purchases, leading to exclusive discounts.	Increases customer engagement and incentivizes more spending.

SWOT Analysis of gamification in E-Commerce

Conducting a SWOT analysis provides a comprehensive framework to evaluate the strengths, weaknesses, opportunities, and threats associated with implementing gamification in e-commerce. This analysis enables a deeper understanding of the internal and external factors that can impact the success and sustainability of gamification strategies within the e-commerce landscape. *Table 7* represents a SWOT Analysis of gamification.

Table 7: Displaying a SWOT analysis of gamification.

Positive		**Negative**	
Strengths	**Description**	**Weaknesses**	**Description**
Enhanced User Engagement:	Gamification increases user participation and time spent on platforms, as interactive elements and rewards stimulate interest.	Implementation Challenges	Poorly executed gamification efforts may lead to confusion or disengagement among users.
Behavioural Change	It effectively motivates users to perform desired actions or behaviours through intrinsic and extrinsic rewards.	Dependency on Rewards	Users may lose interest if rewards are not consistently provided, leading to decreased engagement.
Data Collection	Gamification provides valuable user data, enabling personalized marketing and improved targeting.	Resistance to Change	Some users may resist gamified elements if they disrupt the original purpose of the platform.
Brand Loyalty	Successful gamification strategies can foster emotional connections and brand loyalty among users.	Lack of Long-term Impact	Gamification might generate short-term engagement spikes but fail to sustain long-term interest.
Learning and Skill Development	It can be used for educational purposes, encouraging users to learn and develop skills while interacting.	Complexity	Designing effective gamification strategies requires a deep understanding of user behaviours and preferences.

Opportunity	Description	Threats	Description
Market Growth	As the digital landscape expands, there are more opportunities to incorporate gamification in various industries.	Saturation	As more platforms adopt gamification, users may become overwhelmed, causing engagement fatigue.
Technological Advancements	Emerging technologies like AR/VR can enhance gamification experiences, providing unique opportunities.	Privacy Concerns	Collecting user data for gamification purposes can raise privacy and security issues.
Personalizatio n	Gamification can be tailored to individual preferences, leading to more engaging and relevant experiences.	Negative User Experience	Poorly designed gamification can annoy or frustrate users, leading to a negative perception
Behavioral Analytics	Advanced data analysis can provide insights into user behaviours, aiding in refining gamification strategies.	Evolving User Preferences	Users' preferences and expectations may change over time, requiring continuous adaptation.
Global Reach	Online platforms allow gamified experiences to reach a wide and diverse audience.	Competition	Rival companies implementing gamification strategies can intensify competition for user attention.

Recommender System

With the rapid expansion of the internet and smart devices, e-commerce platforms like Alibaba and Amazon have become increasingly convenient and prevalent. However, the extensive variety of products available on these platforms creates a challenge for customers in finding suitable items from the vast selection. This challenge can lead to reduced customer interest and lower sales for businesses. To address this issue, recommender systems play a crucial role in enhancing both customer experience and business performance (Hussien, et al., 2021).

A recommender system is a software tool that automates the process of suggesting products to customers based on their preferences. These systems collect information from customers and analyze it to generate recommendations that align with the customer's tastes and needs. While many existing systems rely solely on purchasing information, modern recommendation systems consider various factors beyond just past purchases (Hwangbo, et al., 2018). Some types of recommender systems are:

Collaborative Recommender System: Collaborative Filtering (CF) is a prevalent technique in personalized recommendation systems, relying on the similarity of user preferences for future suggestions (Iwanaga, et al., 2019). CF algorithms utilize past user ratings to predict and propose new items for specific user groups (Alhijawi & Kilani, 2020). This approach identifies neighbours for each user, aiding in generating recommendations that align with their interests (Nilashi, et al., 2013). Amazon effectively employs CF for product recommendations (Nilashi, et al., 2013). Categorized as user-based and item-based, CF methods analyze the user-item matrix for insights (Deng, et al., 2019). Challenges such as the cold start problem and scalability have driven advanced recommendation methods. Despite their success, traditional CF algorithms can be computationally demanding, especially with large datasets, sparking ongoing research for optimization (Li, et al., 2018).

Content Recommendation Systems: Content recommendation systems suggest items akin to consumers' past preferences, assessing item similarity based on shared attributes. For example, a positively reviewed comedic book might lead to recommendations from the comedy genre. These systems also utilize user-specific categories and employ product attributes to build classifiers for predicting preferences. Lin and Jingtao (2015) introduced contextual data like clicks, accesses, purchases, and reads to gauge item favorability, enhancing recommendations. Their approach's key contributions encompass converting user behaviour into a uniform metric, measuring product similarity using a ripple-like method, and distinguishing expendable from nonexpendable items (Lin & Jingtao, 2015). Although leveraging navigational and behavioural data addresses rating gaps, it might not consistently represent actual consumer preferences.

Demographic-Based Recommender System: Demographic-based recommenders personalize suggestions using client profiles, considering age, language, and social circles (Jabakji & Dağ, 2016). This advantageous approach doesn't require rate history, setting it apart from collaborative methods (Gujarathi, et al., 2018). Recommendations can align with clients' interests, social preferences, and influences. E-commerce platforms employ suggestions to boost sales, but fake recommendations from push or nuke attacks can impact satisfaction (Jiang, et al., 2016). To combat the cold start, location features suggest items based on similar users' interests (Ramesh & Reeba, 2017). Location-based social networks merge online-offline data, enhancing real-virtual connections (Bridge, et al., 2005).

Knowledge-Based Recommender Systems: Knowledge-based (KB) systems employ domain expertise to suggest products based on specific attributes that match client preferences and needs. Case-based algorithms, like the prominent Case-Based Reasoning (CBR), use similarity functions to gauge problem descriptions and solutions for approximating client requirements (Ricci, et al., 2006). These systems emphasize linking product attributes to client needs and often leverage user profiles (Wang, et al., 2020). Notably, Google utilizes user queries for recommendations. Wang et al. propose a personalized recommendation framework using learning clustering representation, addressing traditional limitations. While beneficial for mitigating issues like cold starts and data sparsity, KB methods might not suit smaller e-commerce platforms (Lorenzi & Ricci, 2005).

Hybrid Recommendation Systems: Hybrid recommendation systems blend multiple methods to address drawbacks and enhance performance (Zahir, et al., 2019). By integrating collaborative and content-based approaches, these systems aim to mitigate the limitations of each. The hybrid approach is adaptable based on domain and data characteristics, leading to various hybrid recommendation classes, including augmentation, combination, mixed, switching, weighted, meta-level, and cascade systems (Ramesh & Reeba, 2017). A 2018 study by P. Kumar et al. introduced a graph-based system merging two collaborative filtering strategies, achieving a high accuracy rate of 97.2% using Stanford SNAP datasets (Kumar & Reddy, 2018). However, this accuracy pertains specifically to the SNAP dataset.

Table 8 displays companies that use various types of recommendation systems along with the benefits they gain. These companies utilize different recommendation system types to enhance their services and provide more relevant and personalized suggestions to their users.

Table 8: Displaying companies that use various types of recommendation systems along with the benefits they gain.

Type of Recommendation System	Company	Benefits
Collaborative Filtering	Netflix, Amazon	Enhanced user satisfaction and increased sales through personalized suggestions based on user behaviour and preferences.
Content-Based	Spotify, YouTube	Improved user engagement and retention by recommending relevant content based on user preferences and content attributes.
Knowledge-Based	Pandora, Last. fm	Enhanced music discovery and user satisfaction by suggesting songs and artists aligned with user preferences and music attributes.
Hybrid	TripAdvisor, Airbnb	More accurate and diverse recommendations, leading to increased user engagement, satisfaction, and business growth.

Table 9 displays a SWOT analysis of recommender systems. This SWOT analysis provides a comprehensive overview of the strengths, weaknesses, opportunities, and threats associated with recommender systems.

Table 9: Displaying a SWOT analysis of recommender systems.

Positive		**Negative**	
Strengths	**Description**	**Weaknesses**	**Description**
Personalization	Recommender systems provide personalized suggestions that enhance user experience and engagement.	Data Dependency	Recommender systems heavily rely on the quality and quantity of available data, which can lead to potential inaccuracies.
Personalization	Recommender systems provide personalized suggestions that enhance user experience and engagement.	Cold Start Problem	Recommending to new users or items with sparse data can be challenging, known as the cold start problem.
Efficient Automation	Automation in recommending saves users time and effort in discovering relevant items.	Limited Interpretability	The decision-making process of recommendations may lack clear interpretability.
Versatility	Recommender systems can handle various types of data, such as ratings, preferences, and behaviour.	Vulnerability to Attacks	Recommender systems can be vulnerable to manipulation and attacks, impacting recommendation quality.
Opportunity	**Description**	**Threats**	**Description**
AI Integration	Recommender systems can integrate with AI and machine learning advancements for improved accuracy.	Privacy Concerns	Privacy concerns and regulations can affect data collection and usage, impacting system performance.

Diversification	Recommender systems can expand into new industries like healthcare and education for broader applications.	Competition	Intense competition can result in similar recommendations across different platforms, diminishing uniqueness.
Social Network Data	Utilizing social network data presents opportunities for enhanced recommendations.	User Resistance	Users might resist automated suggestions, leading to dissatisfaction and loss of serendipity.
Ethical Consideration	Recommender systems can fine-tune algorithms to address ethical and societal implications.	Technological Evolution	Rapid technological changes necessitate continuous updates and adaptation to remain effective.

In conclusion, as Albert Einstein once said, "The only source of knowledge is experience." Recommender systems serve as the bridge between users and relevant content, shaping experiences and enhancing engagement. Embracing collaborative, content-based, knowledge-based, and hybrid approaches, these systems drive sales, while challenges like data quality and privacy underline the need for continual refinement. In a world flooded with information, these systems illuminate the path towards personalized and meaningful interactions.

Responsive Web Design

Responsive web design is a design approach that ensures a website's optimal display across various devices and screen sizes. It involves creating a single design that adapts and rearranges its layout and content based on the user's device, whether it's a desktop, tablet, or smartphone. This adaptability is achieved using flexible grids, media queries, and CSS techniques. By providing a consistent and user-friendly experience regardless of the device, responsive web design enhances user satisfaction and engagement (Marcotte, 2010).

In the context of the excerpt, responsive web design aligns with the idea of embracing the web's flexibility. Just as the web lacks the constraints of print, responsive design allows websites to dynamically adjust to different screen sizes, accommodating the ever-changing ebb and flow of user devices and preferences.

Responsive web design has become crucial due to the prevalence of mobile devices. In 2021, there were 7.1 billion mobile users worldwide, with around 6.5 billion being smartphone users, and this number is expected to rise (Statista, 2021). Mobile devices generated 54.4% of global website traffic in 2021 (Statista, 2022), and internet users spent 50% of their online time on mobile devices (DataReportal, 2021). Moreover, 55.4% of internet users used mobile phones for online shopping, and 53.8% of web designers considered responsive design essential. Mobile optimization led to a 48% increase in mobile website shopping during the COVID-19 pandemic. Speed is key as even a 0.1-second improvement resulted in an 8% increase in conversions and an 8% bounce rate improvement. Approximately 15 billion mobile devices were connected to 5G in 2021, indicating the growing significance of fast connections (Statista, 2022). *Figure 11* displays the forecast number of mobile users worldwide from 2020 to 2025 (in billions).

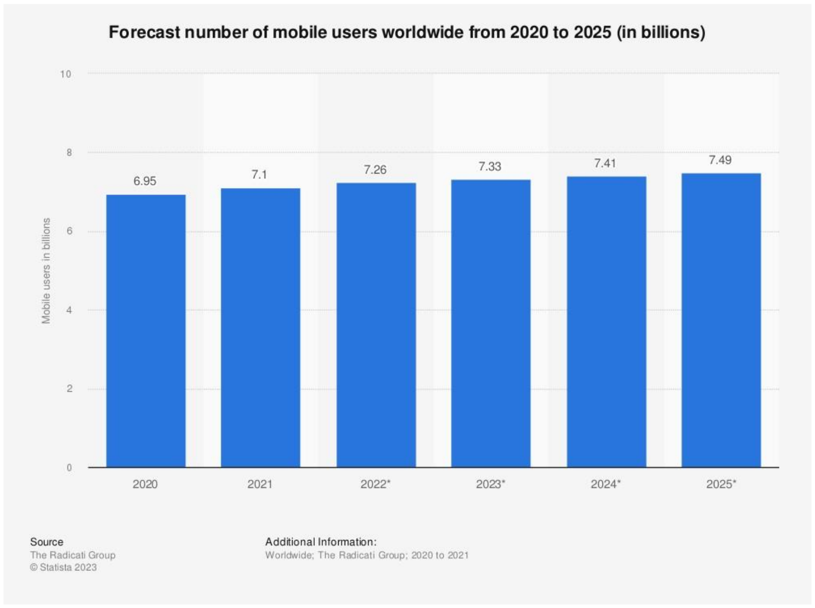


Figure 11: Displaying forecast number of mobile users worldwide from 2020 to 2025 (in billions).

Table 10 represents different companies that use responsive web design and their benefits.

Table 10: Displaying different companies that use responsive web design and their benefits.

Company	Benefit
Apple	Consistent user experience across devices
Amazon	Improved mobile shopping experience
Google	Better accessibility for diverse users
Spotify	Seamless music streaming across devices

Responsive web design helps these companies provide a seamless and user-friendly experience across various devices, leading to improved engagement, conversions, and customer satisfaction.

Table 11 displays a SWOT analysis of responsive web design. This SWOT analysis provides a comprehensive overview of the strengths, weaknesses, opportunities, and threats associated with responsive web design.

Table 11: Table displaying SWOT analysis for responsive web design.

Positive		**Negative**	
Strengths	**Description**	**Weaknesses**	**Description**
Consistent User Experience	Responsive web design ensures a uniform and seamless user experience across various devices, enhancing user satisfaction and loyalty.	Initial Development Time	Implementing responsive design can initially take longer due to the need for careful planning and coding.
Improved Mobile Accessibility	It caters to the increasing mobile usage trend, making websites accessible and user-friendly on smaller screens.	Performance Concerns	Complex designs and heavy content can impact website performance, leading to slower loading times.
Broad Device Compatibility	Responsive design allows websites to function well on a wide range of devices, reducing the need for separate designs for each platform.	Complex Design Implementation	Designing responsive layouts for different devices can be intricate and challenging, requiring skilled designers.
Opportunity	**Description**	**Threats**	**Description**
Growing Mobile Usage	As mobile usage continues to rise, responsive design can capitalize on this trend to reach a larger audience.	Rapid Technological Changes	Ongoing technological advancements may necessitate constant updates and adjustments to maintain compatibility.

Enhanced User Engagement	By providing an optimal user experience, responsive websites can increase user engagement and interaction.	Browser Compatibility Issues	Variations in browser behaviour can pose challenges to ensuring consistent rendering and functionality.
Higher Conversions	Improved user experience and accessibility can lead to higher conversion rates and improved business outcomes.	Intense Market Competition	With many websites adopting responsive design, staying competitive requires continuous improvement and innovation.

Responsive web design offers consistent user experience and improved accessibility on mobile devices, capitalizing on the increasing mobile usage trend. It provides opportunities for higher user engagement, conversions, and wider audience reach. However, there are challenges such as longer initial development times, potential performance concerns, and the complexity of design implementation. Rapid technological changes and browser compatibility issues pose threats to its effectiveness. Additionally, the intense competition in the market may demand constant updates and improvements to stay ahead.

Responsive web design is a pivotal approach in the digital landscape, catering to the ever-growing mobile user base. It ensures a consistent and user-friendly experience across devices, leveraging the opportunities presented by increasing mobile usage. While initial challenges like development time and design complexity exist, the benefits of improved accessibility, engagement, and conversions underscore its importance in creating effective and competitive online platforms.

Project Related Choices

In this section, we will explore additional aspects that extend beyond the foundational principles of e-commerce and delve into project-related decisions. Specifically, we will delve into considerations such as the selection of software tools for constructing the application. This exploration aims to provide a comprehensive background research to inform and guide the project's implementation.

Software Development Tools

The incorporation of software development tools will significantly enhance the project's efficiency and outcomes. By utilizing these tools, the development process becomes more organized, allowing for systematic coding, seamless debugging, and efficient version control (Sommerville, 2016). Collaboration among team members becomes smoother, fostering effective communication and teamwork (Dybå & Dingsøyr, 2008). Moreover, these tools offer features such as automated testing and deployment, ensuring the application's reliability and stability (Bass, et al., 2012). This integration of software development tools into the project not only expedites the development timeline but also guarantees a well-structured, high-quality end product that aligns precisely with the project's goals and requirements (Pressman & Maxim, 2014).

In assessing the landscape of software development tools for this project, a variety of options were explored to ensure the most suitable choices were made. Alongside the considered tools of CSS, HTML, JavaScript, XAMPP, Apache, MySQL, PHP, PhpMyAdmin, and Visual Studio Code, alternatives such as React, Angular, Node.js, Django, and Ruby on Rails were also evaluated.

After careful evaluation, the decision was reached to opt for CSS, HTML, and JavaScript for front-end design due to their flexibility and widespread compatibility. For back-end development and database management, XAMPP, Apache, MySQL, and PHP were chosen for their robust capabilities and seamless integration. PhpMyAdmin's intuitive interface was selected for efficient database management, and Visual Studio Code was embraced as the integrated development environment due to its powerful features and community support.

The chosen tools collectively align with the project's goals of creating a dynamic and user-friendly web application. Their extensive industry adoption, ease of use, and comprehensive functionalities made them the ideal choices for developing a cohesive and high-performance web application. By harnessing the capabilities of these selected tools, the project aims to achieve a successful and optimized outcome that fulfils its objectives effectively.

All the chosen tools are open-source and available for free, which aligns with the project's goal of cost-effective development while maintaining a high standard of quality and functionality.

Version Control System

Version control is crucial for the project as it offers systematic management of code changes, collaborative development, and safeguarding against errors. It allows tracking and documenting modifications made to the project's source code over time, making it easier to identify and fix issues (Chacon & Straub, 2014).

The project will benefit significantly from using version control by promoting efficient collaboration among team members. Version control systems like Git enable multiple developers to work concurrently on different aspects of the project while maintaining a centralized repository that synchronizes changes (Dabbish, et al., 2012). This prevents conflicts, facilitates seamless integration of changes, and enhances the overall development process.

Furthermore, version control ensures a robust backup mechanism, preserving different versions of the codebase. This not only guards against accidental data loss but also offers a convenient means of reverting to previous stable states if unforeseen issues arise (Loeliger & McCullough, 2009).

Incorporating version control also aligns with best practices in software development, fostering a more organized and transparent workflow. It enhances accountability by attributing code changes to specific contributors and assists in reviewing and managing proposed alterations before merging them into the main codebase (Sommerville, 2016).

In the exploration of version control tools, several options were considered, including Git, Mercurial, and Subversion. After careful evaluation, Git was chosen as the version control tool for the project, coupled with the hosting platform GitHub.

Git was selected due to its distributed nature, efficiency in handling both small and large projects, and its popularity within the software development community (Chacon & Straub, 2014). Its decentralized architecture allows each team member to have a local copy of the entire code repository, which enhances collaboration, reduces dependency on a central server, and provides resilience against server failures.

GitHub was chosen as the hosting platform for its user-friendly interface, powerful collaboration features, and widespread adoption. It simplifies remote repository management, facilitates code review, and enables seamless integration with other development tools (Dabbish, et al., 2012).

Both Git and GitHub are open-source tools and are available at no cost, making them accessible and cost-effective choices for version control and project management.

Software Development Methodology

A Software Development Methodology is crucial for the project as it provides a structured framework for planning, executing, and managing the development process. It helps ensure that the project progresses systematically and efficiently, leading to successful outcomes.

The project will benefit from using a Software Development Methodology in several ways. Firstly, it enhances project organization and communication, allowing all team members to be on the same page regarding tasks, timelines, and goals (Pressman, 2014). Secondly, it mitigates risks by identifying potential challenges early on and implementing strategies to address them (Sommerville, 2015). Thirdly, it aids in managing resources effectively, optimizing time and effort allocation. Lastly, a methodology like Agile promotes adaptability, allowing adjustments based on feedback and changing requirements (Cohn, 2010).

By adhering to a Software Development Methodology, the project can avoid common pitfalls, streamline processes, maintain clear documentation, and foster a collaborative environment.

In the context of software development methodologies, various options were considered, including Waterfall, Agile, Scrum, and Kanban. After careful evaluation, the project chose the Agile methodology due to its recognized flexibility, adaptability, and collaborative approach (Beck, et al., 2013). Agile was selected to align with the project's dynamic nature, fostering iterative development, continuous feedback, and incremental enhancements (Highsmith, 2002).

The Agile methodology, as chosen, emphasizes customer satisfaction, swift delivery of functional software, and the ability to accommodate evolving requirements (Cohn, 2009). It fosters frequent communication among the development team and stakeholders, enhancing transparency and reducing potential misunderstandings (Schwaber & Sutherland, 2012).

The cost associated with implementing Agile can vary based on specific practices and tools adopted, typically offering cost-effectiveness due to its emphasis on value-driven outcomes and efficient resource utilization (Conforto, et al., 2016).

Summary

In this section, a comprehensive exploration of various facets of e-commerce and the integration of emerging technologies has been conducted. The evolution and impact of e-commerce, along with its advantages and limitations, were examined to provide a foundational understanding. The integration of AI, gamification, responsive design, and recommender systems were analyzed, showcasing their potential to enhance user experiences, streamline interactions, and boost overall business outcomes.

AI-driven personalization offers tailored user experiences through customized recommendations and interactions. Gamification introduces interactive elements to engage users and enhance their involvement. Responsive web design ensures optimal user experiences across devices, facilitating seamless navigation. The recommender system utilizes data to suggest relevant products, enhancing customer satisfaction and cross-selling opportunities.

These insights serve as the groundwork for the subsequent recommendation section, where strategic decisions will be outlined based on the outcomes of this comprehensive research.

Recommendations

The specifics of how the functionality mentioned in Step 4 will be implemented in the application are going to be covered in this section.

Chatbot

The chatbot feature enhances user interaction and support. When users have inquiries, the chatbot provides instant responses, guiding them through the shopping process. This real-time assistance ensures users find the products they need, leading to more successful purchases. The chatbot's availability 24/7 adds convenience, eliminating the need to wait for business hours. Furthermore, the chatbot can offer personalized recommendations based on user preferences and purchase history, creating a tailored shopping experience. This personalized engagement not only boosts customer satisfaction but also increases the likelihood of repeat purchases and customer loyalty.

Responsive Web Design

Responsive web design ensures seamless user experience across devices. Users access the platform on smartphones, tablets, or desktops, and the design adapts accordingly. This adaptation enhances accessibility and usability, allowing users to navigate and interact effortlessly. Whether shopping on a small phone screen or a large desktop monitor, the interface remains visually appealing and user-friendly. This approach fosters customer satisfaction and encourages extended engagement, ultimately boosting conversion rates and fostering brand loyalty.

Recommender System

The recommender system will enhance the user experience by offering personalized product recommendations based on their preferences and behaviours. This feature will save users time by presenting them with products that align with their interests, increasing the likelihood of finding items they'll love and potentially boosting their engagement and satisfaction with the application.

Gamification

The gamification feature introduces a VIP Club with bronze, silver, and gold packages. Users can join by creating an account and making their first purchase, earning 50 points. As they shop, they accumulate points based on their spending level. For bronze, spending up to $250 earns 1 point per dollar; for silver, spending up to $1000 earns 1.25 points per dollar; for gold, spending up to $1500 earns 1.5 points per dollar. These points can be redeemed for exclusive discounts, encouraging users to engage more, shop frequently, and enjoy tailored rewards based on their spending preferences.

Requirements Analysis

Introduction

The first phase of the software development life cycle, requirement analysis, focuses on meticulous documentation and comprehension of client needs. Its essence can be summed up by Stephen Covey's saying, "Begin with the end in mind." This chapter delves into defining the project's scope, objectives, and functional and non-functional requirements. To ensure clarity in subsequent stages, personas are created to represent typical users, with use cases detailing interactions and information architecture displayed through sitemaps and activity diagrams.

The Process of Producing the Requirements

A diverse strategy was used during the project's requirements collecting phase to achieve full knowledge and implementation of critical aspects. Initially, tremendous documentation research was conducted, including assignment briefing sheets and module units. This provided a foundational grasp of the project's academic and conceptual foundations, which guided the initial design of the software artefact.

A thorough analysis of pertinent applications was conducted, with a focus on major e-commerce sites like Amazon and eBay. The popularity and simplicity of these platforms are largely due to the functionality and design elements that this research helped to establish. The user interface and experience of the project were developed in large part thanks to the ideas gathered from this comparison analysis. The design and operation of the product catalogue, search, and filtering capabilities were guided by insights from using this approach.

The StudyNet platform facilitated interviews with academic professionals, which provided crucial expert insights and recommendations. Supervisory and other academic professionals provided a distinct perspective, especially in integrating the project with current academic standards and expectations. Using this approach, the need for strong analytics and administrative tools was underlined, which prompted the development of an advanced admin dashboard with data visualization capabilities.

During this stage, background research was essential and covered a wide range of subjects, from user behaviour on digital platforms to technological advances in software development. To guarantee that the project was not only technically solid but also relevant and adaptable to both present and emerging market trends, this research was crucial. A recommendation engine for individualized user experiences and a chatbot for improved user engagement were identified by the review of current technology developments.

A major strategy used was requirement gathering based on personas. The project was able to concentrate on the unique requirements, preferences, and behaviours of various user groups by creating thorough user personas. This method made sure that the features were user-centric in addition to being theoretically and technically sound, which improved the end product's overall usability and efficiency. Using this approach developing thorough user personas made it easier to modify features to suit the unique requirements and preferences of various user groups, improving the user experience as a whole.

Each of these approaches made a substantial contribution to the development process, guaranteeing that the final software artefact met industry and academic standards.

Personas

A user persona is defined as a dynamic and inferred profile of a user based on their chat history and interactions. It includes the user's qualities, preferences, and behavioural patterns, which are not expressly mentioned but are determined via an analysis of their communication style and content. This inferred persona is then utilized to modify responses in a dialogue system, ensuring that the interaction is individualized and resonates with the unique features of each user.

User personas are extremely important in design and development processes. User personas are a fundamental tool for comprehending and connecting with the target audience. Designers and developers can gain insights into their users' needs, preferences, and behaviours by developing comprehensive, imaginary representations of key user groups. This user-centric approach ensures that products, services, and circumstances are not only practical but also have a strong emotional connection with the intended audience. The usage of personas assists in making informed decisions throughout the design process, aligning the team's focus, and avoiding the risks of developing based on assumptions or personal prejudices. Finally, adopting user personas results in more engaging, relevant, and successful results that fulfil the requirements and expectations of end users.

The following are some of the advantages of user persons:

- Enhanced Understanding of User Needs: User personas assist designers and stakeholders in gaining a better knowledge of the needs, interests, and behaviours of the target users.
- Improved Design Relevance: Designs can be better adapted to match the needs and expectations of the end users by concentrating on particular user groups.

- Effective Communication Tool: In design teams and with clients, personas are a helpful communication tool that helps to bring everyone's understanding of the target user into alignment.
- Guidance for Decision Making: Personas serve as a point of reference when making design decisions, ensuring that these decisions are user-centric.
- Increased Engagement and Satisfaction: Persona-based designs are more likely to resonate with people, leading to improved engagement and pleasure.

Tables 12 and 13 show an illustration of the personas established to provide an insight into the typical customer of online health and wellness stores.

Table 12: Displaying User Persona One.

	Name: Emily Johnson Age: 29 Job: Marketing Specialist Working Hours: Full-time, typically 9 am to 5 pm Hobbies: Yoga, cooking healthy meals, and reading wellness blogs Emily Johnson, 29, is a marketing specialist with a passion for health and wellness. Balancing a busy work life with yoga and cooking, she relies on online shopping for quality health products. Efficient, informative, and user-friendly e-commerce experiences are essential to her.
Needs and Desire	**How this system helps**
Emily has a busy schedule and prefers shopping online for health products.	The platform offers a user-friendly interface with easy navigation, meeting Emily's need for convenience.
She looks for a wide range of quality health supplements and wellness products.	Detailed product pages and user reviews help her make informed purchases.

Needs detailed product information to make informed choices.	Efficient search and filter functions allow her to quickly find the products she needs.
Relies on user feedback to assess product effectiveness.	The secure and streamlined checkout process saves her time.
Prefers a hassle-free payment and checkout process.	

Table 13: Displaying User Persona two.

	Name: David Smith Age: 35 Job: Personal Fitness Trainer Working Hours: Irregular, often early mornings, and late evenings Hobbies: Weightlifting, hiking, and blogging about fitness David Smith, 35, is a dynamic fitness trainer with a busy schedule. He's constantly seeking the latest in fitness supplements and enjoys platforms that offer personalized recommendations and bulk ordering. David appreciates interactive and gamified e-commerce experiences that fit his active lifestyle.
Needs and Desire	**How this system helps**
Wants to stay updated with the latest in fitness supplements.	The recommendation engine suggests products based on his browsing and purchase history.
Often orders products in bulk for his training facility.	The order history feature allows David to easily track and manage his expenses.

Looks for products tailored to his fitness regime.	Gamification features engage him more deeply with the platform, offering rewards and challenges related to fitness products.
Needs to keep track of past purchases for budgeting.	Bulk order and easy reordering options cater to his professional needs.
Enjoys engaging with interactive and gamified platforms.	Responsive design ensures he can access the site conveniently from any device, matching his on-the-go lifestyle.

The Requirements

The requirement analysis stage is the first level of the software development life cycle. This phase's primary purpose is to accurately document and comprehend the client's actual requirements. The purpose of requirement analysis is to determine what the system requires. This is one of the most vital phases of the software development life cycle. The outcome of requirement analysis, the Software Requirement Specification, is a complete description of how the software to be produced should operate (Dora & Dubey, 2013).

In software development, requirement analysis is critical for ensuring that client needs are fully understood, preventing scope difficulties, and aligning expectations. It's the compass that helps initiatives succeed by reducing risks and optimizing resources. As Stephen Covey (2004) wisely said, 'Begin with the end in mind.' This quote encapsulates the importance of requirement analysis succinctly, emphasizing the significance of starting a project with a clear grasp of the desired outcome.

During the requirement analysis, a thorough grasp of the project's scope and objectives will be attained. This step entails unravelling numerous aspects, including the process of developing requirements, which outlines how the project's demands will be transformed into criteria. Personas will be established to represent typical users, assisting in the design of solutions that meet their needs. The system's intended functions will be detailed in functional requirements, while non-functional requirements such as performance and security will be identified. To capture interactions between users and the system, use cases will be developed. The project's content will be structured by information architecture, which will be shown as a site map, and system operations will be visually depicted by activity diagrams, improving clarity in design and implementation.

What distinguishes a functional requirement from a non-functional requirement, and how can one know the difference?

The system's functional requirements outline its intended functions. It includes the features and services that the system is intended to offer. While non-functional requirements specify the qualities of the system, like security and performance, and how the system is to operate. In conclusion, the primary distinction between functional and non-functional requirements is that functional requirements describe a system's specific functions and behaviour, whereas non-functional requirements describe the system's quality attributes and characteristics.

Functional Requirements

Table 14 presents a comprehensive overview of the software artefact's Core and Advanced Functional Requirements.

Table 14: Displaying the software artefact's Core and Advanced Functional Requirements.

No.	Requirements	Requirement Type	Access
FR1	Registration	Core	Public
FR2	Login	Core	Public
FR3	User Profile Management	Core	Registered User
FR4	Product Catalogue	Core	Public
FR5	Search Functionality	Core	Public
FR6	Product Filtering	Core	Public
FR7	Product Detail Page	Core	Public
FR8	Add to Cart	Core	Registered User
FR9	Shopping Cart Management	Core	Registered User
FR10	Checkout Process	Core	Registered User
FR11	Payment Integration	Core	Registered User
FR12	Order Confirmation	Core	Registered User

FR13	Order History	Core	Registered User
FR14	User Reviews and Ratings	Core	Registered User
FR15	Account Management	Core	Registered User
FR16	Admin Dashboard	Core	Administrator
FR17	Chatbot Integration	Advanced	Public
FR18	Recommendation Engine	Advanced	Public/ Registered User
FR19	Gamification Features	Advanced	Registered User
FR20	Advanced Analytics and Data Visualization	Advanced	Administrator
FR21	Responsive Design	Advanced	Public

The functional requirements have advanced to include a comprehensive breakdown that goes into great detail about each requirement. This includes an overview of the inputs, validation, output, and how the requirement functions, along with the logic behind implementing each functionality. This will help with determining the best way to design the user interface and which database tables will be needed to store the data. The breakdown of the first requirement is shown in the table below (See Table 15). To view the entire list of developed requirements, refer to Appendix A for both functional and non-functional requirements.

Table 15: Displaying the breakdown of the first requirement.

Requirement	**Registration**
Requirement Number	FR1
Description	To facilitate purchases on our platform, consumers are required to create an account, a process that is both straightforward and secure. The registration form, prominently accessible from the homepage, requires users to provide their first and last names, a valid email address, and a strong password, with an added link to the login page for previously registered users. Upon submission, the system generates a new user record in the database, assigning them a default "user-type" status and immediately informing them via a confirmation message that

	their registration was successful and they are now ready to log in and start shopping. This streamlined process ensures a user-friendly experience, guiding new customers seamlessly from account creation to accessing the full range of features available on our platform.
Rationale	A Registration form is essential on our platform as it allows users to create a personal account, critical for accessing a variety of features and services tailored to enhance their shopping experience. By providing their first and last names, a valid email address, and a secure password, users establish a secure login credential set, enabling them to access the platform confidently for future purchases. This registration process is integral not only for transactional purposes but also for unlocking an array of user-centric benefits, such as personalized product recommendations, participation in a loyalty program, and eligibility for special discounts and promotions. The inclusion of a direct link to the login page within the registration form streamlines the experience for existing users, ensuring efficiency and ease of access across the board. In essence, the registration form is a gateway for users to fully immerse themselves in the unique and personalized shopping experience our platform offers, fostering a sense of belonging and engagement from the very start.

Non-Functional Requirements

As mentioned before, non-functional requirements specify the qualities of the system, like security and performance, and how the system is to operate. Table 16 presents a comprehensive overview of the software artefact's Non-Functional Requirements.

Table 16: Displaying the software artefact's Non-Functional Requirements.

No.	Requirements	Rationale
NFR1	Consistency	Consistency across the application in terms of design, interactions, and workflows ensures a coherent user experience, reducing the learning curve for new users and enhancing overall user satisfaction.
NFR2	Maintainability	The codebase, architecture, and technologies used should allow for easy maintenance and updates. This ensures that the application can evolve, new features can be added, bugs can be fixed promptly, and the overall health of the application is preserved.

NFR3	Browser Compatibility	The application should function correctly across various browsers (Chrome, Firefox, Safari, etc.), ensuring that all users, regardless of their choice of browser, have access to a fully functional and visually consistent application.
NFR4	Usability	The application should be intuitive and user-friendly, with clear navigation and accessible features. This ensures a positive user experience, potentially leading to increased user retention and conversion rates.
NFR5	Database Security	Protecting user data is paramount. Ensuring that the database is secure from unauthorized access, injection attacks, and data leaks is critical to maintaining user trust and complying with data protection regulations.
NFR6	Application Security	The overall security of the application is crucial. Implementing security measures such as HTTPS, data encryption, and secure coding practices helps to protect the application and its users from various cyber threats.
NFR7	Scalability	The application should be designed to handle growth, whether it's an increase in the number of users, data volume, or transaction frequency. This ensures that the application continues to perform optimally as it scales, providing a reliable experience for all users.
NFR8	Performance	The application should be optimized for fast load times and smooth interactions, as performance directly impacts user satisfaction and can influence bounce rates and user retention.
NFR9	Availability	The application should be available for access at all times, minimizing downtime and ensuring that users can access the services they need whenever they need them. This is crucial for maintaining user trust and ensuring the reliability of the application.

UML Diagrams

A use case diagram is a visual aid which demonstrates how users (also known as "actors") interact with a system to accomplish particular tasks. In software development, it is an essential tool for comprehending a system's functional needs. The use case diagram clearly illustrates who will interact with the system and how by outlining the primary features of the system as well as the roles of various users. This graphic, which emphasizes how the system should behave from the viewpoint of an outside observer, is crucial in capturing the intended behaviour of the system.

Figure 12 illustrates an overview Use Case of the eCommerce application, displaying the various use cases for a new and registered user.

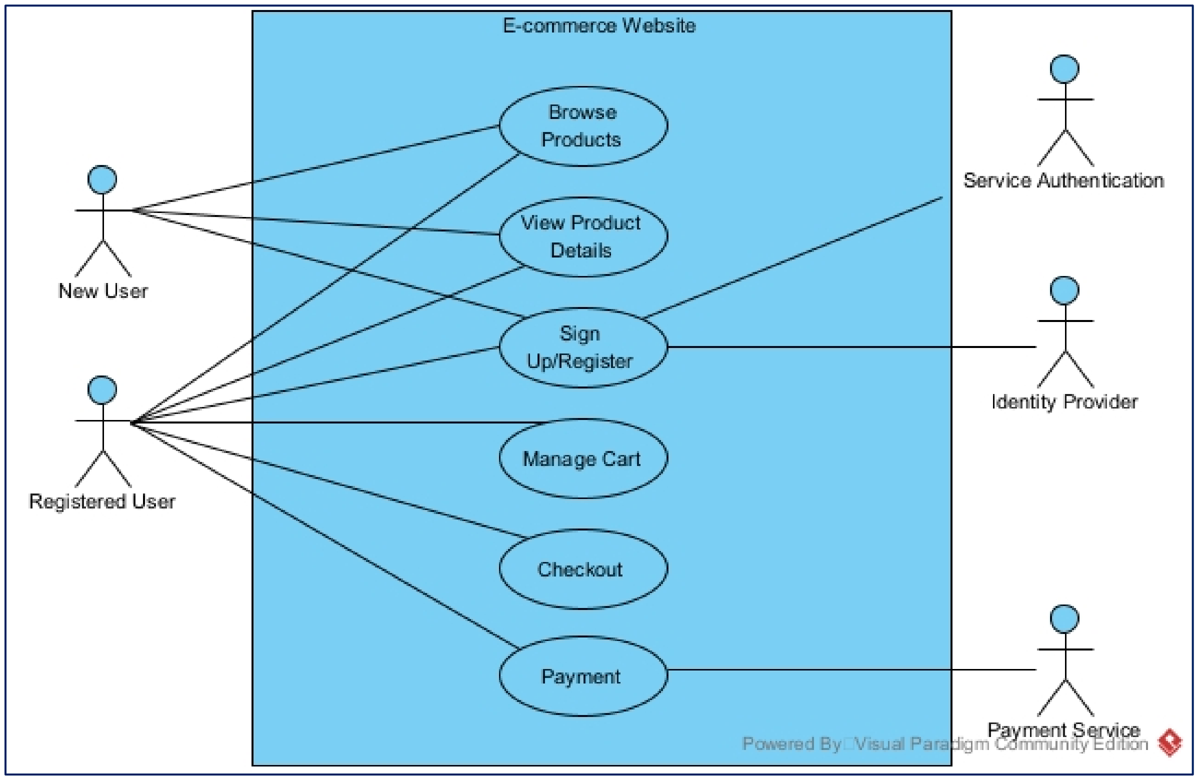


Figure 12: Displaying an overview Use Case of the eCommerce application, displaying the various use cases for a new and registered user.

Figure 12 will be dissected to provide a complete understanding of the numerous use cases related to an eCommerce application. This diagram is useful for demonstrating the variety of functionalities and interactions available to different types of users, notably new and registered users. It is a roadmap to understanding how users engage with the platform, from product browsing to ultimate purchase and beyond. We may obtain a better understanding of the user journey inside an eCommerce environment and the various touchpoints that support a seamless and secure buying experience by studying each part of this figure. Below the typical use cases for the e-commerce platform will be broken down considering different types of users:

New User:

- Browse Products: New users can browse the platform’s product catalogue. This often entails browsing through multiple categories, filtering products based on various criteria such as price, and brand and arranging the products.
- View Product Details: New users can click on specific products to view additional information in-depth. The product description, cost, availability, user reviews, and ratings are usually included in this.
- Signup/Register: New users must register if they wish to save things for later or decide to make a purchase. During the signup process, users need to enter their name, email address, and password.

Register User:

- All New User Capabilities: Registered users get access to all of the same features as new users, such as browsing and viewing product details.
- Manage Cart: Users can add items to their shopping cart for later purchase. This includes changing the quantity of the products in the cart as well as removing items from the cart.
- Checkout: When ready to make a purchase, customers can go to the checkout page and enter their billing and shipping details.
- Make Payment: Users choose a payment option such as credit/debit card, or PayPal and finish the transaction at the checkout.

Identity Provider:

- Signup/Register: Third-party services that authenticate user identities are known as identity providers. They let users sign up or register using their existing credentials from other services (such as Google, Facebook, and so on).
- Service Authentication: They ensure security and convenience by authenticating the user's identity during the signup or login procedure.

Payment Service:

- Payment Processing: The transaction is handled by a payment provider that is incorporated into the eCommerce platform. It secures consumer payments, including credit/debit card processing, and may provide many payment choices.

These use cases cover the standard functionalities of the eCommerce platform, offering a seamless and safe buying experience for all the different types of users.

The Use Case for the checkout feature has been illustrated in Figure 13:

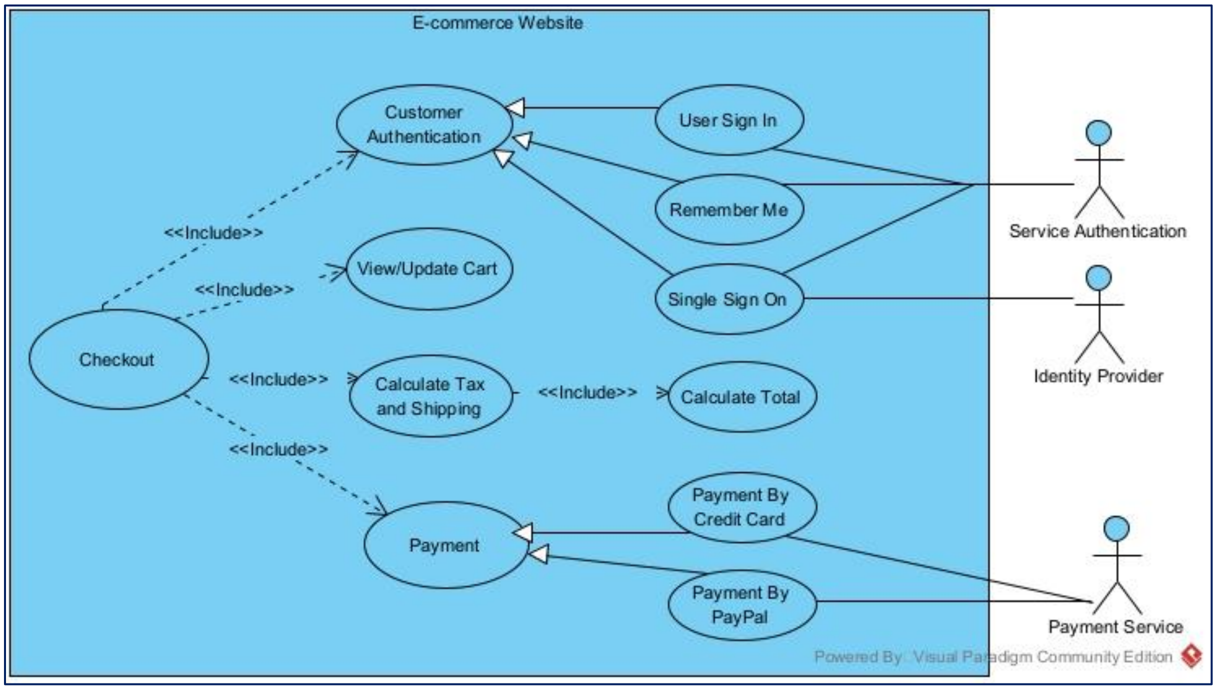


Figure 13: Displaying the Use Case for the checkout feature.

The following analysis will analyse and thoroughly examine the checkout feature's use case in an eCommerce scenario. The diagram above shows the sequence of actions and interactions that make up the checkout process. This procedure is an important part of the customer experience since it represents the final steps a user takes after confirming their order and making the payment.

Customer Authentication: The first step toward a safe checkout is customer authentication, which offers choices for both a standard login and practical features like "Remember Me."

View/Update Cart: This stage allows the buyer to finalize their decisions before advancing, ensuring complete control over their intended purchase.

Checkout: The focal point is the checkout function, which is tightly linked to the calculation of taxes, shipping, and the final total. This step is critical in moving from selection to sale.

Payment: The payment action represents the financial exchange, with several methods such as credit card or PayPal available to satisfy user preferences.

Service Authentication: Running simultaneously with the customer's checkout experience is service authentication, which is frequently enabled by an Identity Provider, especially if Single Sign-On is used.

Payment Service: Following checkout, the Payment Service secures the financial transaction and completes the purchase.

In conclusion, this use case diagram offers a well-planned and comprehensive representation of the checkout function, including every crucial stage and the corresponding actions that go along with it. By understanding every aspect of this process, we can ensure that the system is user-centric and promotes a secure and seamless transactional experience.

The use case for viewing products in the eCommerce application is depicted in Figure 14. Customers can search, and browse products, as well as view recommended products and add them to their shopping basket.

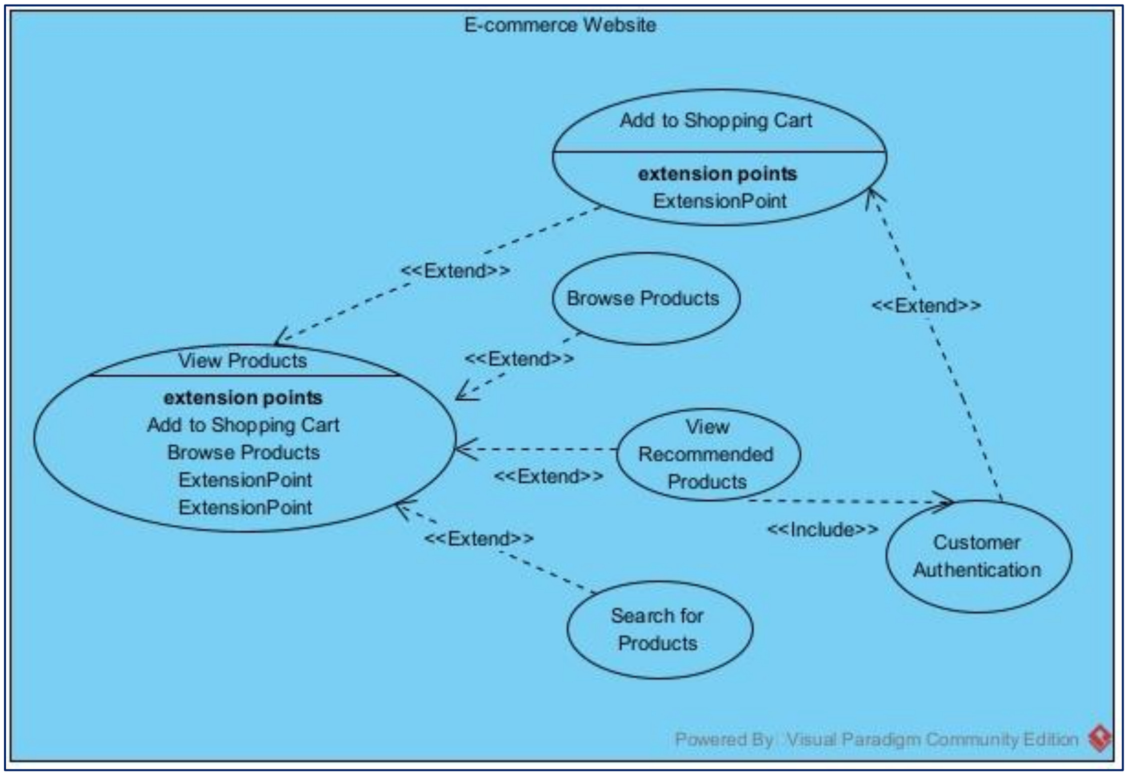


Figure 14: Displaying use case for viewing products.

The following analysis will analyse the diagram above which represents a Use Case diagram for an eCommerce website, focusing on the product-related interactions that a customer can have with the system. Here's the breakdown of the use cases depicted:

View Products: This is the primary use case, from where users begin their journey. It portrays a user browsing the products on the website.

Extension Points:

- Add to Shopping Cart: This is an extension of "View Products," which means that a user can add products to their shopping cart while viewing products.
- Browse Products: Another extension of "View Products," implying that users can browse the products more extensively than when examining individual products.

Extended Use Cases:

- Add to Shopping Cart: As previously stated, this is a possible action that a user may take when browsing products. It is an action that goes beyond simply watching.
- Browse Products: This is an activity that can lead to other activities, such as "View Recommended Products" or "Search for Products," which are alternative methods for customers to interact with the products.

View Recommended Products:

- This is an expanded use case from "Browse Products," implying that when users browse, they may be provided with products that the system recommends based on criteria such as browsing history, purchase history, or popular trends.

Search for Products: Another expanded use case from "Browse Products," this time illustrating that users can actively search for specific products (using a search bar).

Include Relationships:

- Customer Authentication: This use case has an "include" relationship with "View Recommended Products," which means that a user needs to be authenticated by the system to view personalized recommendations, which are likely to deliver tailored suggestions based on the user's past activity.

The diagram depicts a user's journey through the eCommerce website, focused on product interactions, beginning with the initial product on display and progressing through numerous exploration options such as browsing, searching, and receiving suggestions, all while being able to add products to their cart. The usage of customer authentication when seeing recommended products underlines the importance of individualized user experience.

Information Architecture

In general, information architecture refers to how information is organized, structured, and presented in digital environments. It entails the design of information environments to improve usability and findability, including features such as website navigation, classification, and overall user experience when engaging with digital systems.

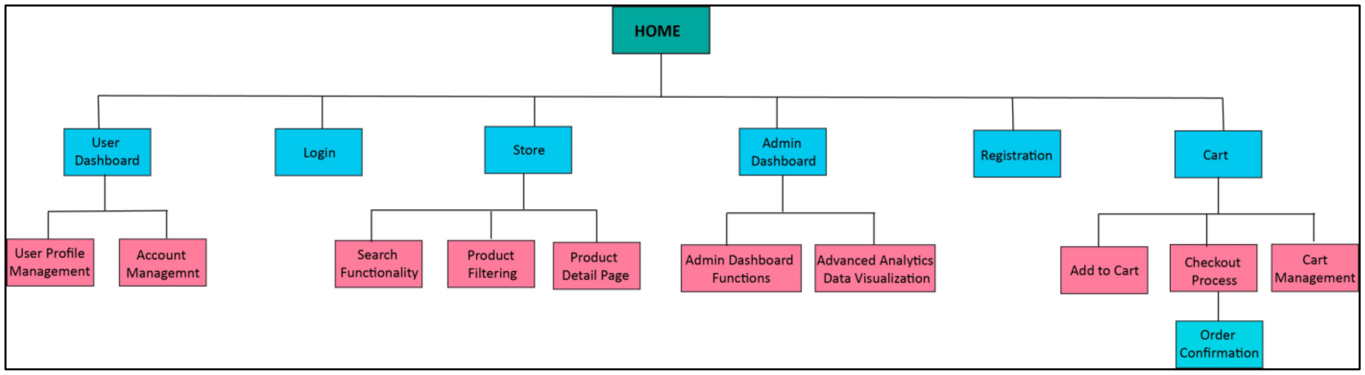


Figure 15: Displaying the application site map.

In the upcoming analysis, we will examine the sitemap displayed above. A sitemap is a web application's navigational plan. A sitemap like this one is an effective instrument that shows how different pages and features are related to one another within a website's hierarchical structure. Similar to a floor plan for an architectural building, it describes the paths people can take to explore and engage with the available content and services. For developers, designers, and content strategists to envision and organize the user's path across the website, this graphical representation is essential. We can learn more about the user experience, information accessibility, and general web application flow by carefully examining the sitemap. Let's explore the different nodes and links that form this intricate collection of interactions.

Home: This is the website's root node and beginning point.

User Dashboard: It is available after logging in and includes:

- User Profile Management: Where users can manage their personal information.
- Account Management: To manage account preferences and settings.

Login: The process or landing page where users authenticate to get access to specific sections of the website.

Store: The section where the commercial transactions occur, featuring:

- Search Functionality: To search for products.
- Product Filtering: To narrow down product choices.
- Product Detail Page: Displaying detailed product details.

Admin Dashboard: Only website administrators have access, it has:

- Admin Dashboard Functions: For general admin tasks.
- Advanced Analytics and Data Visualization: For analysing and visualizing website data.

Registration: Where prospective users may register to use the website.

Cart: This includes:

- Add to Cart: Where users add items for future purchases.
- Checkout Process: Where users finalize their purchases.
- Cart Management: For managing items in the cart.

Order Confirmation: The final stage of the ordering process, where users verify their orders.

Having a sitemap facilitates comprehension of a website's general design and user interface. Planning and structuring website content is crucial for web developers, content managers, and UX designers.

Activity Diagram

An activity diagram is an example of a UML (Unified Modeling Language) diagram used to model a system's dynamic aspects. It is simply a flowchart that depicts the flow of control from one action to the next, and it is especially effective in modelling functional activities.

Here are the key characteristics of an activity diagram:

1. **Flow of Control:** It outlines the flow of control in a system and is used to characterize the system's dynamic features. This comprises the sequence of activities as well as the conditions that cause those activities to occur.
2. **Activities**: These are the rounded rectangles that indicate the function or activities. They can be viewed as individual components of work or process operations.
3. **Transitions:** Arrows or lines that connect activities and represent the flow from one to the next.
4. **Decision Points**: Diamond shapes represent decision paths. This is the point at which you must decide on the following activity.
5. **Start and End Points:** Typically illustrated by filled circles at the beginning and surrounded by filled circles at the end.
6. **Swimlanes:** These are used to categorize activities carried out by the same actor or within the same entity.

7. Synchronization and Concurrency: They are represented by bars and denote the beginning (split) or end (join) of concurrent activity.

8. **Objects:** They can be used to demonstrate how data is transported from one operation to the next.

In the fields of business process modelling and software development, activity diagrams are commonly used to represent the sequence of operations and comprehend the flow of control in a system. They aid in the identification of possible issues and inefficiencies in a process.

Figure 16 Displays the activity diagram for the process order:

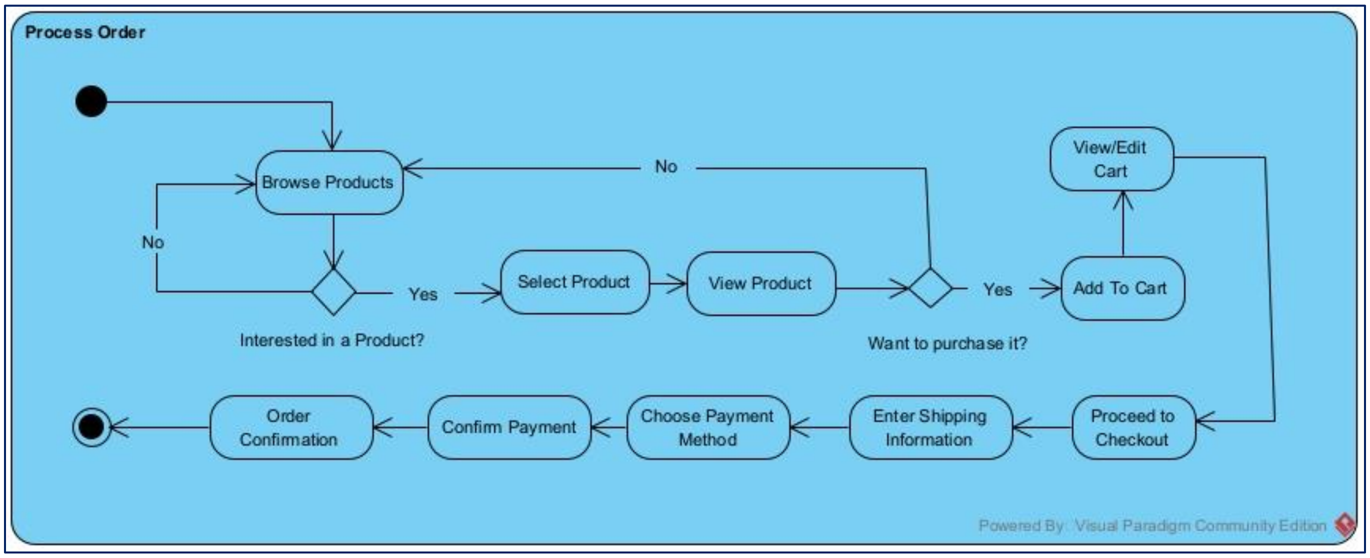


Figure 16: Displaying the activity diagram for the process order.

The image above displays an activity diagram, which is a form of flowchart used to model the workflow of a process—in this example, the e-commerce website's order process. The diagram illustrates all the steps a user takes when placing an order, from browsing products to order confirmation. Here is a thorough breakdown of each step:

1. **Start (Black Circle):** This represents the start of the order procedure.

2. **Browse Products:** This is the initial action in which a user browses the products available on the eCommerce site.

3. **Decision: Interested in a Product?** If the customer finds a product of interest, the procedure continues. If not, it returns to the browsing step (Step 2).

4. **Select Product:** When a customer is interested in a product, they select it to view the product details.

5. **View Product:** Following selection, the user examines the product and decides whether or not they wish to purchase it.
6. **Decision: Want to Purchase It?** If the user decides to buy the product, they can add it to their cart. If not, they can return to browsing products (Step 2).
7. **Add to Cart:** The user adds the selected product to their cart.
8. **View/Edit Cart:** The customer can view their cart and make any necessary adjustments.
9. **Proceed to Checkout:** After finalizing the cart, the user proceeds to checkout.
10. **Enter shipping Information:** The user enters their shipping information.
11. **Choose Payment Method:** The user selects their preferred payment method.
12. **Confirm Payment:** The user enters payment details, and the payment is processed.
13. **Order Confirmation:** The order process closes with confirmation that the order was successfully placed.
14. **End (Black Circle):** This represents the end of the order procedure.

Design

Introduction

A meticulously designed software design can pave the way for simple implementation, decreasing the likelihood of significant changes at later stages of development. During the design phase, both front-end programming languages and back-end databases that are critical to the project's artefact are prioritized. During this phase, a variety of tools will be used for both back-end and front-end development. While the back end will use the Entity Relationship Diagram (ERD) to depict its logical and conceptual structures, the front end will use wireframing and prototyping techniques.

Database Design (Back End Design)

The initial stage in the design phase of this project is to identify all relevant entities and data. This phase's goal is to translate the specified system requirements into a comprehensive database architecture, specifying the required entities (tables) and their attributes. This ensures that the system can store, retrieve, and handle data efficiently, by the essential functionalities.

The following entities were identified as a result of the requirements study.

User Management and Authentication (FR1 Registration, FR2 Login, FR3 User Profile Management) - We identified the need for a **'users'** entity to handle user registration, login, and profile management functions. This entity will hold crucial information about users, such as credentials, contact information, and personal information.

Product Management and Browsing (FR4 Product Catalogue, FR5 Search Functionality, FR6 Product Filtering, FR7 Product Detail Page) - These requirements involve the existence of a **'products'** entity that maintains extensive product information, allowing for product display, search, and filter functionality.

Order Management and Checkout (FR8 Add to Cart, FR9 Shopping Cart Management, FR10 Checkout Process) - We identified the **'orders'** entity to handle shopping cart functionalities and manage the checkout process. The orders entity will handle placing and managing orders.

Review (FR14 Reviews and Ratings) - The **'review'** entity is created to allow users to give product reviews and ratings.

Spinner Game (FR19 Gamification Feature) – A need for a 'spinner_winners' and 'user_spins' entity was identified to make the gamification feature possible. It Directly supports the gamification feature, adding an element of fun and potential rewards to enhance user retention and satisfaction. The spinner winner entity is used to record the game winners while the user_spins entity is used to record all the user spins.

Reporting and Analytics (FR20 Analytics and Data Visualization) - Supports administrative tasks such as analytics and data visualization, helping to monitor and improve the platform.

Relationships:

We effectively extracted the necessary entities and their attributes required for database design by thoroughly studying the system requirements. This design establishes the groundwork for a reliable system capable of handling user management, product browsing, shopping cart functionality, and the checkout process. The next crucial step in refining the database design will be to ensure adequate relationships and data integrity restrictions between these entities.

Outlined below are the relationships between the entities that were derived based on the functional requirements and the attributes of the entities.

1. **Products and Categories: (One-to-Many Relationship)**

- Relationship: Each product belongs to one category, but each category can have many products.
- Rationale: Products include a category_id field, meaning that products are categorized into categories. This classification allows products to be categorized for better organization and searchability.

2. **Products and Spinner_Winners: (One-to-Many Relationship)**

- Relationship: A product may be associated with many winning entries in the spinner_winners table, but each winner is associated with one product.
- Rationale: The Spinner_Winners entity includes a product_id field, which indicates that each winner record is associated with a certain product.

3. **Products and Reviews: (One-to-Many Relationship)**

- Relationship: One product can have multiple reviews, but each review is associated with one product.
- Rationale: The Review entity contains a product_id attribute, which indicates that each review belongs to a specific product.

4. **Products and Reports (One-to-Many Relationship)**

- Relationship: One product can be included in multiple reports, but each report includes one product.
- Rationale: The Reports entity includes a product_id attribute that associates each report with a distinct product. This is required for keeping track of the report's history and managing user order reports.

5. **Users and Reviews (One-to-Many Relationship)**

- Relationship: One user can make multiple reviews, but each cart review is associated with one user.
- Rationale: Each review is associated with a specific user, as indicated by the Review entity's user_id field. This enables users to make multiple reviews on different products.

Users and Spinner_Winners (One-to-Many Relationship)

- Relationship: One user can be in multiple winning entries, and one winning entry can only be associated with one user.
- Rationale: Each winning entry is associated with a specific user, as indicated by the Spinner Winners entity's user_id field. This enables users to win multiple times.

6. **Users and Orders (Many-to-Many Relationship)**

- Relationship: A user can place multiple orders, and each order can only be associated with one user.
- Rationale: The Orders entity includes a user_id field, which indicates that each order is associated with a certain user.

7. **Users and User_Spins (One-to-Many Relationship)**

- Relationship: One user can attempt multiple spins, but each spin is made by one user.
- Rationale: A user can have many entries in the user_spins table, each representing an individual spin in a game or promotional event. The user_id in user_spins tracks which spins belong to which user.

8. Orders and Reports (One-to-Many Relationship)

- Relationship: One order can be included in many reports, but each report is associated with one order alone.
- Rationale: The Reports entity contains an order_id attribute, which indicates that each report belongs to a specific order.

By establishing these relationships, we ensure that the database is capable of handling and retrieving related data across different entities, resulting in a comprehensive solution that fits all of the system's requirements.

Entity Relationship Diagram (ERD)

An Entity-Relationship Diagram (ERD) was designed to visually illustrate the database structure, using the entities and their properties as previously established and discussed. Please see the detailed representation below:

Table 17: Displaying the list of entities and their attributes.

Entity	Attributes
products	product_id, cat_id, product_name, product_desc, product_price, product_quantity, product_status, product_tags, product_date, product_comment_count, productimg_1, productimg_2, productimg_3, productimg_4, stripe_product_id, stripe_product_url
categories	cat_id, cat_title, cat_img, is_active
reviews	review_id, product_id, user_id, review_author, review_email, review_content, review_status, review_date
users	user_id, user_email, user_name, user_password_hash, user_type, user_image, create_at, reset_token_hash, reset_token_expires_at
spinner_winners	winner_id, product_id, user_id, win_date
orders	order_id, user_id, stripe_session_id, order_amount, order_transaction, order_status, order_currency, create_at
user_spins	id, user_id, spin_date
reports	report_id, product_id, stripe_product_id, stripe_price_id, user_id, order_id, product_price, product_title, product_quantity, product_total, create_at

Normalization Process

The next stage involved determining the relationship between these entities as well as the cardinality of these relationships using the normalization process. Normalization is the process of structuring data to eliminate redundancy and reliance. It consists of dividing a database into tables and creating relationships between the tables based on rules designed to safeguard the data and make the database more flexible by removing redundancy and inconsistent dependency. The figure below illustrates an entity relationship diagram, which includes all of the entities, their attributes, and the relationships that exist between them.

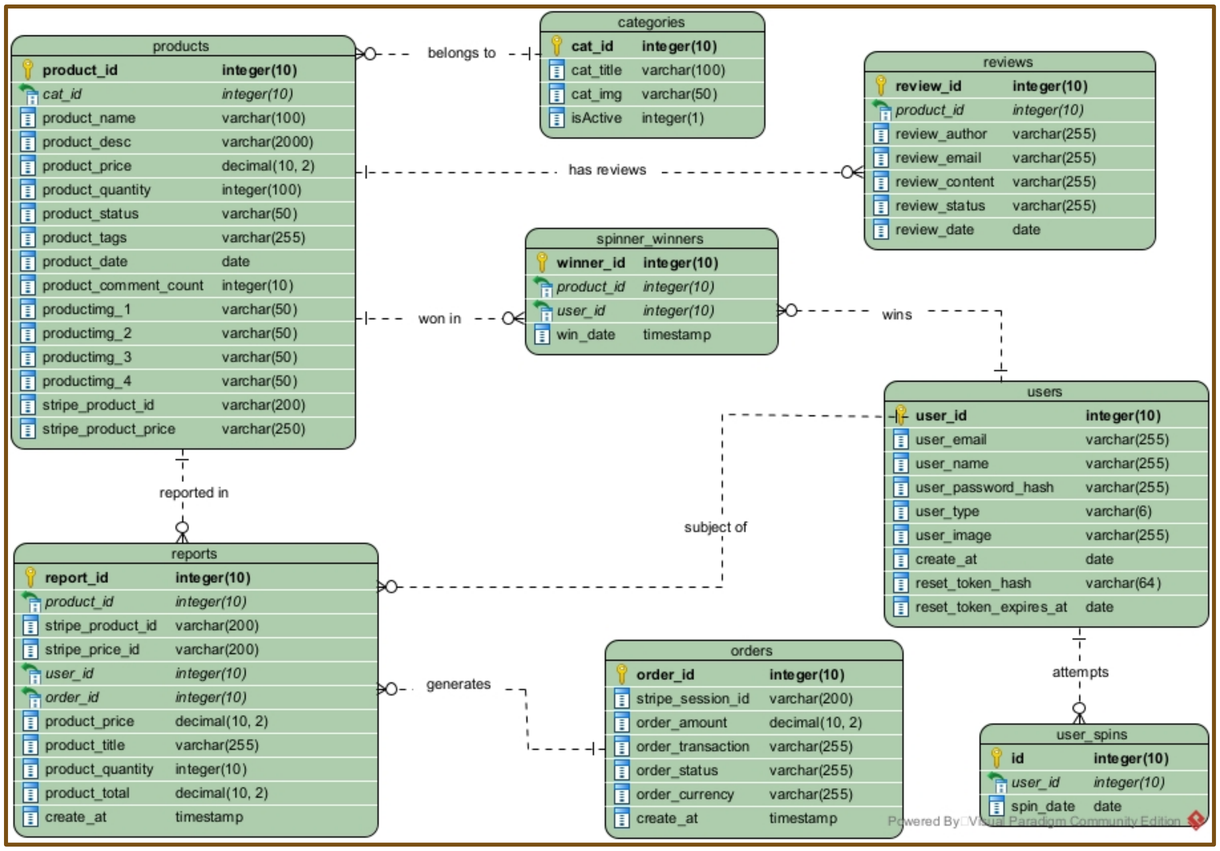


Figure 17: Displaying ERD diagram.

Explanation of the normalization process evident in the diagram above:

First Normal Form (1NF): Every table has a primary key that makes its rows uniquely identifiable. For instance, the primary key in the users table is id. Additionally, the tables do not include any repeated groupings, meeting the 1NF condition.

Second Normal Form (2NF):

Since the tables are already in 1NF and all of the non-key attributes are dependent on the main key, they are in 2NF. For example, in the orders database, characteristics such as order_amount, order_status, and so on are dependent upon the main key, id, and not just a subset of it.

Third Normal Form (3NF): The tables are in 3NF, which means that all of the attributes are only reliant on the primary key, preventing transitive dependencies. For example, the order table has its primary key, and user-related information is linked to it via a foreign key rather than contained in the orders table, avoiding transitive dependencies.

Relationships: The diagram illustrates relationships such as one-to-many or many-to-one by linking the tables with lines. These connections indicate the foreign key constraints required for referential integrity. The orders table, for example, includes a foreign key that connects to the users table, indicating that each order is placed by a user.

Normalization Beyond 3NF: Although normalization forms other than 3NF, such BCNF or 4NF, are not specifically shown in the figure, the structure demonstrates a well-organized schema that probably complies with higher normalization standards, guaranteeing that the database stays free of anomalies and redundancies.

In conclusion, the diagram represents a normalized database structure in which information is logically dispersed throughout multiple tables, each of which is dedicated to a certain entity type and linked by foreign keys. This structure keeps data integrity throughout the database, minimizes data duplication, and facilitates effective data retrieval.

Logical Model

The logical model step is critical for converting our conceptual understanding of the system into a more structured and database-oriented style. It defines each entity's attributes and establishes primary and foreign keys, maintaining data integrity throughout the system. The following table below represents the Logical Data Model based on the previously provided tables and attributes, highlighting the main and foreign keys.

Table 18: Displaying the Logical Schema.

Entities	Attributes
products	<u>product_id</u>, cat_id*, product_name, product_desc, product_price, product_quantity, product_status, product_tags, product_date, product_comment_count, productimg_1, productimg_2, productimg_3, productimg_4, stripe_product_id, stripe_product_url
categories	<u>cat_id</u>, cat_title, cat_img, is_active
reviews	<u>review_id</u>, product_id*, user_id*, review_author, review_email, review_content, review_status, review_date
users	<u>user_id</u>, user_email, user_name, user_password_hash, user_type, user_image, create_at, reset_token_hash, reset_token_expires_at
spinner_winners	<u>winner_id</u>, product_id, user_id, win_date

orders	order_id, user_id*, stripe_session_id, order_amount, order_transaction, order_status, order_currency, create_at
user_spins	id, user_id*, spin_date
reports	report_id, product_id*, stripe_product_id, stripe_price_id, user_id*, order_id*, product_price, product_title, product_quantity, product_total, create_at

Data Dictionary

The data dictionary for the users table is displayed in Table 19 below. View Appendix B for the entire data dictionary.

Table 19: Displaying Data Dictionary.

USERS							
Description: User Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
user_id	integer	10	PK	No	None	Autoincremented	Uniquely identifies every user.
user_email	varchar	255		No	None	Must be in a valid email format	Email of the user.
user_name	varchar	255		No	None		Name of the user.
user_password_hash	varchar	255		No	None		Hashed password of the user.
user_type	varchar	6		No	None		Type of user (e.g., admin, customer).

user_image	varchar	255		Yes	None		URL or path to the user's image.
create_at	date			No	None		Date the user was created.
reset_token _hash	varchar	64		Yes	None		Hash of the password reset token.
reset_token _expires_at	date			Yes	None		Expiration date of the reset token.

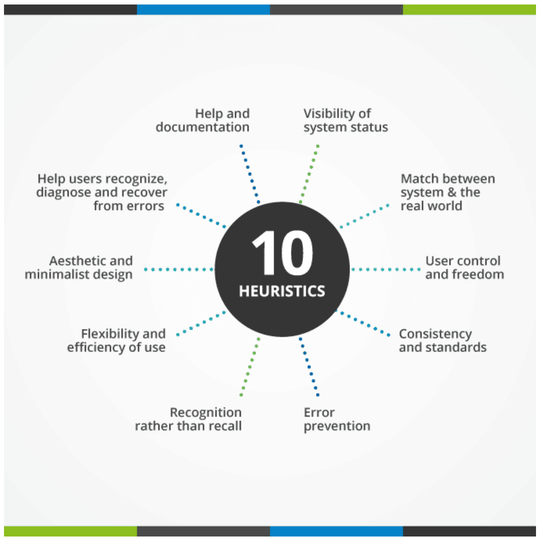
Application Design (Front End Design)

User experience is the main focus of front-end development, which employs coding and design strategies to produce sophisticated, user-friendly, quick, and secure interfaces. Front-end developers need to make sure that users receive consistent, excellent user experiences across a range of devices and use scenarios as mobile and smart device applications continue to rise in popularity (Cloudinary, n.d.).

Nielsen’s Heuristics

Heuristic Evaluation (HE), which emphasizes User Centred Design concepts, is a popular technique for evaluating interactive systems' usability. This is a method used by experts to assess the usability of user interfaces, revealing insights that can help design teams improve product usability from early stages of development (Interaction Design Foundation, 2019). Figure 20 below represents the 10 Neilsen’s Heuristics for usability.

Table 20: Displaying 10 Neilsen's Heuristics for usability.


10
HEURISTICS
Visibility of system status
Match between system & the real world
User control and freedom
Consistency and standards
Error prevention
Recognition rather than recall
Flexibility and efficiency of use
Aesthetic and minimalist design
Help users recognize, diagnose and recover from errors
Help and documentation

The table below illustrates one of the Nielsen's heuristics strategies that will be employed in the application.

Table 21: Displaying the Consistency and Standards principle.

Principle 2: Consistency and Standards	
Description: Users should not have to guess whether various words, contexts, or actions indicate the same thing. Adhere to platform and industry conventions.	
How will the software artefact adhere to the principle?	**Consistent Terminology:** The program will use consistent terminology across every page and function. If a shopping cart is referred to as a "Cart" in one part of the program, it will not be designated as a "Basket" in another. **Standardized Navigation:** The program will maintain a uniform layout and design for all navigation elements, including buttons, menus, and links. This makes sure that no matter where they are in the program, users can consistently go to specific sections after they learn where to access certain functions or information. **Adherence to Platform Conventions:** Whether the application is an iOS, Android, or web app, it will adhere to the norms and regulations of the platform it is operating on. This aids in fulfilling consumers' expectations

	stemming from their previous interactions with other apps on the same platform.

Sketches

Sketching is a crucial and much underestimated aspect of the user interface design process (even for non-designers). Starting your drawings offline and using pen and paper early in the process will help individuals better visualize the project's potential. It's often difficult to define what your imagination is conjuring up at this stage... The good news is that you don't have to be an artist to get your thoughts down on paper. All you need is an idea and some familiarity with the concept. A sketch of the login page is displayed below whiles the rest can be found in the Appendix F:

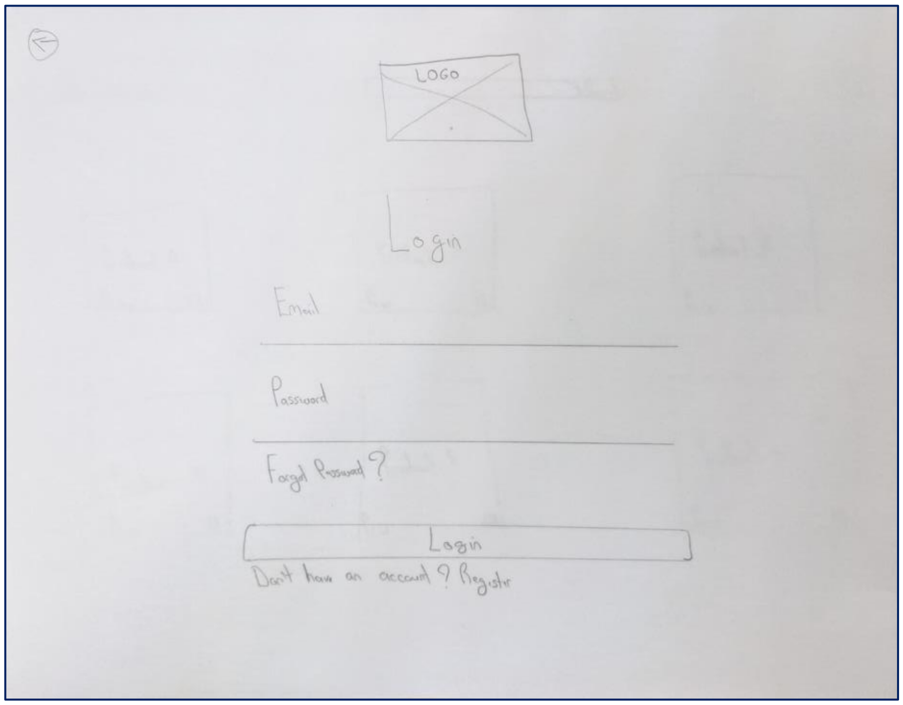


Figure 17A: Displaying a sketch of the login page.

Wireframes

Wireframes is a vital component in the website design and development process, serving as a visual reference for the structure, layout, and operation of a site's pages. They aid in the planning of user interface elements, navigational flow, and content placement, resulting in a user-centric design approach. Wireframes are useful for arranging complex information, tools, and resources when it comes to any artifact in an easy-to-access and user-friendly manner. Furthermore, wireframes promote clear communication among team members, assisting in the alignment of visions and expectations prior to the start of the development phase, thus saving time and money. Wireframes are critical in aligning website design with heuristic principles, which ensures a user-friendly experience. Designers can prepare for the presence of familiar buttons, similar colours, and a consistent layout across multiple pages by employing wireframes, while following to established usability guidelines. This aids in the development of an intuitive navigation structure, allowing users to interact with the website and find the information they require more easily. Wireframes, in essence, help to facilitate a design process that prioritizes usability, resulting in a more efficient and gratifying user experience. The following figure illustrates a wireframe made for the product page.

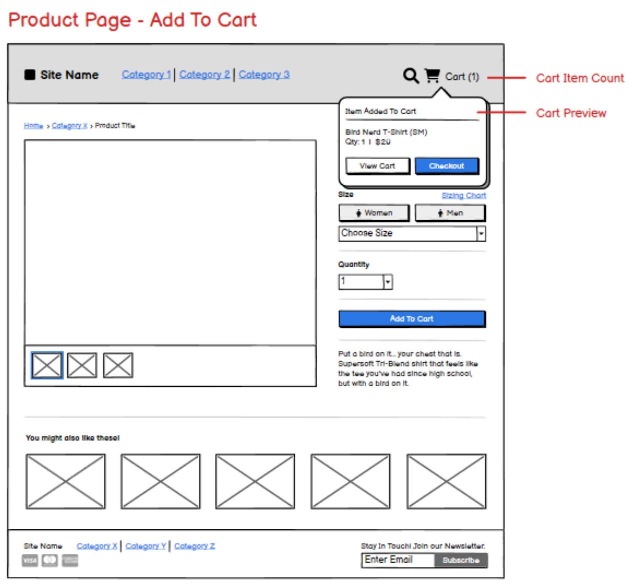

Product Page - Add To Cart
Site Name
Category 1 | Category 2 | Category 3
Cart (1)
Cart Item Count
Cart Preview
Item Added To Cart
Bird Nerd T-Shirt (SM)
Qty: 1 | $20
View Cart
Checkout
Home › Category X › Product Title
Size
Sizing Chart
Women
Men
Choose Size
Quantity
1
Add To Cart
Put a bird on it... your chest that is. Supersoft Tri-Blend shirt that feels like the tee you've had since high school, but with a bird on it.
You might also like these!
Site Name
Category X | Category Y | Category Z
Stay In Touch! Join our Newsletter.
Enter Email
Subscribe

Figure 18: Displaying wireframe for product details page.

Interactive Prototype

An interactive prototype is a dynamic model of a website that enables designers, developers, and stakeholders to interact with and experience the user interface and navigational flow prior to the commencement of actual production. It goes beyond static wireframe and mock-up representations by including clickable elements, transitions, and animations to replicate real user interactions. The usage of interactive prototypes in the website development process allows for early testing of the design's usability and efficacy, allowing for the detection and resolution of user experience concerns. This improves communication within the development team as well as with stakeholders because everyone can see and understand the planned functionality and design, resulting in better informed decisions and fewer misunderstandings. Additionally, the simplicity with which adjustments may be made during the prototype stage saves a great deal of time and money when compared to later stages of development, which contributes to a more effective and successful website project.

Figures 19, 20 and 21 depict some of the visual components that will be part of the prototypes, as well as the brand style guidelines. This covers the primary image on the home page, the logo, and the prototype's colour scheme, fonts, and typography and all of the general styles and spacing in CSS code. Additionally, CSS is included for elements like font sizes and button styles.

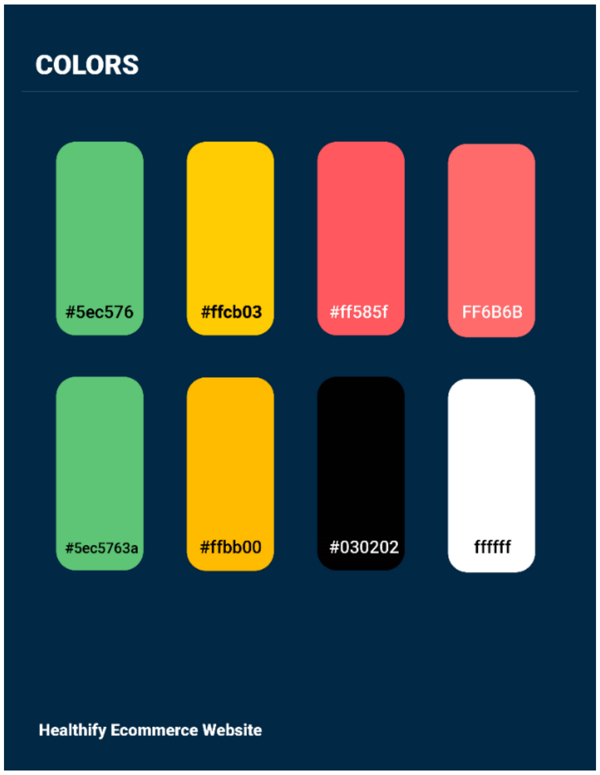
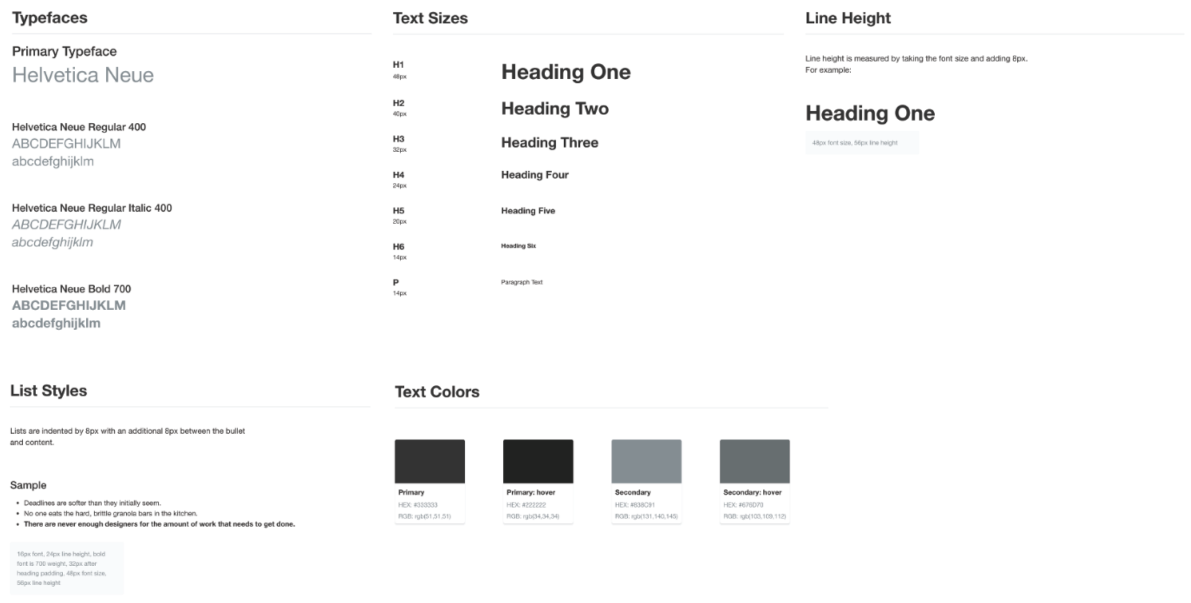


Figure 20: Displaying colours for website.

Figure 19: Displaying home page main picture.



Typefaces

Primary Typeface
Helvetica Neue

Helvetica Neue Regular 400
ABCDEFGHIJKLM
abcdefghijklm

Helvetica Neue Regular Italic 400
ABCDEFGHIJKLM
abcdefghijklm

Helvetica Neue Bold 700
ABCDEFGHIJKLM
abcdefghijklm

Text Sizes

H1 48px	Heading One
H2 40px	Heading Two
H3 32px	Heading Three
H4 24px	Heading Four
H5 20px	Heading Five
H6 14px	Heading Six
P 14px	Paragraph Text

Line Height

Line height is measured by taking the font size and adding 8px.
For example:

Heading One

48px font size, 56px line height

List Styles

Lists are indented by 8px with an additional 8px between the bullet
and content.

Sample

- Deadlines are softer than they initially seem.
- No one eats the hard, brittle granola bars in the kitchen.
- **There are never enough designers for the amount of work that needs to get done.**

16px font, 24px line height, bold font is 700 weight, 32px after heading padding, 48px font size, 56px line height

Text Colors

Primary	Primary: hover	Secondary	Secondary: hover
HEX: #333333	HEX: #222222	HEX: #838C91	HEX: #676D70
RGB: rgb(51,51,51)	RGB: rgb(34,34,34)	RGB: rgb(131,140,145)	RGB: rgb(103,109,112)

Figure 21: Displaying brand style guidelines.

Implementation

Introduction

This chapter of document encompasses of the transformation of the Software Design Document into executable code. It emphasizes the importance of harmoniously marrying the back end and front end when transforming plans into reality. The recommender system, gamification, and responsive design all come to life as well as the core functionalities. There is a review of how tools and languages such as HTML, CSS, JavaScript, PHP, and others play a critical role in realizing the vision. In addition, the incorporation of security measures and version control systems ensures that development is streamlined.

Database Implementation

The database for the application was made using MySQL, an open-source relational database management system that is part of the Laragon server. A database entitled "healthify_db" was established, and tables derived from the data dictionary in the design chapter were added to the database. In every database table, the primary and foreign keys were also established, along with the restrictions listed in the data dictionary. All the tables were created using the MySQL interface. The diagram below displays the MySQL interface that was used to create the database and the tables.

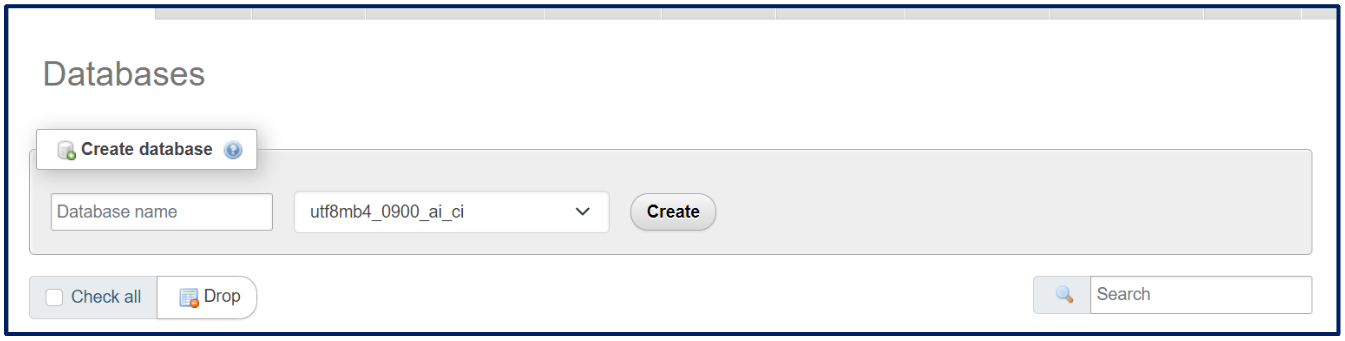


Figure 22: Displaying database creation.

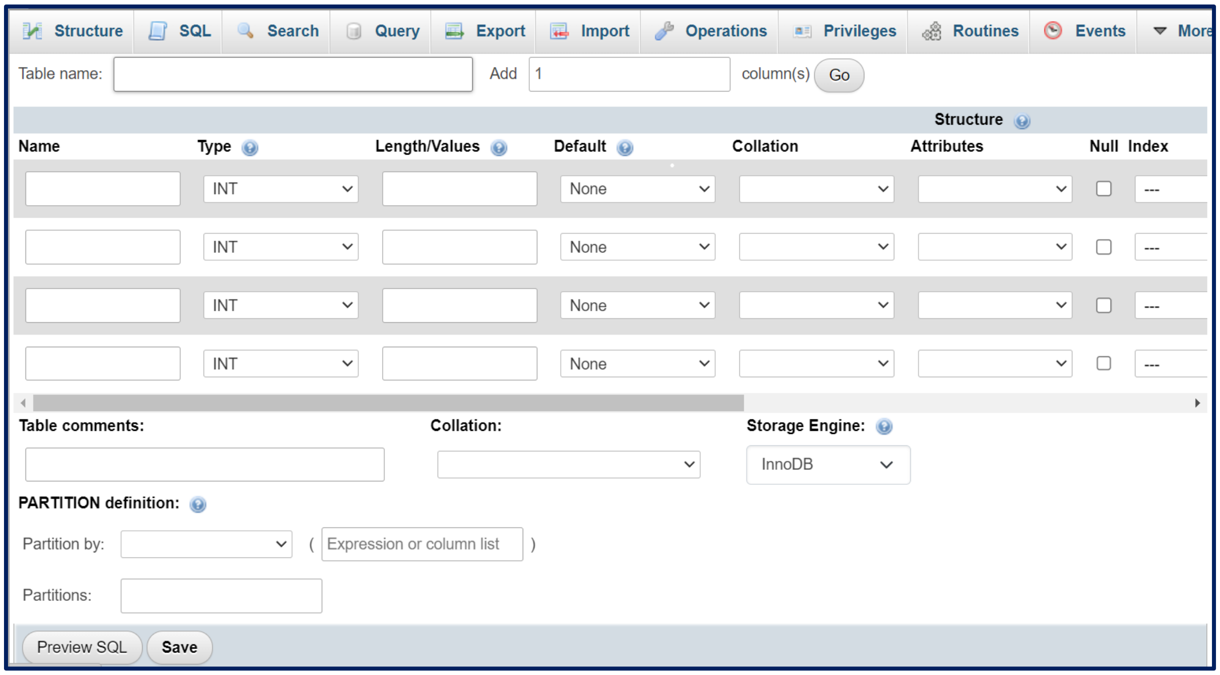


Table name: Add 1 column(s)

					Structure		
Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
	INT		None			☐	---
	INT		None			☐	---
	INT		None			☐	---
	INT		None			☐	---

Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Partitions:

Figure 23: Displaying interface for creating tables.

Connection between Database and Front End

A crucial aspect of this web application and all websites in general is a database connection, which allows it to interact with a database server to save, retrieve, and update data. It is fundamental to building dynamic, data-driven online applications and is crucial for guaranteeing data security, scalability, and integrity.

Explanation for the code in the diagram 24 below:

This code is located in a file named "config.php". The mysqli class is used to create a new MySQLi database connection. This constructor takes four parameters which are host, username, password and database name in my case these parameters are:

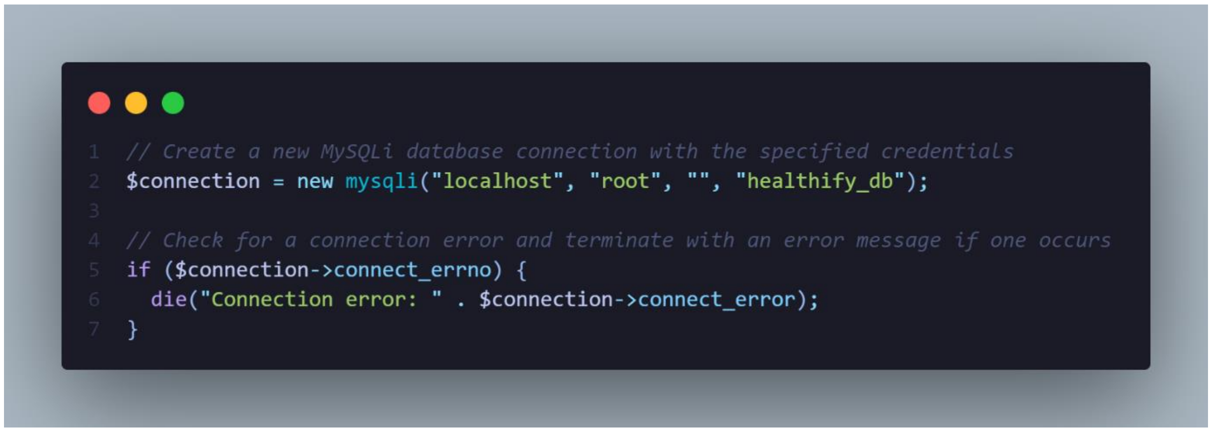
- **localhost:** This is the hostname or server name on which the MySQL database is hosted. It's the local server for this project.
- **root:** This represents the username that is used to login to the MySQL database. In this case, it is "root".
- **":** This is the password used to connect the MySQL database. For this project, the password is nothing, indicating no password was set.

- **Healthify_db:** The name of the database to which you want to connect. In this case, the name of the database is "healthify_db."

After establishing the connection, the code examines the connect_errno attribute of the $connection object for a connection error. If there is an error, it indicates that the database connection could not be established.

The die function is triggered with an error message in the event of a connection error. The exact error message that was retrieved from $connection->connect_error is presented in the error message after the words "Connection error: "

This code guarantees that the database connection is made, and it prevents the script from running and displays an error message if there are any problems when trying to connect.



```
// Create a new MySQLi database connection with the specified credentials
$connection = new mysqli("localhost", "root", "", "healthify_db");

// Check for a connection error and terminate with an error message if one occurs
if ($connection->connect_errno) {
  die("Connection error: " . $connection->connect_error);
}
```

Figure 24: Displaying code for database connection.

User Interface Implementation

In this section, the core and advanced features will be implemented and illustrated. All of the sixteen core functionalities have been completed. All of the advanced features have also been completed. Table 22 provides an update on all finished features, along with a status column indicating the percentage of features completed. The implementation of each feature will be covered in further detail and with supporting data in the two sub-sections that follow.

Table 22: Displaying features.

No.	Requirements	Requirement Type	Complete

FR1	Registration	Core	100%
FR2	Login	Core	100%
FR3	User Profile Management	Core	100%
FR4	Product Catalogue	Core	100%
FR5	Search Functionality	Core	100%
FR6	Product Filtering	Core	100%
FR7	Product Detail Page	Core	100%
FR8	Add to Cart	Core	100%
FR9	Shopping Cart Management	Core	100%
FR10	Checkout Process	Core	100%
FR11	Payment Integration	Core	100%
FR12	Order Confirmation	Core	100%
FR13	Order History	Core	100%
FR14	User Reviews and Ratings	Core	100%
FR15	Account Management	Core	100%
FR16	Admin Dashboard	Core	100%
FR17	Chatbot Integration	Advanced	100%
FR18	Recommendation Engine	Advanced	100%
FR19	Gamification Features	Advanced	100%
FR20	Advanced Analytics and Data Visualization	Advanced	100%
FR21	Responsive Design	Advanced	100%

Core Features Implemented

FR1 – Registration

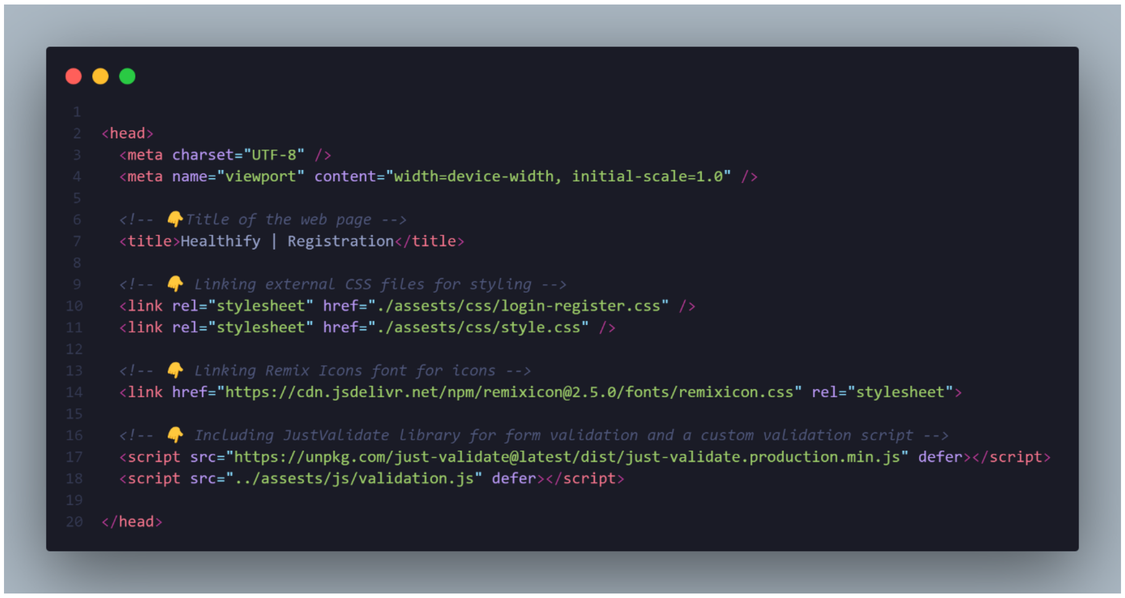
Overall Explanation of the Functionality

Users begin the registration process by completing a registration form with their selected username, email, and password. The Just Validate package is used on the client side to guarantee that the input data fulfils a specified criterion. To check email availability, an asynchronous JavaScript (AJAX) request is issued to a server-side PHP script. PHP server-side validation verifies the integrity and format of the data, providing a password hash for storage. Users who successfully register are redirected to a success page, whereas failures/errors, such as email duplication, are displayed to the user.

BACK END | Let's take a closer look at the registration process (How the functionality was implemented):

1. HTML Configuration (see Figure 25):

The registration page's HTML structure includes the necessary components for configuring viewport settings, character encoding, and resource connections. It also includes custom validation scripts and the JustValidate library, which are attentively delayed for execution after HTML processing. This configuration ensures accurate form validation as well as an appealing layout. See diagram 25 below for the code discussed.

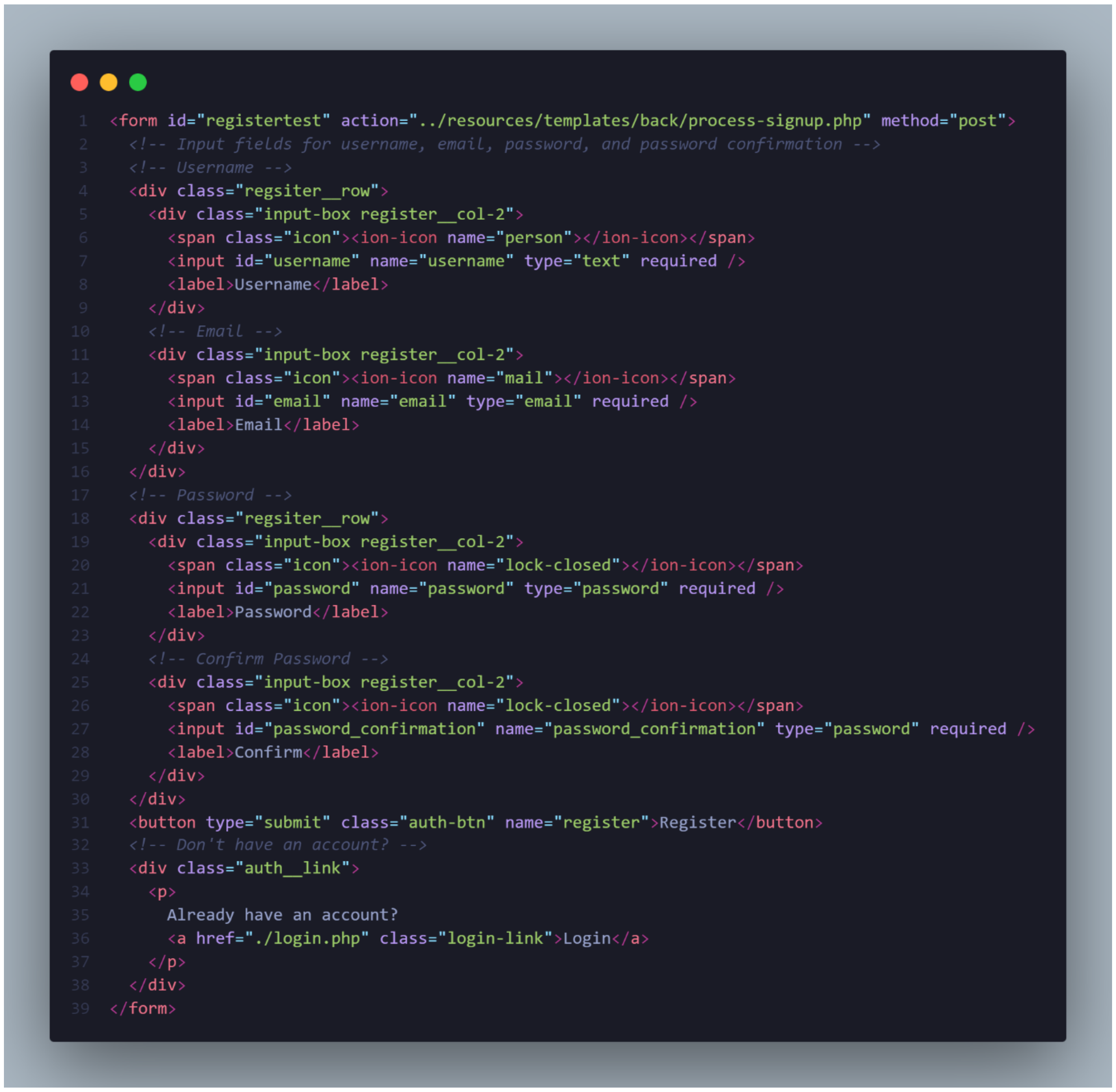


```html

<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />

  <!-- 👇Title of the web page -->
  <title>Healthify | Registration</title>

  <!-- 👇 Linking external CSS files for styling -->
  <link rel="stylesheet" href="./assests/css/login-register.css" />
  <link rel="stylesheet" href="./assests/css/style.css" />

  <!-- 👇 Linking Remix Icons font for icons -->
  <link href="https://cdn.jsdelivr.net/npm/remixicon@2.5.0/fonts/remixicon.css" rel="stylesheet">

  <!-- 👇 Including JustValidate library for form validation and a custom validation script -->
  <script src="https://unpkg.com/just-validate@latest/dist/just-validate.production.min.js" defer></script>
  <script src="../assests/js/validation.js" defer></script>

</head>
```

Figure 25: Displaying HTML configuration.

2. Registration Form (see Diagram 26):

The registration form includes input fields for a password, username, and email address, as well as a password confirmation field. The layout of the form is comprised of elements. Each input field has a label and an icon. The form data is transmitted via the POST method to a server-side PHP script (process-signup.php) to ensure secure data transmission. At the bottom of the form, there is a signup button and a link to the login page for individuals who possess existing accounts. For those who are unfamiliar with website interactions, this form is intended to be both functional and simple to use. The code in diagram 26 below represents the user registration form.

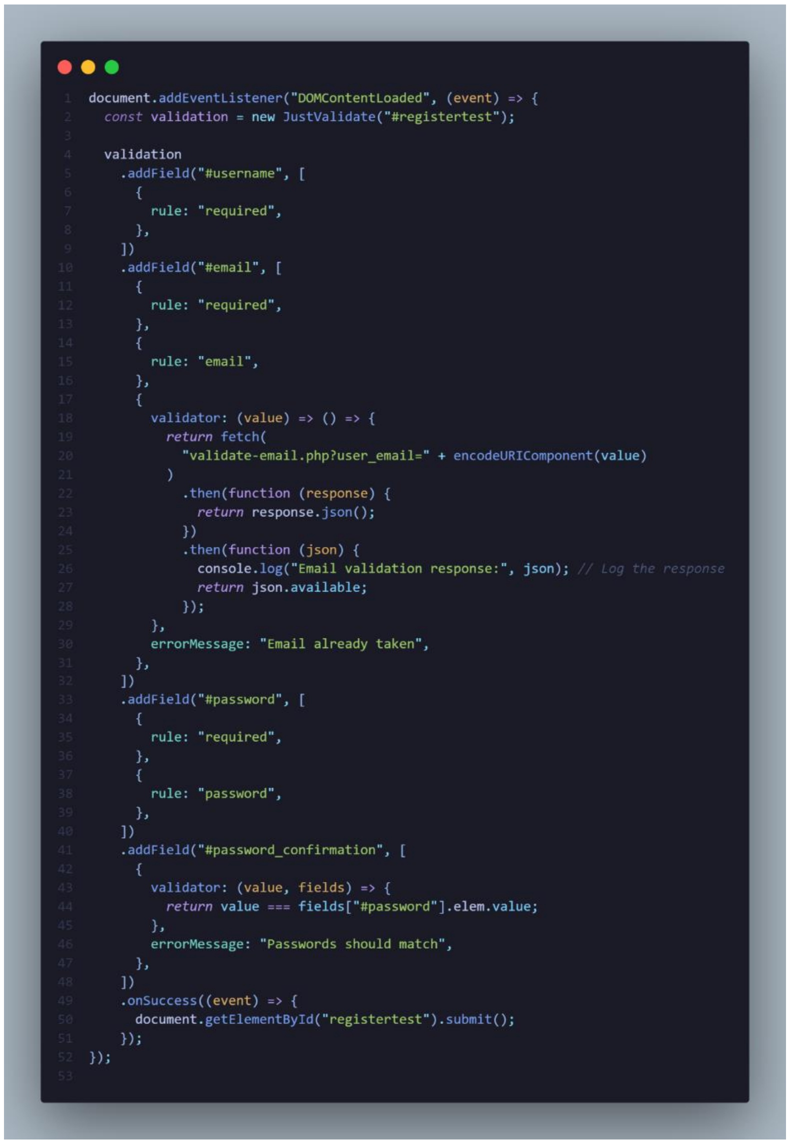


```html
<form id="registertest" action="../resources/templates/back/process-signup.php" method="post">
  <!-- Input fields for username, email, password, and password confirmation -->
  <!-- Username -->
  <div class="regsiter__row">
    <div class="input-box register__col-2">
      <span class="icon"><ion-icon name="person"></ion-icon></span>
      <input id="username" name="username" type="text" required />
      <label>Username</label>
    </div>
    <!-- Email -->
    <div class="input-box register__col-2">
      <span class="icon"><ion-icon name="mail"></ion-icon></span>
      <input id="email" name="email" type="email" required />
      <label>Email</label>
    </div>
  </div>
  <!-- Password -->
  <div class="regsiter__row">
    <div class="input-box register__col-2">
      <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
      <input id="password" name="password" type="password" required />
      <label>Password</label>
    </div>
    <!-- Confirm Password -->
    <div class="input-box register__col-2">
      <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
      <input id="password_confirmation" name="password_confirmation" type="password" required />
      <label>Confirm</label>
    </div>
  </div>
  <button type="submit" class="auth-btn" name="register">Register</button>
  <!-- Don't have an account? -->
  <div class="auth__link">
    <p>
      Already have an account?
      <a href="./login.php" class="login-link">Login</a>
    </p>
  </div>
</form>
```

Figure 26: Displaying code for registration form.

3. Client-Side Validation (see Diagram 27):

Client-side validation is handled by JavaScript, which is driven by the JustValidate library. It applies validation criteria to several fields after waiting for the HTML document to fully load. The email field requires an appropriate email format as well as an additional asynchronous check utilizing a server-side script (validate-email.php) to identify whether the email address is already in use. The username and password fields are straightforward to fill out. The entered password is checked to ensure that it matches the password confirmation field. The form is sent once all validations have been completed successfully. This approach reduces server load by pre-validating data on the client side and enhances user experience by providing immediate feedback on input errors. The JavaScript code shown in Figure 27 configures client-side validation for the registration form.

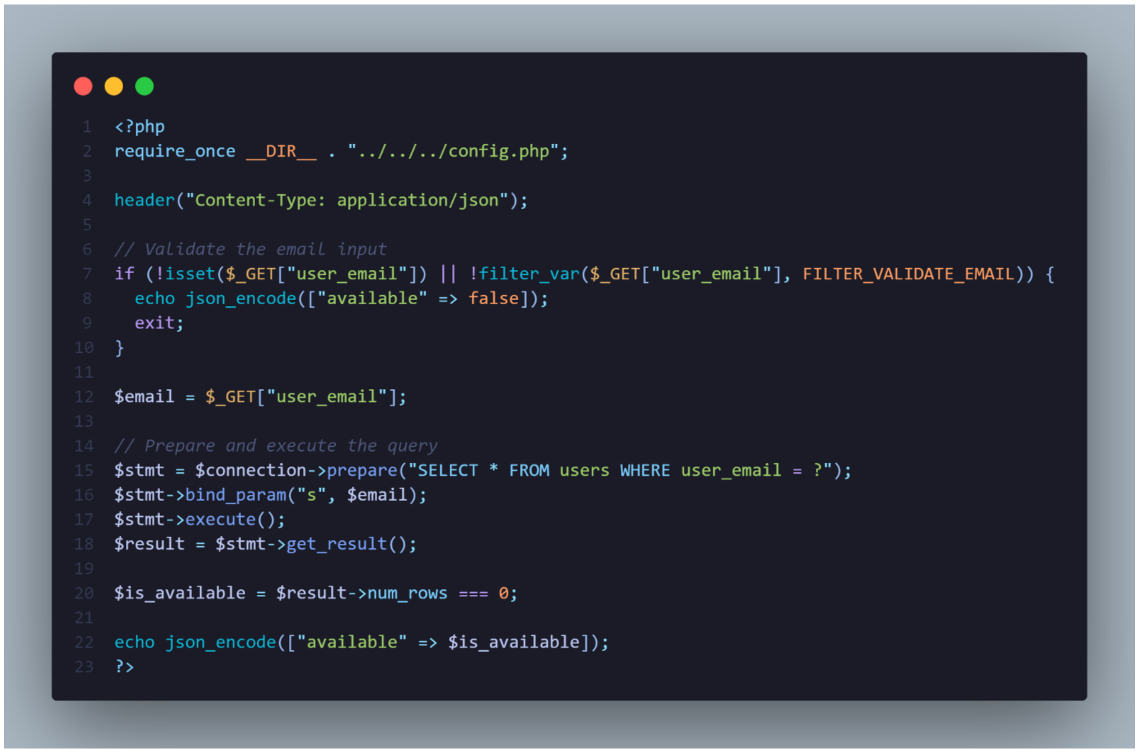


```javascript
document.addEventListener("DOMContentLoaded", (event) => {
  const validation = new JustValidate("#registertest");

  validation
    .addField("#username", [
      {
        rule: "required",
      },
    ])
    .addField("#email", [
      {
        rule: "required",
      },
      {
        rule: "email",
      },
      {
        validator: (value) => () => {
          return fetch(
            "validate-email.php?user_email=" + encodeURIComponent(value)
          )
            .then(function (response) {
              return response.json();
            })
            .then(function (json) {
              console.log("Email validation response:", json); // Log the response
              return json.available;
            });
        },
        errorMessage: "Email already taken",
      },
    ])
    .addField("#password", [
      {
        rule: "required",
      },
      {
        rule: "password",
      },
    ])
    .addField("#password_confirmation", [
      {
        validator: (value, fields) => {
          return value === fields["#password"].elem.value;
        },
        errorMessage: "Passwords should match",
      },
    ])
    .onSuccess((event) => {
      document.getElementById("registertest").submit();
    });
});
```

Figure 27: Displaying code for client-side validation.

4. **Email Availability Check (see Figure 28):**

A PHP script is responsible for determining email availability in real-time. It validates the email format, queries the database for the email, and includes a database configuration file. If the email cannot be identified in the database, a JSON object stating that the email is available is returned. This is used for real-time email validation on the registration form. This PHP script, shown in Figure 28, detects whether an email address is currently stored in a database.

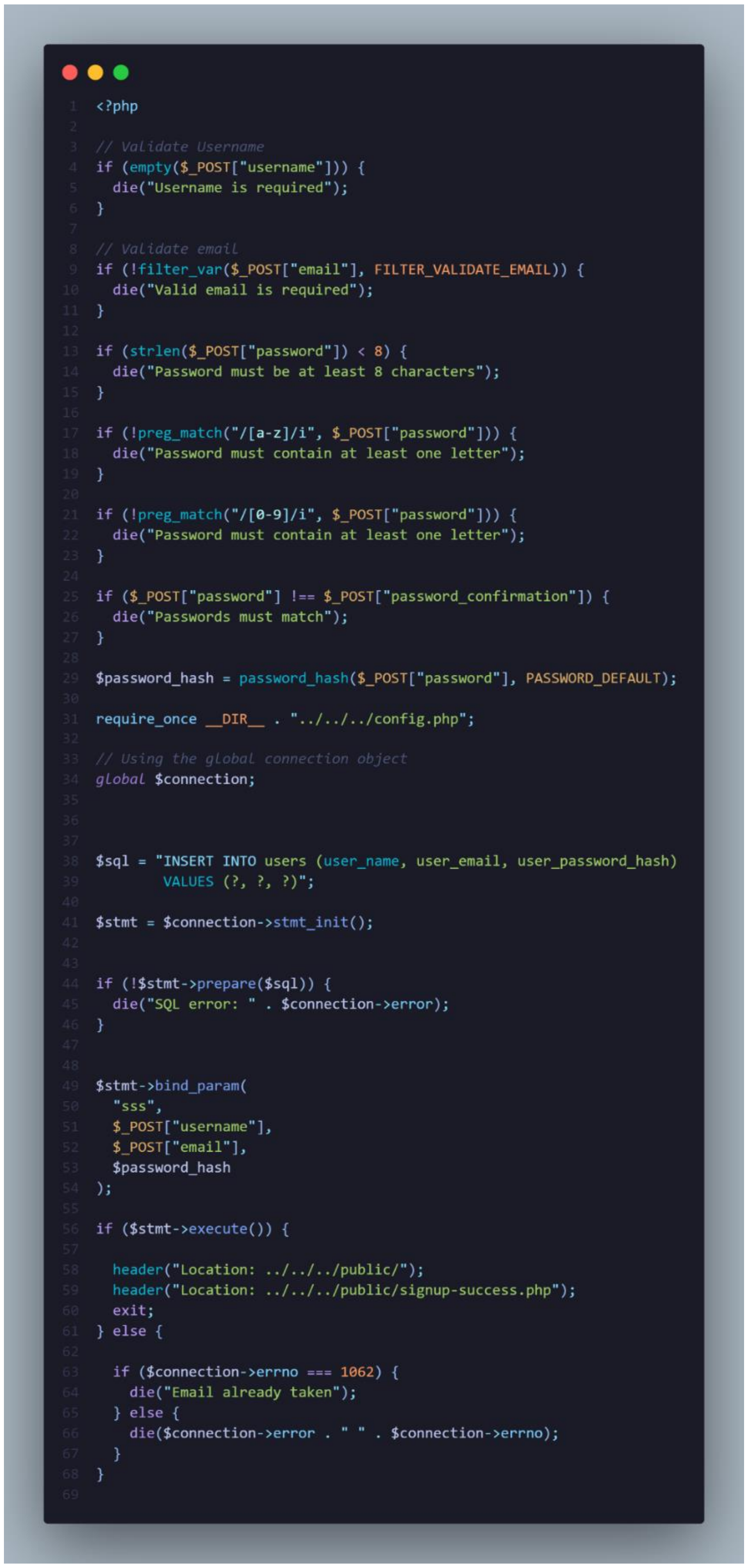


```php
<?php
require_once __DIR__ . "../../../config.php";

header("Content-Type: application/json");

// Validate the email input
if (!isset($_GET["user_email"]) || !filter_var($_GET["user_email"], FILTER_VALIDATE_EMAIL)) {
  echo json_encode(["available" => false]);
  exit;
}

$email = $_GET["user_email"];

// Prepare and execute the query
$stmt = $connection->prepare("SELECT * FROM users WHERE user_email = ?");
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

$is_available = $result->num_rows === 0;

echo json_encode(["available" => $is_available]);
?>
```

Figure 28: Displaying the code responsible for determining email availability.

5. **Server-Side Validation (see Figure 29):**

A PHP script validates user inputs on the server. It ensures that the username, email format, and password are all at least eight characters long, contain a letter and a number, and match the confirmation password before processing any input. If any validation fails, an error message is presented when the script finishes. It then hashes the password to assure its confidentiality. It then prepares an SQL query that enters the new user's information (username, email, and hashed password) into the "users" table after including the database's configuration file. If the insert process is successful, the user is directed to a success page; if not, the script either displays the SQL error or searches for a duplicate email error (SQL error code 1062), in which case the relevant message is displayed. When registering users, this script ensures input validation as well as safe user data processing. Figure 29 displays the PHP script that processes the user registration form.



```php
<?php

// Validate Username
if (empty($_POST["username"])) {
  die("Username is required");
}

// Validate email
if (!filter_var($_POST["email"], FILTER_VALIDATE_EMAIL)) {
  die("Valid email is required");
}

if (strlen($_POST["password"]) < 8) {
  die("Password must be at least 8 characters");
}

if (!preg_match("/[a-z]/i", $_POST["password"])) {
  die("Password must contain at least one letter");
}

if (!preg_match("/[0-9]/i", $_POST["password"])) {
  die("Password must contain at least one letter");
}

if ($_POST["password"] !== $_POST["password_confirmation"]) {
  die("Passwords must match");
}

$password_hash = password_hash($_POST["password"], PASSWORD_DEFAULT);

require_once __DIR__ . "../../../config.php";

// Using the global connection object
global $connection;



$sql = "INSERT INTO users (user_name, user_email, user_password_hash)
        VALUES (?, ?, ?)";

$stmt = $connection->stmt_init();


if (!$stmt->prepare($sql)) {
  die("SQL error: " . $connection->error);
}


$stmt->bind_param(
  "sss",
  $_POST["username"],
  $_POST["email"],
  $password_hash
);

if ($stmt->execute()) {

  header("Location: ../../../public/");
  header("Location: ../../../public/signup-success.php");
  exit;
} else {

  if ($connection->errno === 1062) {
    die("Email already taken");
  } else {
    die($connection->error . " " . $connection->errno);
  }
}

```

Figure 29: Displaying code for server-side validation.

FRONT END | Let's take a closer look at how the user experiences the registration process:

1. User-Friendly Registration Form (see Diagram 30)

Users encounter an intuitively designed registration form. Immediate client-side validation feedback enhances the user experience by providing instant error notifications. The diagram below illustrates the registration form.

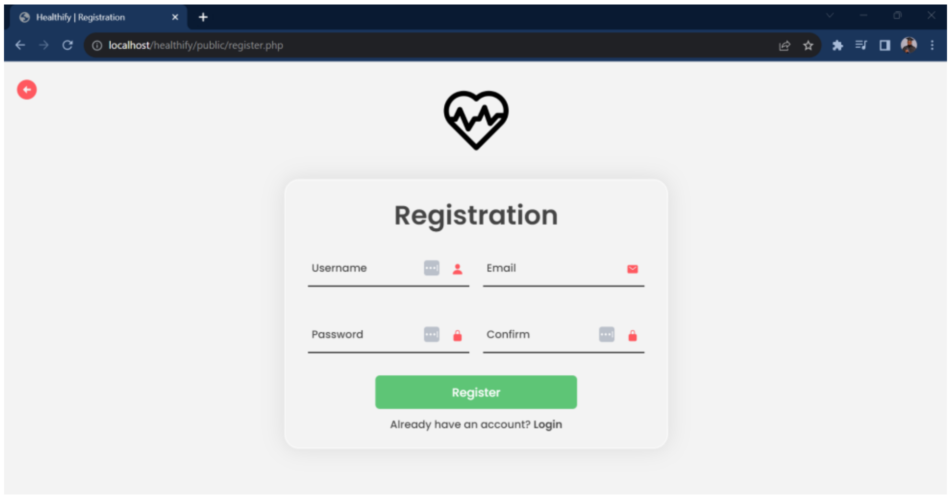


Figure 30: Displaying registration form.

2. Precise Validation Errors (see Figure 31):

When a user attempts to register but does not meet the validation criteria. Using the just validate library, as indicated at the beginning of this section, an appropriate message would be displayed. Take a look at Figure 31 for an example.

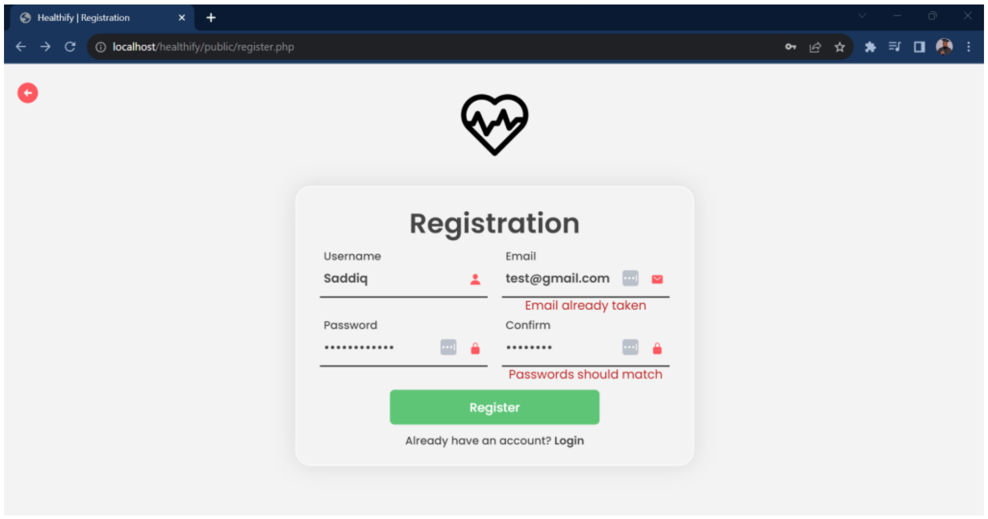
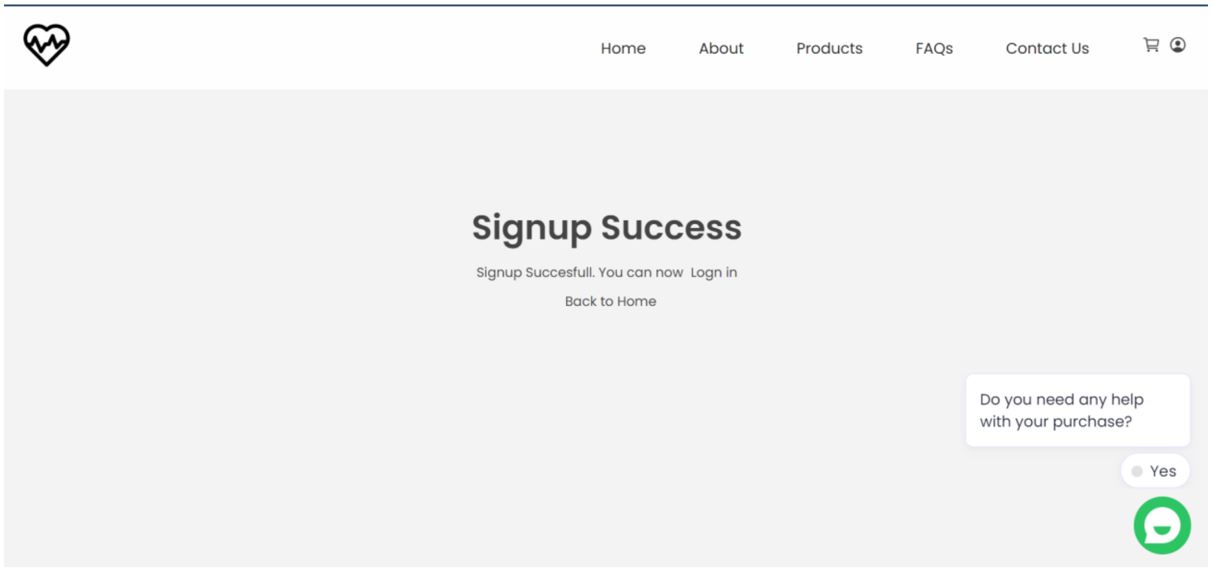


Figure 31: Displaying registration form with validation errors.

3. **Successful Registration (see Figure 32)**

These errors will persist if the user continues to attempt to register without meeting the validation requirements. The graphic below illustrates what would occur if they met the validation criteria and clicked the register button.



Signup Success

Signup Succesfull. You can now Login in

Back to Home

Do you need any help with your purchase?

Figure 32: Displaying registration success page.

After successfully registering, the user is presented with an appropriate message and given the option to log in or return to the home page, as shown above.

FR2 – Login

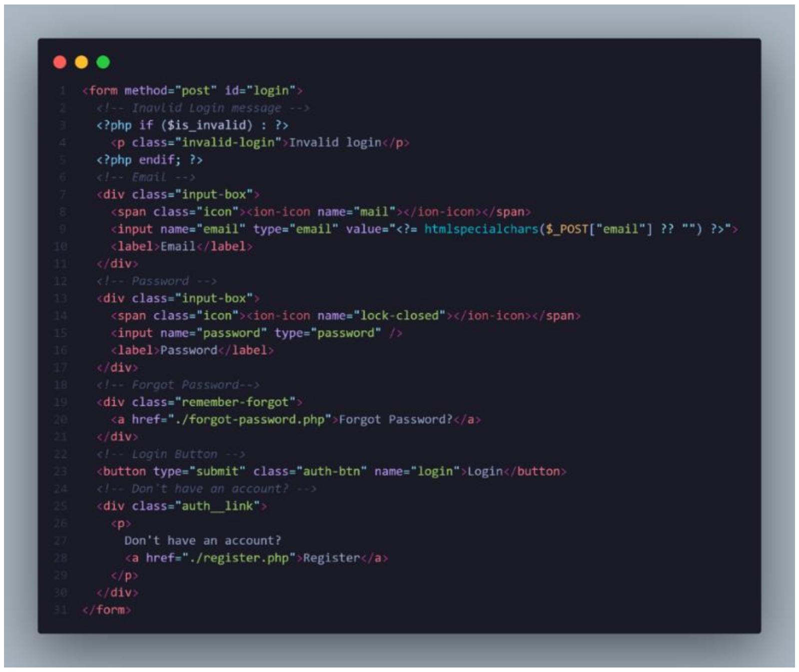
Overall Explanation of the Functionality

This section looks at the code for the login page. The server-side processing of the login form is handled by the PHP script in the beginning. It determines whether the form was submitted via POST, validates the email and password against the database using prepared statements, and initiates a new session after successful login, establishing session variables and forwarding to the homepage. If the login attempt fails, the script additionally sets the flag $is_invalid to true, which is used to display an error message on the form.

BACK END | Let's take a closer look at the login process (How the functionality was implemented):

1. **HTML Login Form (see Figure 33)**

The login form is designed with HTML and PHP. It starts with a PHP conditional block that checks to determine if a previous login attempt was unsuccessful, as indicated by the $is_invalid flag, and then displays an "Invalid login" message. The two key input fields on the form are the user's email address and password. The div components that contain both fields are stylized as input boxes, which visually improves the layout. To prevent HTML injection, any previously entered value is safely handled by htmlspecialchars and pre-populated in the email field. Each input box has an icon (represented by) for aesthetic purposes. There is also a link that will direct users to a page where they may reset their password if they have forgotten it. Figure 33 below displays the code that was utilized to create the login form.

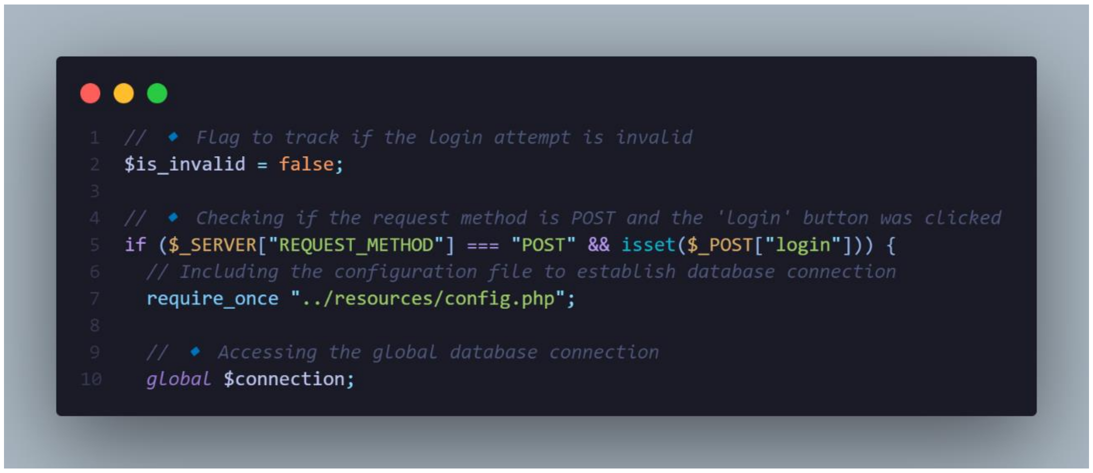


```
<form method="post" id="login">
  <!-- Inavlid Login message -->
  <?php if ($is_invalid) : ?>
    <p class="invalid-login">Invalid login</p>
  <?php endif; ?>
  <!-- Email -->
  <div class="input-box">
    <span class="icon"><ion-icon name="mail"></ion-icon></span>
    <input name="email" type="email" value="<?= htmlspecialchars($_POST["email"] ?? "") ?>">
    <label>Email</label>
  </div>
  <!-- Password -->
  <div class="input-box">
    <span class="icon"><ion-icon name="lock-closed"></ion-icon></span>
    <input name="password" type="password" />
    <label>Password</label>
  </div>
  <!-- Forgot Password-->
  <div class="remember-forgot">
    <a href="./forgot-password.php">Forgot Password?</a>
  </div>
  <!-- Login Button -->
  <button type="submit" class="auth-btn" name="login">Login</button>
  <!-- Don't have an account? -->
  <div class="auth__link">
    <p>
      Don't have an account?
      <a href="./register.php">Register</a>
    </p>
  </div>
</form>
```

Figure 33: Displaying code for login form.

2. PHP Login Script (see Figure 34)

The server-side login procedure is managed by this script. To track unsuccessful login attempts, it first resets the $is_invalid flag to false. The script includes a configuration file that establishes a database connection when the login form is submitted (determined by $_SERVER["REQUEST_METHOD"] === "POST" and isset($_POST["login"])). See figure **34 below** for the code.

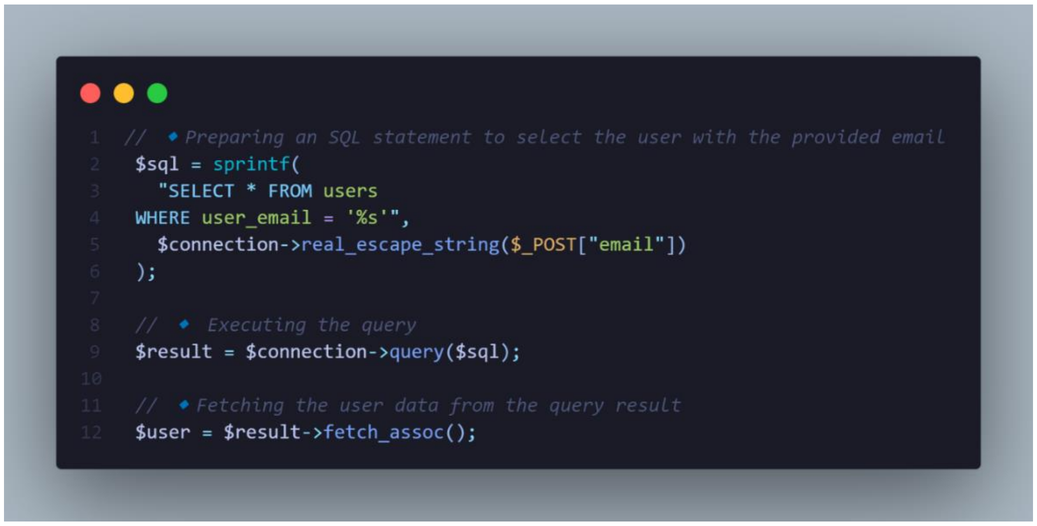


```php
// 🔹 Flag to track if the login attempt is invalid
$is_invalid = false;

// 🔹 Checking if the request method is POST and the 'login' button was clicked
if ($_SERVER["REQUEST_METHOD"] === "POST" && isset($_POST["login"])) {
  // Including the configuration file to establish database connection
  require_once "../resources/config.php";

  // 🔹 Accessing the global database connection
  global $connection;
```

Figure 34: Displaying PHP login script.

3. SQL Query and Prepared Statements (see Figure 35)

The script creates an SQL query to select a user from the 'users' table depending on the email given in the form. It utilizes real_escape_string to prevent SQL injection. If a user with the provided email address is found, the script retrieves their information.

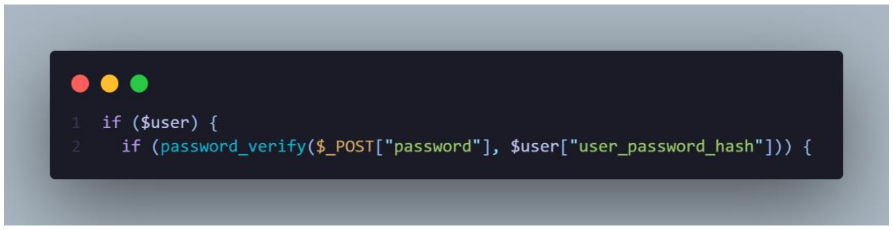


```php
// 🔹Preparing an SQL statement to select the user with the provided email
 $sql = sprintf(
   "SELECT * FROM users
 WHERE user_email = '%s'",
   $connection->real_escape_string($_POST["email"])
 );

 // 🔹 Executing the query
 $result = $connection->query($sql);

 // 🔹Fetching the user data from the query result
 $user = $result->fetch_assoc();
```

Figure 35: Displaying SQL query.

4. Password Verification:

The script utilizes the password_verify function to compare the entered password with the hashed password stored in the database to ensure secure password handling.

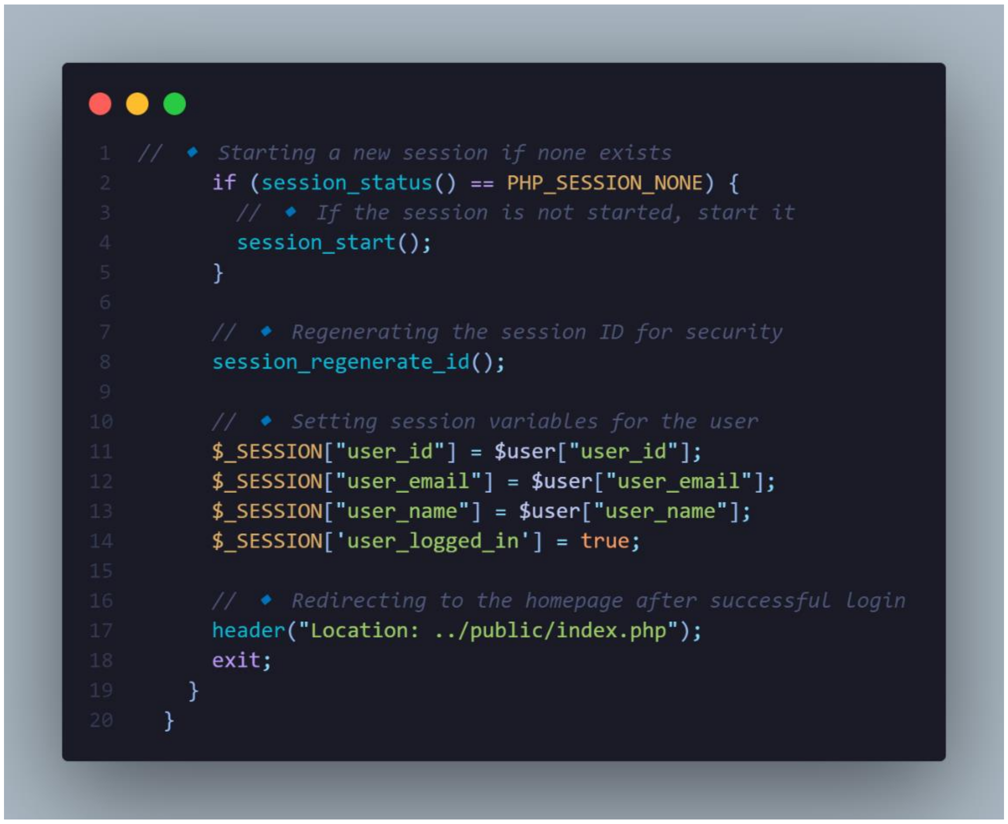


```php
if ($user) {
  if (password_verify($_POST["password"], $user["user_password_hash"])) {
```

Figure 36: Displaying code for password verification.

5. **Session Management:**

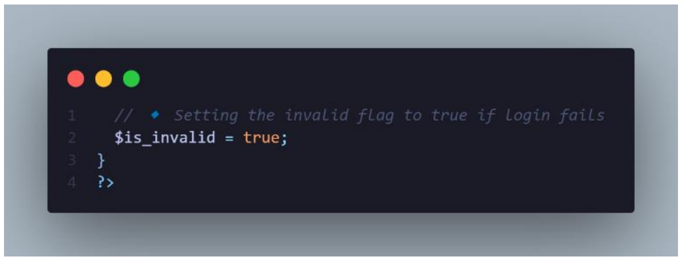
If the login is successful, the script starts a new session (if it hasn't already), populates session variables with user data, and regenerates the session ID for security. The user is subsequently redirected to the homepage, indicating that the login was successful.



```php
//  🔹 Starting a new session if none exists
      if (session_status() == PHP_SESSION_NONE) {
        //  🔹 If the session is not started, start it
        session_start();
      }

      //  🔹 Regenerating the session ID for security
      session_regenerate_id();

      //  🔹 Setting session variables for the user
      $_SESSION["user_id"] = $user["user_id"];
      $_SESSION["user_email"] = $user["user_email"];
      $_SESSION["user_name"] = $user["user_name"];
      $_SESSION['user_logged_in'] = true;

      //  🔹 Redirecting to the homepage after successful login
      header("Location: ../public/index.php");
      exit;
    }
  }
```

Figure 37: Displaying code for session management.

6. **Failed Login Handling -** If the password does not match or the user cannot be discovered, the $is_invalid flag is set to true. This parameter is used to notify users of failed login attempts by showing an error message on the form.

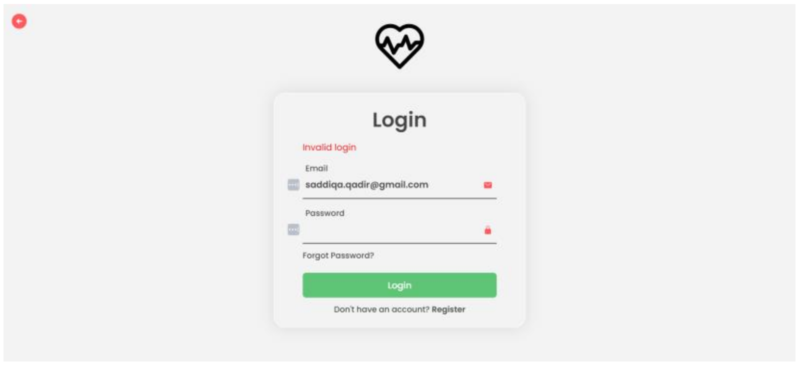


```
    // ◆ Setting the invalid flag to true if login fails
    $is_invalid = true;
  }
  ?>
```

Figure 38: Displaying code for failed login handling.

FRONT END | Let's take a closer look at how the user experiences the registration process:

1. User-Friendly Login Form - The login form provides a simple user interface. Login attempts that fail result in a visible error message. Figure 39 provides an example.



Login

Invalid login

Email: saddiqa.qadir@gmail.com

Password

Forgot Password?

Login

Don't have an account? Register

Figure 39: Displaying login form.

2. Successful login attempt - Following a successful login attempt, users get directed to the main page. Figure 40 provides one example.

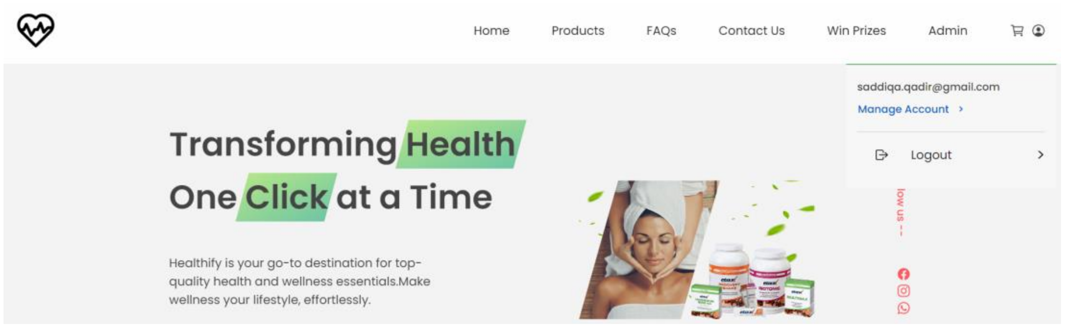


Figure 40: Displaying login success page.

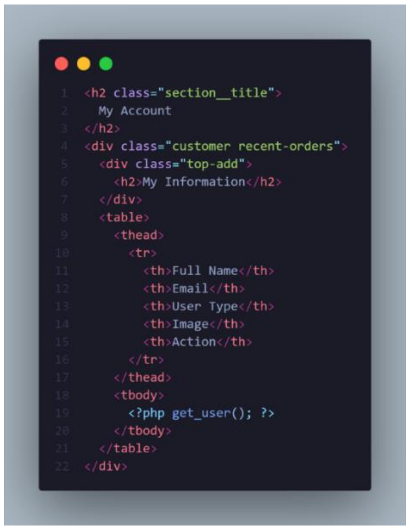
FR3 – User Profile Management

Overall Explanation of the Functionality

For this functionality, HTML is used for the webpage and contains a section named "My Account." A table will show the user data, including full name, email, user type, and picture. Additionally, it has an action column where a user can edit their information. The information about the user is retrieved from the database via the PHP function get_user. By using their email address that was saved in the session after the login process, it fetches information for the user who is presently logged in. After that, the program creates table rows dynamically for every element of user data. The table shows the name, email address, user type, placeholder image, and 'Edit' link for every user. The user can modify their account details by clicking the ‘Edit Link’ and changing their information. This configuration aims to give users an easy-to-use interface for managing their accounts, enabling them to view and modify their data.

BACK END | Let's take a closer look at the login process (How the functionality was implemented):

1. **HTML Structure (see Figure 41):** This functionality's HTML layout comprises a "My Account" part with a subsection called "My Information." A table is constructed in this section to display user data. The table headers are as follows: "Full Name," "Email," "User Type," "Image," and "Action." These headers relate to the user data that will be displayed in the table rows (tbody>). The PHP function get_user() is responsible for dynamically populating this table with user data, ensuring that the information for the currently logged-in user is acquired from the database. Each user's "Action" column has an "Edit" link, allowing users to edit their information. For managing and displaying account information for users, this structure provides an orderly and user-friendly interface.

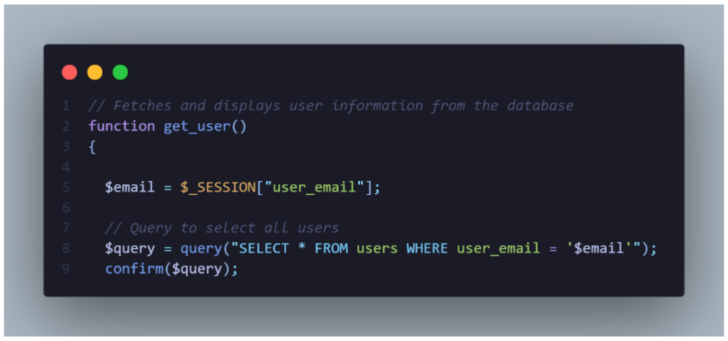


```html
<h2 class="section__title">
  My Account
</h2>
<div class="customer recent-orders">
  <div class="top-add">
    <h2>My Information</h2>
  </div>
  <table>
    <thead>
      <tr>
        <th>Full Name</th>
        <th>Email</th>
        <th>User Type</th>
        <th>Image</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      <?php get_user(); ?>
    </tbody>
  </table>
</div>
```

Figure 41: Displaying code for the layout of the account management page.

2. **PHP Function get_user() (see Figure 42):**

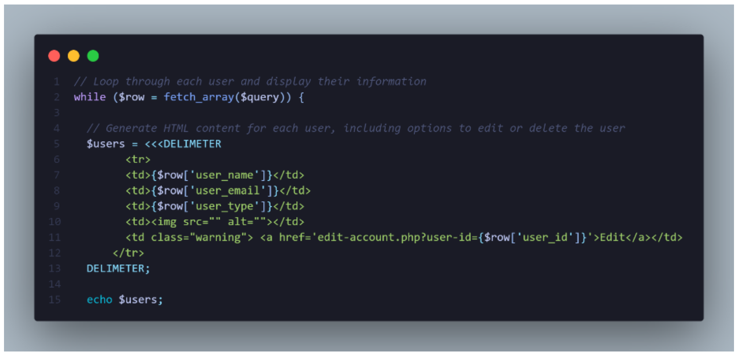
This PHP code retrieves and displays data from the database for a specified user. It begins by reading the session data for the email address of the currently logged-in user ($_SESSION["user_email"]). Using this email address, a database query is executed to retrieve all records with matching email addresses from the 'users' table.



```php
// Fetches and displays user information from the database
function get_user()
{

  $email = $_SESSION["user_email"];

  // Query to select all users
  $query = query("SELECT * FROM users WHERE user_email = '$email'");
  confirm($query);
```

Figure 42: Displaying the first section of the get_user function.

The while loop then iterates through the query result, generating HTML table rows (tr>) for each obtained user record. Each row contains the user's full name, email address, user type, and a picture placeholder. In addition, each row has a 'Edit' link that takes the user to the 'edit-account.php' page, with their user ID as a parameter in the URL. The heredoc syntax (DELIMETER) is used to construct a multi-line HTML string that is then echoed to display the user information. This function displays the logged-in user's information in a tabular manner, allowing them to update their information.

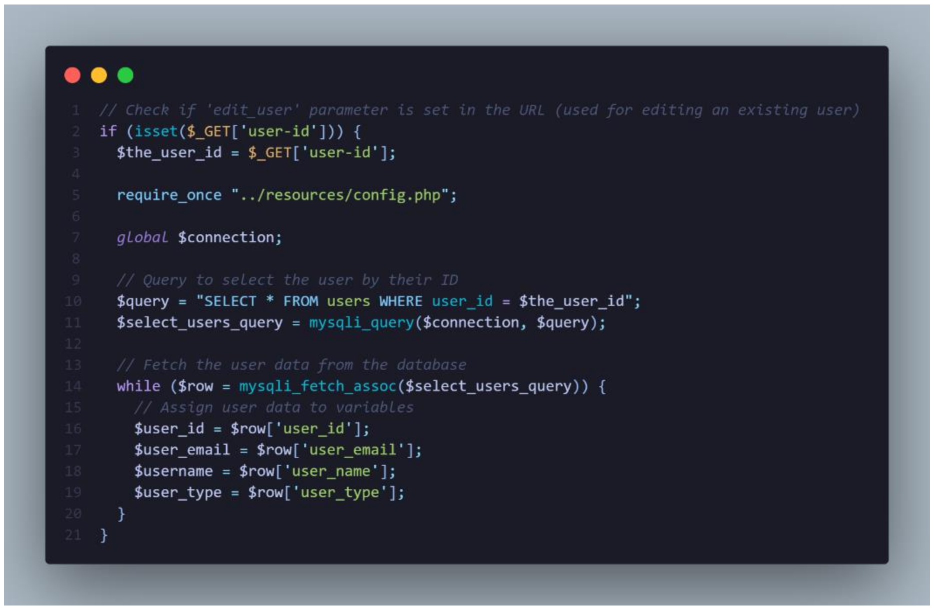


```php
// Loop through each user and display their information
while ($row = fetch_array($query)) {

  // Generate HTML content for each user, including options to edit or delete the user
  $users = <<<DELIMETER
        <tr>
        <td>{$row['user_name']}</td>
        <td>{$row['user_email']}</td>
        <td>{$row['user_type']}</td>
        <td><img src="" alt=""></td>
        <td class="warning"> <a href='edit-account.php?user-id={$row['user_id']}'>Edit</a></td>
      </tr>
  DELIMETER;

  echo $users;
```

Figure 43: Displaying the second section of the get_user function.

3. **PHP Script for Editing User Information (see Figure 44):**

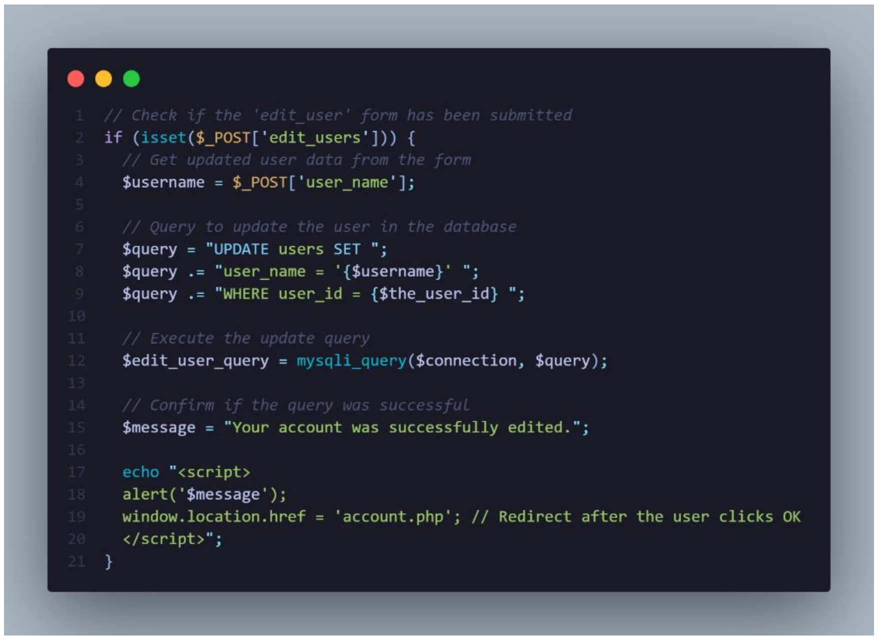
This PHP script facilitates user information editing. When a user commences the editing process, the script searches the URL for a 'user-id' argument to identify the user being edited. If it is found, it retrieves the user's existing information from the database and populates the edit form with it. Within the form, the user can change their information.



```php
// Check if 'edit_user' parameter is set in the URL (used for editing an existing user)
if (isset($_GET['user-id'])) {
  $the_user_id = $_GET['user-id'];

  require_once "../resources/config.php";

  global $connection;

  // Query to select the user by their ID
  $query = "SELECT * FROM users WHERE user_id = $the_user_id";
  $select_users_query = mysqli_query($connection, $query);

  // Fetch the user data from the database
  while ($row = mysqli_fetch_assoc($select_users_query)) {
    // Assign user data to variables
    $user_id = $row['user_id'];
    $user_email = $row['user_email'];
    $username = $row['user_name'];
    $user_type = $row['user_type'];
  }
}
```

Figure 44: Displaying the PHP Script for Displaying the User Information in the Edit Form.

Following the submission of the modified information, the script uses a SQL query to update the user's details in the database's 'users' table. When this update is completed properly, it displays a confirmation message and returns the user to the account page. This setting allows users to edit their personal information seamlessly, improving the user experience.

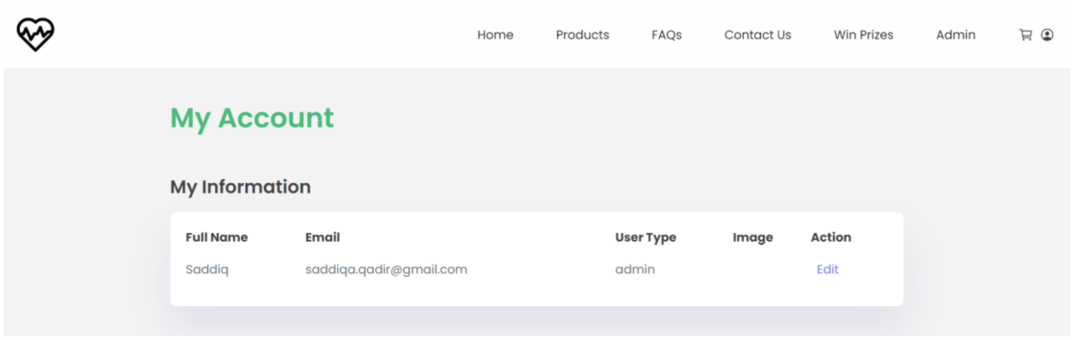


```php
// Check if the 'edit_user' form has been submitted
if (isset($_POST['edit_users'])) {
  // Get updated user data from the form
  $username = $_POST['user_name'];

  // Query to update the user in the database
  $query = "UPDATE users SET ";
  $query .= "user_name = '{$username}' ";
  $query .= "WHERE user_id = {$the_user_id} ";

  // Execute the update query
  $edit_user_query = mysqli_query($connection, $query);

  // Confirm if the query was successful
  $message = "Your account was successfully edited.";

  echo "<script>
  alert('$message');
  window.location.href = 'account.php'; // Redirect after the user clicks OK
  </script>";
}
```

Figure 45: Displaying PHP Script for Editing User Information.

FRONT END | Let's take a closer look at how the user experiences the registration process:

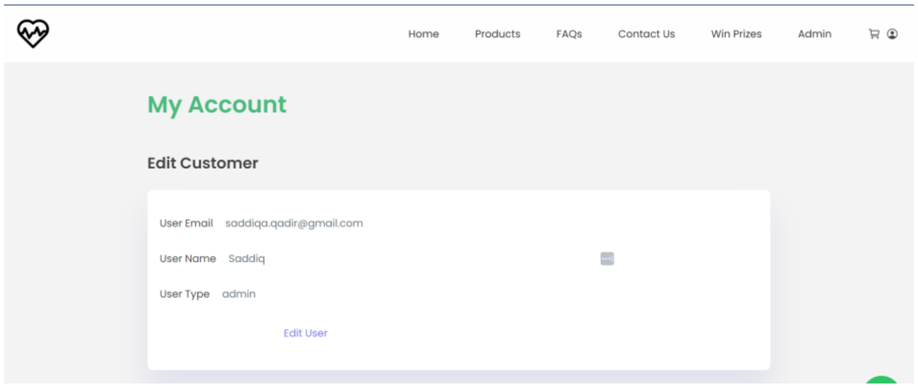
1. **Viewing Account Information:** The image below showcases the structured presentation of the user account information, making it easy for users to review and stay updated.



Full Name	Email	User Type	Image	Action
Saddiq	saddiqa.qadir@gmail.com	admin		Edit

Figure 46: Displaying the 'My Account' page.

2. **Editing Information:** Clicking the "Edit" like on the image above page leads you to edit the user form displayed below where users can conveniently modify their information.



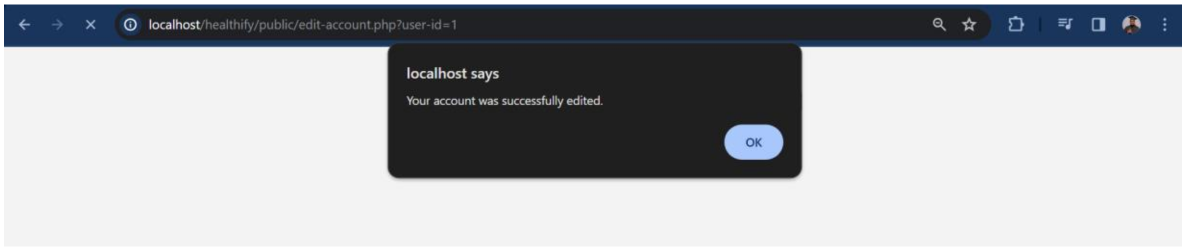
User Email saddiqa.qadir@gmail.com

User Name Saddiq

User Type admin

Figure 47: Displaying Edit Form.

3. **Success Message:** After editing your details, users will receive a confirmation message, indicating that the changes have been successfully saved. See image 48 below for an example.

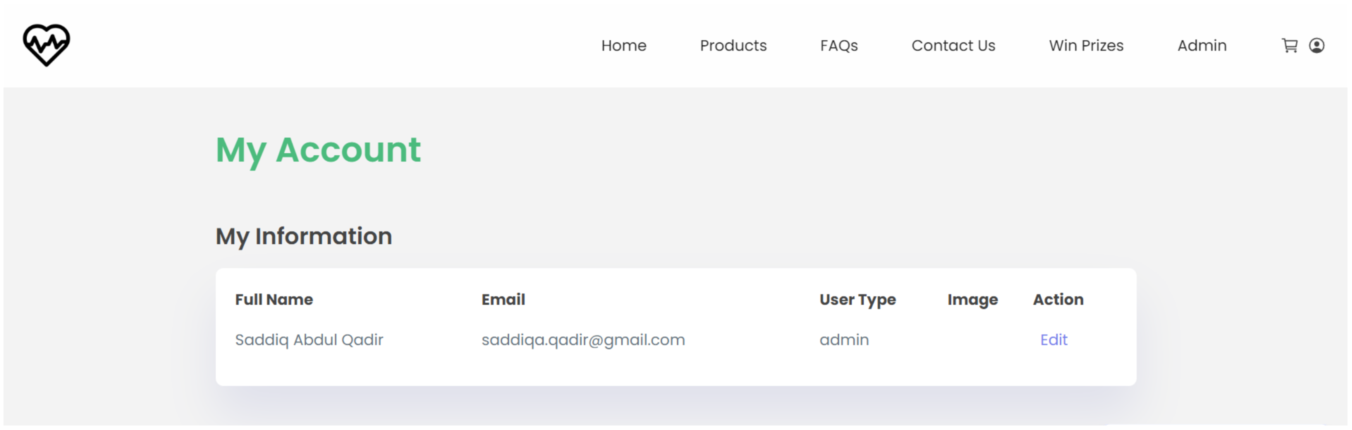

localhost/healthify/public/edit-account.php?user-id=1

localhost says

Your account was successfully edited.

Figure 48: Displaying success message.

4. **Updated Account Information:** When the user clicks "OK" on the success message, they'll see their updated account information, reflecting the changes they made.



My Account

My Information

Full Name	Email	User Type	Image	Action
Saddiq Abdul Qadir	saddiqa.qadir@gmail.com	admin		Edit

Figure 49: Displaying updated information.

Conclusion

The Account Management feature allows users to easily view and edit their account information. It provides a user-friendly interface for personalizing user data by employing PHP scripts and dynamic HTML generation. Users can simply change their information, which improves their interaction with the website.

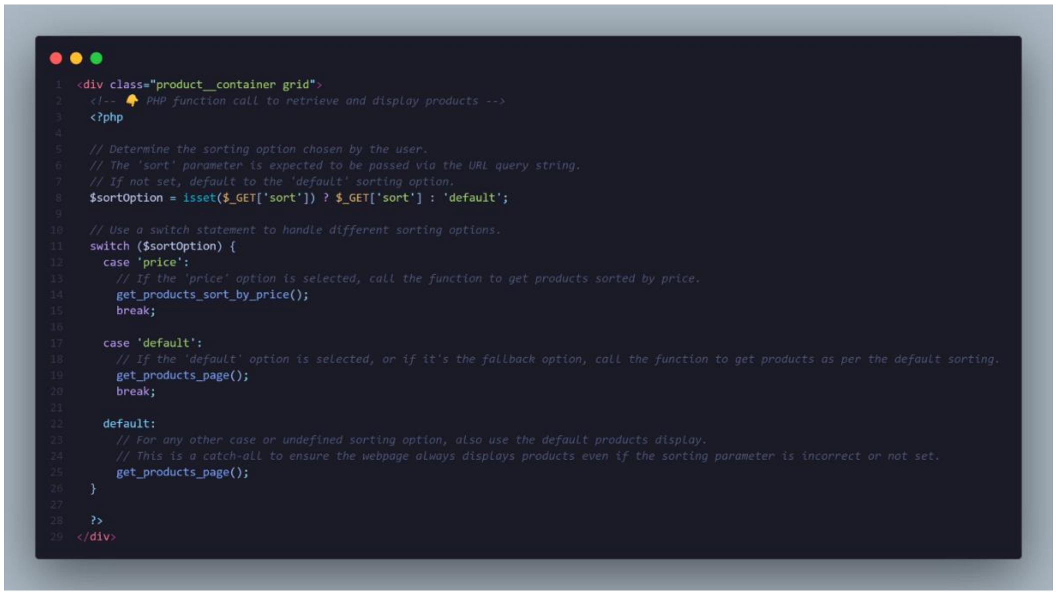
FR4 – Product Catalogue

Overall Explanation of the Functionality

This section will look at implementing the product catalogue functionality. In this function, products are dynamically displayed according to user-selected sorting options or default settings. It uses PHP to retrieve product information from a database; the default product presentation is taken care of by a get_products_page() function. This configuration makes sure that products are presented in an orderly manner and includes add-to-cart options, names, prices, and photos to improve the user's browsing experience.

BACK END | Let's take a closer look at the product catalogue functionality (How the functionality was implemented):

Figure 50 below illustrates part of the code needed to implement a product catalogue. Based on a sorting choice, the PHP code within the div element dynamically fetches and displays products to users. If the user does not select a specific sorting option, or if the option selected does not meet any established criteria, the page defaults to a conventional way of showing products. A function called get_products_page () manages the default display. The main objective of the code is to provide products to consumers in an orderly manner and to provide them with a means to see the products based on their preferences, thereby improving the browsing experience on the website.



```
<div class="product__container grid">
  <!-- 👇 PHP function call to retrieve and display products -->
  <?php

  // Determine the sorting option chosen by the user.
  // The 'sort' parameter is expected to be passed via the URL query string.
  // If not set, default to the 'default' sorting option.
  $sortOption = isset($_GET['sort']) ? $_GET['sort'] : 'default';

  // Use a switch statement to handle different sorting options.
  switch ($sortOption) {
    case 'price':
      // If the 'price' option is selected, call the function to get products sorted by price.
      get_products_sort_by_price();
      break;

    case 'default':
      // If the 'default' option is selected, or if it's the fallback option, call the function to get products as per the default sorting.
      get_products_page();
      break;

    default:
      // For any other case or undefined sorting option, also use the default products display.
      // This is a catch-all to ensure the webpage always displays products even if the sorting parameter is incorrect or not set.
      get_products_page();
  }

  ?>
</div>
```

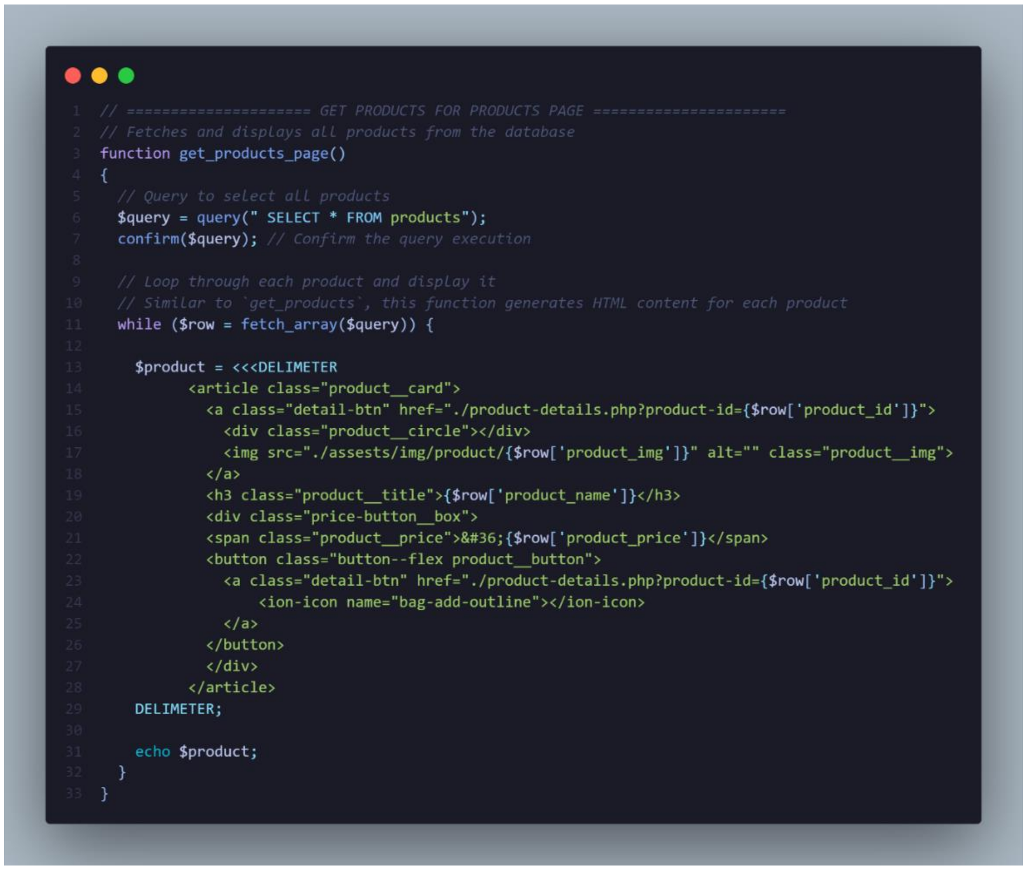
Figure 50: Displaying code for product catalogue.

Product Retrieval and Display with get_products_page() Function:

In the code displayed in figure 51 below the get_products_page PHP function is intended to retrieve and show all products on the product catalog page. It begins by running a database query to retrieve all entries from the ‘products' table. Following the execution of the query, the function loops through each product retrieved from the database.

The function generates a block of HTML code for each product, resulting in a visual representation of the product on the product catalogue page. Each product has an article element (article>) with a clickable image (leading to the product details), the product name, and its price. There's also a button with an icon that adds the product to the shopping cart.

This function works in conjunction with the previously mentioned code snippet. While the previous code handles product sorting depending on user decisions (such as by price or default), this get_products_page function is in charge of retrieving and displaying product data from the database independent of the sorting option. It ensures that the products are always presented to the user with all the necessary details and an appealing layout.



```php
// ===================== GET PRODUCTS FOR PRODUCTS PAGE ======================
// Fetches and displays all products from the database
function get_products_page()
{
  // Query to select all products
  $query = query(" SELECT * FROM products");
  confirm($query); // Confirm the query execution

  // Loop through each product and display it
  // Similar to `get_products`, this function generates HTML content for each product
  while ($row = fetch_array($query)) {

    $product = <<<DELIMETER
          <article class="product__card">
            <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
              <div class="product__circle"></div>
              <img src="./assests/img/product/{$row['product_img']}" alt="" class="product__img">
            </a>
            <h3 class="product__title">{$row['product_name']}</h3>
            <div class="price-button__box">
            <span class="product__price">&#36;{$row['product_price']}</span>
            <button class="button--flex product__button">
              <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
                  <ion-icon name="bag-add-outline"></ion-icon>
              </a>
            </button>
            </div>
          </article>
    DELIMETER;

    echo $product;
  }
}
```

Figure 51: Displaying get_products_page() function.

FRONT END | Let's take a closer look at how the user experiences the product catalogue:

A preview of the product catalogue, where users can browse a vast array of products, is provided in the image below:

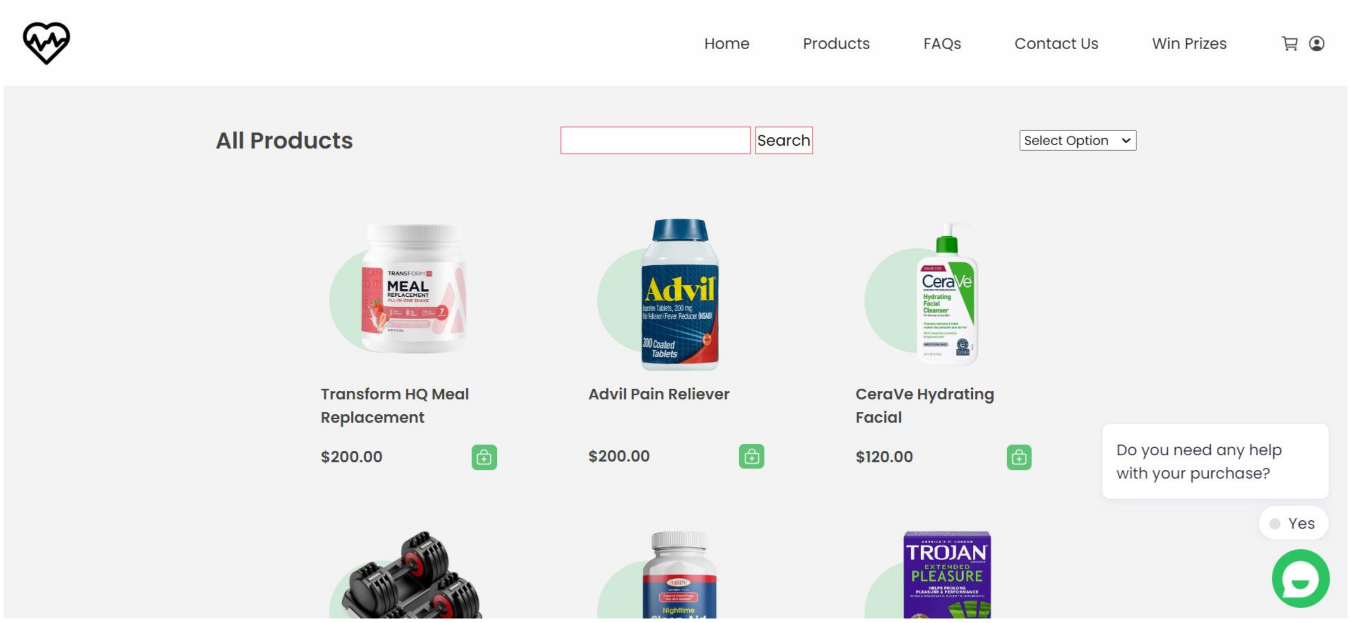


Figure 52: Displaying product catalogue.

FR5 – Search Functionality

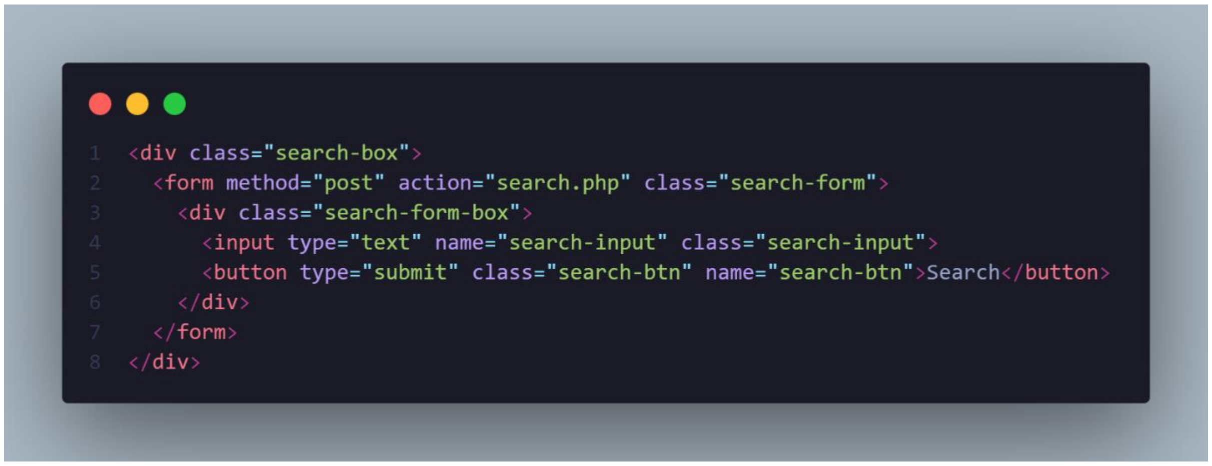
Overall Explanation of the Functionality

In this section the product page's search functionality is implemented. Where users submit search phrases into a search bar, and the appropriate file "search.php" uses POST to process the queries. The PHP script looks for matches in the product names or descriptions in the 'products' table. Then, after creating HTML for each item using a loop, it shows related products along with information like names, prices, and photos. This functionality allows users to search for products effectively and specifically.

BACK END | Let's take a closer look at search process (How the functionality was implemented):

1. HTML Search Form:

The HTML structure for the search functionality includes a search form where users can submit their search queries. The form is meant to send user-entered search queries to a file called "search.php" via the POST method. Users can enter their search query into an input field on the form. This input is part of a div element that is styled as a 'search-form-box,' which aids in the layout and design of the search box. Along with the input area, there is a submit button labelled "Search." When a user enters text into the input field and presses this button, the form transmits the search query (the text supplied by the user) to "search.php" for processing. In Figure 53, the code below presents the html form for the search functionality.



```html
<div class="search-box">
  <form method="post" action="search.php" class="search-form">
    <div class="search-form-box">
      <input type="text" name="search-input" class="search-input">
      <button type="submit" class="search-btn" name="search-btn">Search</button>
    </div>
  </form>
</div>
```

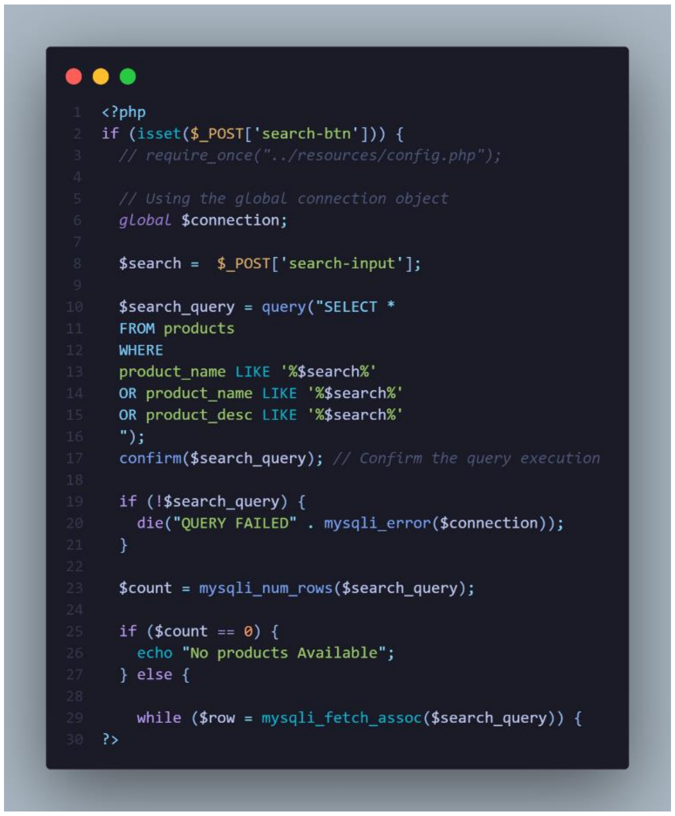
Figure 53: Displaying search form.

2. PHP Script for Search (search.php):

The PHP script is activated when a user enters a search phrase into the search box we discussed earlier. It initially verifies the presence of the 'search-btn' POST variable to see if the search form has been submitted. The user's search input is then retrieved from the 'search-input' box.

The script creates a SQL query to search the database's 'products' table for products using the search phrase entered by the user in the product name or description. This is accomplished by using the LIKE operator in conjunction with wildcards (%) for incomplete matches. The script checks for query success and counts the number of matching products found after executing the search query.

If no products are found ($count == 0), a message indicating that no products are available is displayed. Otherwise, it enters a loop to retrieve and display each matched product using mysqli_fetch_assoc, which will include producing HTML to display these products on the page. This search feature improves the user experience by allowing them to rapidly identify products that correspond to their individual interests or demands. The PHP code in figure 54 below handles the product page's search capability.



```php
<?php
if (isset($_POST['search-btn'])) {
  // require_once("../resources/config.php");

  // Using the global connection object
  global $connection;

  $search =  $_POST['search-input'];

  $search_query = query("SELECT *
  FROM products
  WHERE
  product_name LIKE '%$search%'
  OR product_name LIKE '%$search%'
  OR product_desc LIKE '%$search%'
  ");
  confirm($search_query); // Confirm the query execution

  if (!$search_query) {
    die("QUERY FAILED" . mysqli_error($connection));
  }

  $count = mysqli_num_rows($search_query);

  if ($count == 0) {
    echo "No products Available";
  } else {

    while ($row = mysqli_fetch_assoc($search_query)) {
?>
```

Figure 54: Displaying the PHP script for the search function.

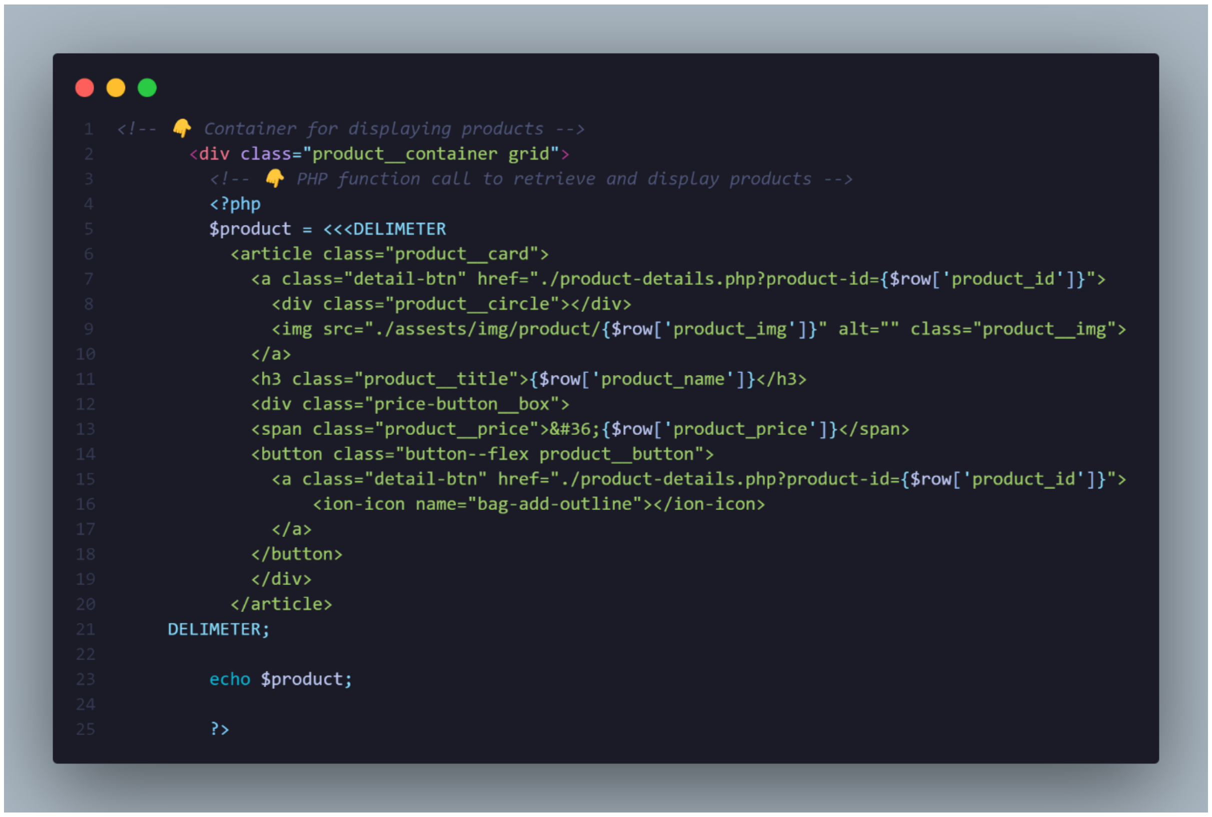
3. Search Results Display:

The code provided in Figure 55 below includes HTML and PHP code that is part of the search functionality and presents products in a visually appealing style. It is used within a loop that retrieves product information from the database, similar to the previous PHP search script.

The area where the products will be shown is the div> with the class product__container grid. PHP is utilized within this container to dynamically generate a card for each product. This is accomplished by using a heredoc syntax (DELIMETER) to specify a block of HTML, which is a useful method for embedding multi-line HTML code within PHP.

Each product card (article class="product__card">) has a clickable image that takes you to a detailed product page (product-details.php), which is identifiable by a unique product ID given in the URL. Each product becomes interactive, allowing users to learn more about it. The product picture, name, and price are dynamically filled in using database data ($row['product_img'], $row['product_name'], and $row['product_price']). There's also a button with an icon, which is for adding the product to a shopping cart.

Overall, this code complements the product search capability. This layout is used to display each product on the page after the search query is processed and goods are retrieved, allowing visitors to browse and interact with the search results.



```
<!-- 👇 Container for displaying products -->
       <div class="product__container grid">
         <!-- 👇 PHP function call to retrieve and display products -->
         <?php
         $product = <<<DELIMETER
           <article class="product__card">
             <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
               <div class="product__circle"></div>
               <img src="./assests/img/product/{$row['product_img']}" alt="" class="product__img">
             </a>
             <h3 class="product__title">{$row['product_name']}</h3>
             <div class="price-button__box">
             <span class="product__price">&#36;{$row['product_price']}</span>
             <button class="button--flex product__button">
               <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
                   <ion-icon name="bag-add-outline"></ion-icon>
               </a>
             </button>
             </div>
           </article>
     DELIMETER;

         echo $product;

         ?>
```

Figure 55: Displaying code that is used to display the search results.

FRONT END | Let's take a closer look at how the user experiences the search process:

1. Product Search:

The image below captures the essence of the product search experience. The user initiates the process by inputting their search query into the search bar. An example of this is shown in the image below with the search query 'face'.

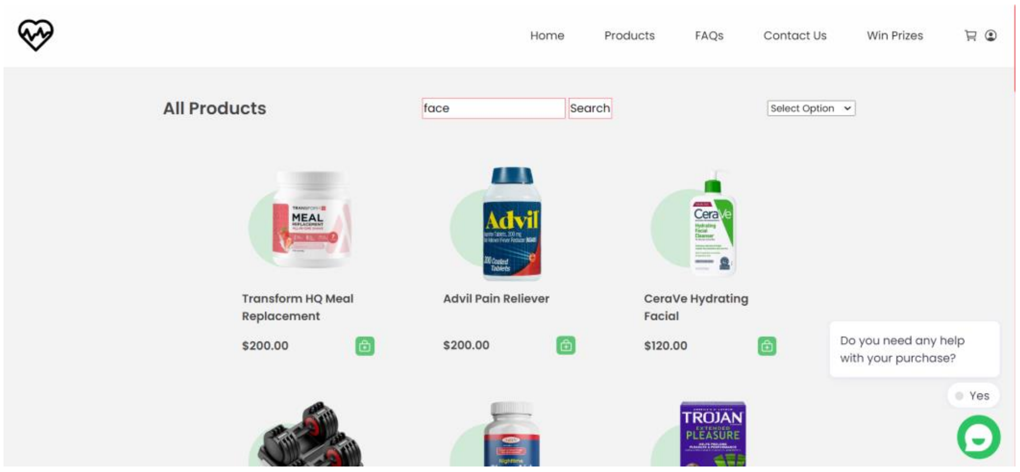


Figure 56: Displaying products.

2. Search Results:

After inputting a search and query and clicking the search button. Users are presented with the results of their search query which are the products related to their search. The image below displays a visually appealing and well-organized design, displaying the result of the user search query (face).

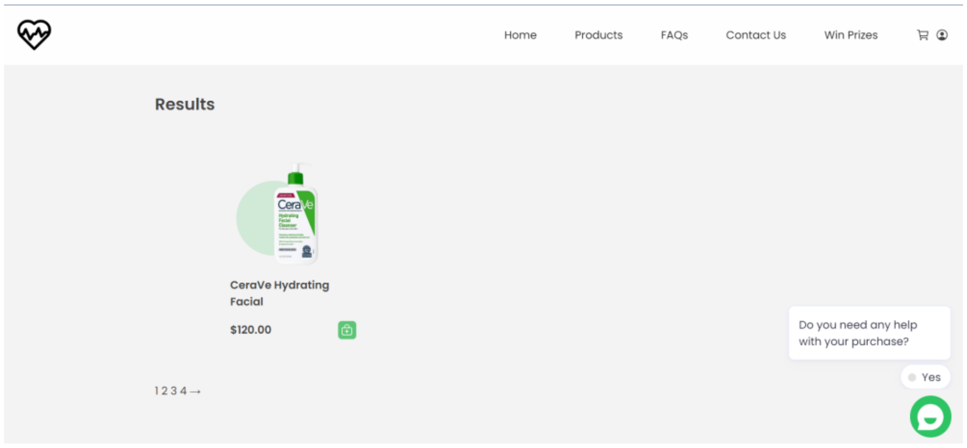


Figure 57: Displaying search results.

FR6 – Product Filtering

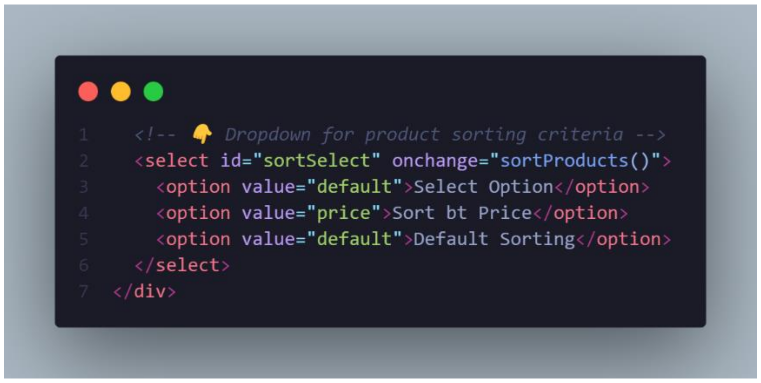
Overall Explanation of the functionality

The section looks at implementing the functionality for product filtering. It will consist of a dropdown menu for product sorting on a display page, including price sorting. SortProducts() is activated by user selections, which updates the display. PHP code will be implemented to decode these selections and fetch and display products according to user preferences for a better browsing experience. It does this by using functions like get_products_sort_by_price() or get_products_page().

BACK END | Let's take a closer look at product filtering process (How the functionality was implemented):

1. Navigating the Sort

This code displayed in Figure 58 below is designed to provide a filtering option for the product display page. There is a dropdown menu (select>) at the top with many filtering options for the user to pick from, such as filtering by price or a default filtering method. When a user selects an option, the sortProducts() function is called, which is intended to update the page with the sorting criteria that the user has chosen.

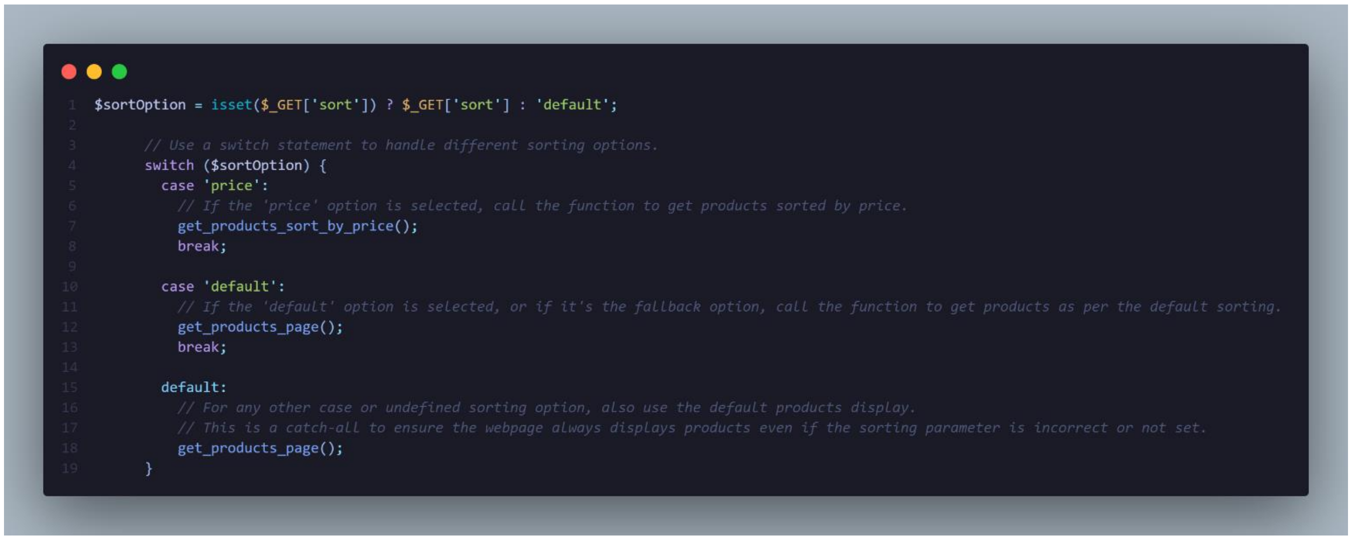


```html
  <!-- 👇 Dropdown for product sorting criteria -->
  <select id="sortSelect" onchange="sortProducts()">
    <option value="default">Select Option</option>
    <option value="price">Sort bt Price</option>
    <option value="default">Default Sorting</option>
  </select>
</div>
```

Figure 58: Displaying code to navigate the sort options.

2. Efficient Product Sorting: Deciphering User Choices:

The process for sorting the products based on the user's choices is handled by the code below the dropdown menu. It determines whether a sorting option ('sort') is specified in the query string of the URL. If it is set, the code employs this option; else, it uses a conventional sorting mechanism. The switch statement then decides which PHP function to invoke based on the sorting option selected. The get_products_sort_by_price() function is invoked if the user chooses to sort by price. The get_products_page() function is invoked if the default option is selected or if no suitable sorting option is provided. These functions are in charge of retrieving and displaying the products in the specified order, thereby improving the user experience by allowing them to browse products based on their preferences.



```php
$sortOption = isset($_GET['sort']) ? $_GET['sort'] : 'default';

      // Use a switch statement to handle different sorting options.
      switch ($sortOption) {
        case 'price':
          // If the 'price' option is selected, call the function to get products sorted by price.
          get_products_sort_by_price();
          break;

        case 'default':
          // If the 'default' option is selected, or if it's the fallback option, call the function to get products as per the default sorting.
          get_products_page();
          break;

        default:
          // For any other case or undefined sorting option, also use the default products display.
          // This is a catch-all to ensure the webpage always displays products even if the sorting parameter is incorrect or not set.
          get_products_page();
      }
```

Figure 59: Displaying code to decipher user choices.

FRONT END | Let's take a closer look at how the user experiences the search process:

The image below takes us into the world of streamlined shopping, where products are automatically arranged by price to improve the user's browsing experience. This feature allows users to easily navigate through a beautifully organized display of products, each diligently organized according to its price point.

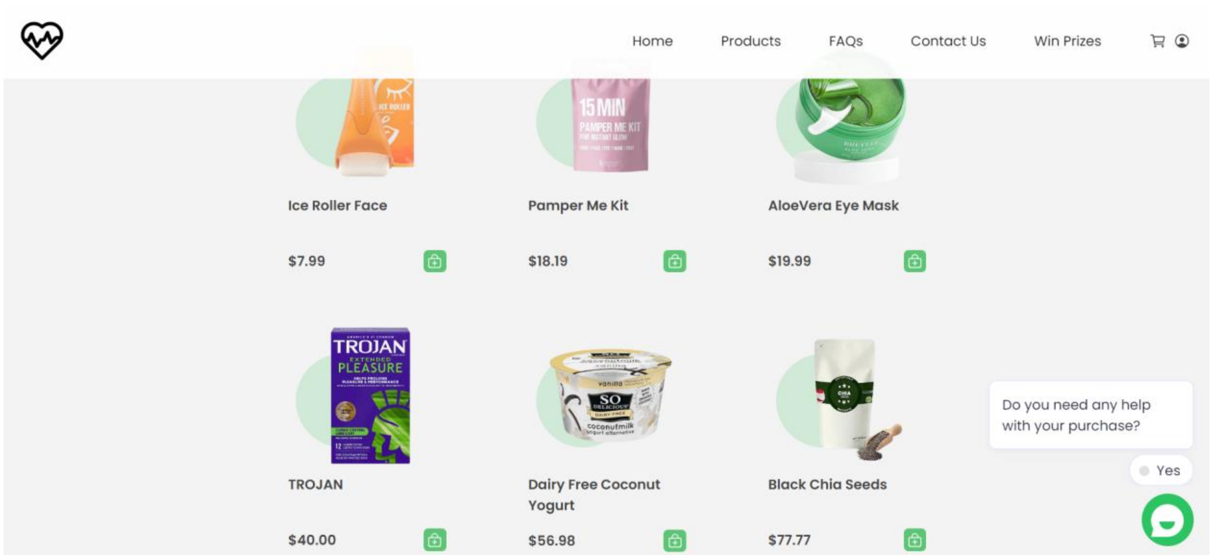


Figure 60: Displaying front end with products filtered by price.

FR7 – Product Detail Page

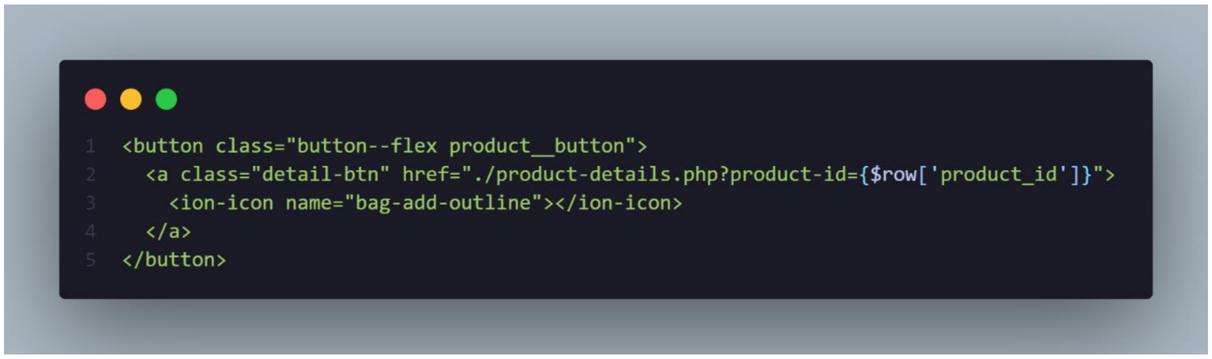
Overall Explanation of the Functionality

This section will look at the product details page functionality. The feature for product details improves how users interact with products on websites. It has an elegant button that takes the user to a page with product details and a URL parameter that contains the product's unique ID. With the help of an icon built inside the button, visitors may browse to comprehensive details about each product. The product's image, name, category, price, and description are among the details that are retrieved and displayed from the database by the accompanying PHP script. By providing users with detailed product information and the ability to add items to their cart, this configuration enhances their shopping experience.

BACK END | Let's take a closer look at the product detail page functionality (How the functionality was implemented):

1. Exploration Button: Unlocking Product Details

An appealing and unique button incorporated perfectly into the product display serves as the key to obtaining extensive product details. The button is stylized with the classes button--flex and product__button, which gives it a distinct look and function. A link (a> tag) styled as detail-btn is contained within the button and connects to a product details page (product-details.php). This link has a query parameter (product-id=$row['product_id']) that dynamically transmits the unique ID of the product currently being displayed. This ID is obtained from the database and is unique to each product. The button also includes an icon (ion-icon name="bag-add-outline">/ion-icon> that indicates an action, adding the product to a shopping cart. When a user hits this button, they are directed to the product's detailed page, where they can view additional information about it or conduct actions such as purchasing or adding it to their cart. This code displayed in figure 61 below adds a button to the product display page that allows users to interact with specific products.

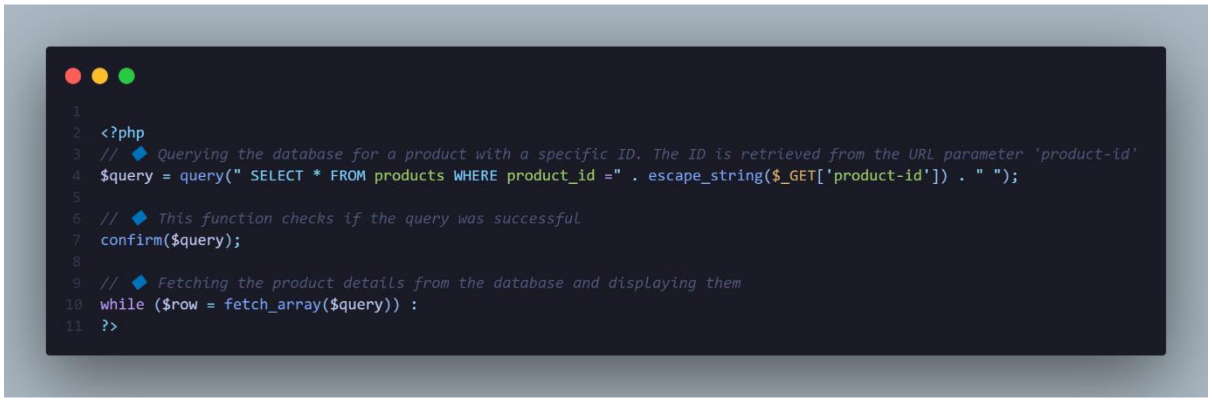


```
<button class="button--flex product__button">
  <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
    <ion-icon name="bag-add-outline"></ion-icon>
  </a>
</button>
```

Figure 61: Displaying code for product details button.

2. PHP Script: Retrieving Product Details

A well-crafted PHP script performs the magic behind the scenes. The script retrieves and displays product information from the database. It begins by building a SQL query to select a product from the 'products' table whose ID matches the ID supplied in the URL parameter ('product-id'). To prevent potential SQL injection attacks, the script uses escape_string to sanitize the input received from the URL. After executing the query, the script uses the confirm method to ensure its success. It then enters a loop to retrieve and handle the query result using fetch_array. The product information (retrieved as an array from the database) are displayed within this loop. This script is used on the product details page to dynamically fetch, and display product information based on the ID provided by the user's selection or navigation. This PHP script displayed in figure 62 below is the PHP script referred to above.

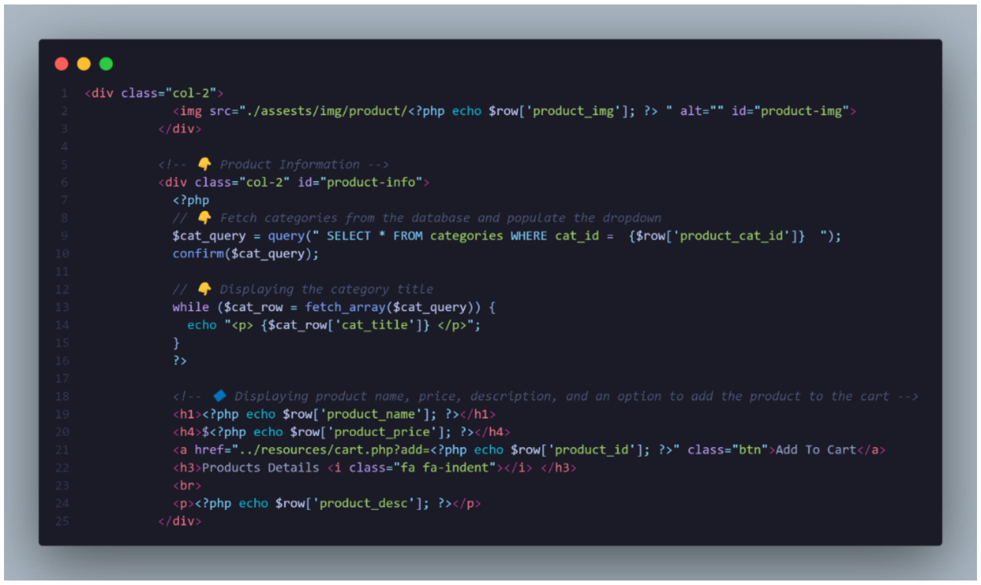


```

<?php
// 🔷 Querying the database for a product with a specific ID. The ID is retrieved from the URL parameter 'product-id'
$query = query(" SELECT * FROM products WHERE product_id =" . escape_string($_GET['product-id']) . " ");

// 🔷 This function checks if the query was successful
confirm($query);

// 🔷 Fetching the product details from the database and displaying them
while ($row = fetch_array($query)) :
?>
```

Figure 62: Displaying script to retrieve product details.

3. **Visualizing Product Details:**

The code displayed in figure 63 below is a thorough component of a product details display process on the website, and it works in tandem with the previously stated product detail fetching script. The first section of code displays the product's image in a col-2 division, with the image source dynamically set using product data retrieved from the database, namely the image file path. The second section then takes an in-depth look at the product's features. It begins by searching the database for the product's category using its category ID, and then displays that category. Furthermore, the product's name, price, and a detailed description are given, all of which are taken from the database, guaranteeing that the information displayed is specific to the selected product. Furthermore, there is a functional button linked to the cart system that allows users to add the products to their cart, increasing the page's interactivity. This style not only provides a visual depiction of the product, but it also improves the user experience by providing detailed and actionable product information.

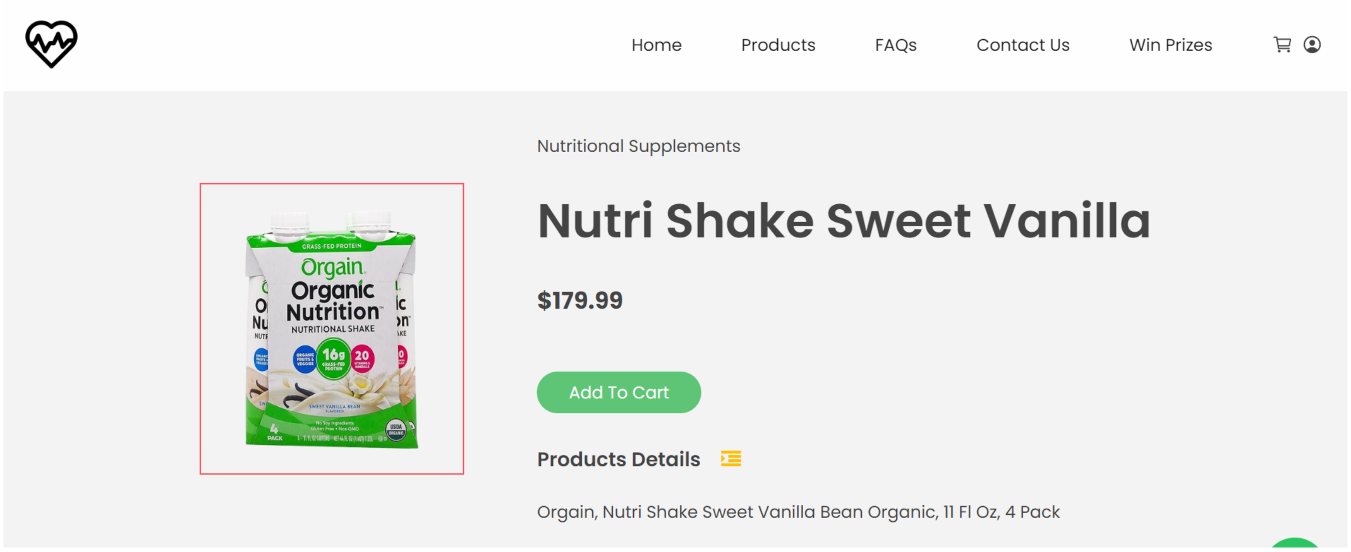


```
<div class="col-2">
            <img src="./assests/img/product/<?php echo $row['product_img']; ?> " alt="" id="product-img">
          </div>

          <!-- 👇 Product Information -->
          <div class="col-2" id="product-info">
            <?php
            // 👇 Fetch categories from the database and populate the dropdown
            $cat_query = query(" SELECT * FROM categories WHERE cat_id =  {$row['product_cat_id']}  ");
            confirm($cat_query);

            // 👇 Displaying the category title
            while ($cat_row = fetch_array($cat_query)) {
              echo "<p> {$cat_row['cat_title']} </p>";
            }
            ?>

            <!-- 🔷 Displaying product name, price, description, and an option to add the product to the cart -->
            <h1><?php echo $row['product_name']; ?></h1>
            <h4>$<?php echo $row['product_price']; ?></h4>
            <a href="../resources/cart.php?add=<?php echo $row['product_id']; ?>" class="btn">Add To Cart</a>
            <h3>Products Details <i class="fa fa-indent"></i> </h3>
            <br>
            <p><?php echo $row['product_desc']; ?></p>
          </div>
```

Figure 63: Displaying code to show product details.

FRONT END | Let's take a closer look at how the user experiences the product details of a product:

In the image below, showcased is how the website improves the buying experience by providing detailed product information. This image provides an exclusive look at the user-friendly and informative product details page, which includes captivating images, extensive descriptions, prices, and the ability to add products to the cart.



Nutritional Supplements

Nutri Shake Sweet Vanilla

$179.99

Products Details

Orgain, Nutri Shake Sweet Vanilla Bean Organic, 11 Fl Oz, 4 Pack

Figure 64: Displaying details of one product.

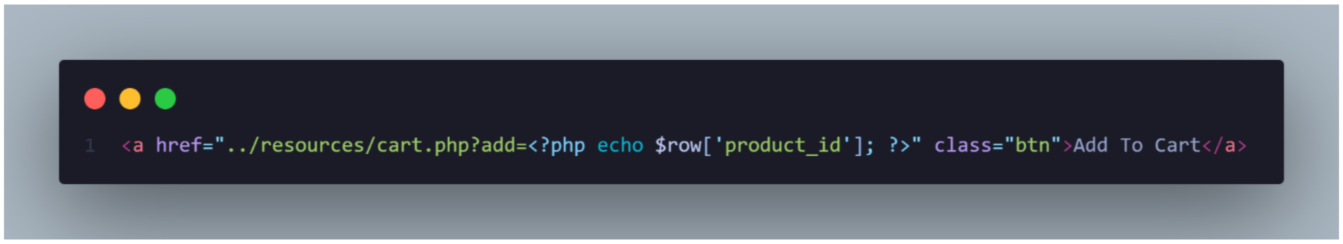
FR8 – Add to Cart

Overall Explanation of the Functionality

This section will look at the add-to-cart functionality. An "Add to Cart" button would be displayed on a website's product detail page. It instantly adds the chosen product to the user's cart when clicked, taking them to the shopping cart page. This button makes buying simple by enabling customers to select and gather all the products they want to purchase.

BACK END | Let's take a closer look at the Add to Cart functionality (How the functionality was implemented):

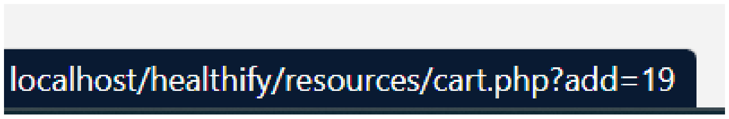
This line of code displayed in fgure 65 below adds a button to a website's product details page, allowing users to add products to their shopping cart. It's an anchor () tag with the class btn styled like a button. The anchor tag's href property is set to point to the 'cart.php' page in the '../resources/' directory. The URL has a query parameter add, which is dynamically set to the ID of the product being viewed (). This product ID is obtained from the database and included in the URL. When a user clicks the "Add To Cart" button, the browser goes to 'cart.php' and passes the product ID along. This ID is then used by the 'cart.php' script to add the exact product to the user's shopping cart. This functionality is critical in the e-commerce website, allowing customers to select products they want to purchase and place them in a shopping cart for checkout.



```
<a href="../resources/cart.php?add=<?php echo $row['product_id']; ?>" class="btn">Add To Cart</a>
```

Figure 65: Displaying add to cart button.

The image below represents the link that displays, when a user hovers over the add to cart button, once the users click this button the product with the ID 19 would be added to the cart.



localhost/healthify/resources/cart.php?add=19

Figure 66: Displaying add to cart link.

FR9 – Shopping Cart Management

Overall Explanation of the functionality

This section will cover the shopping cart management functionality. The shopping cart management feature enables customers to add, remove, or edit items from their carts. The script updates the session quantity and reroutes to the checkout page when a user adds a product. It reduces the product quantity for removal and modifies the items of the cart appropriately. When a product is deleted, the cart is updated, and the session quantity is set to zero. With dynamic cart management features, the script also computes and shows the total cost and number of products in the cart, improving the shopping experience.

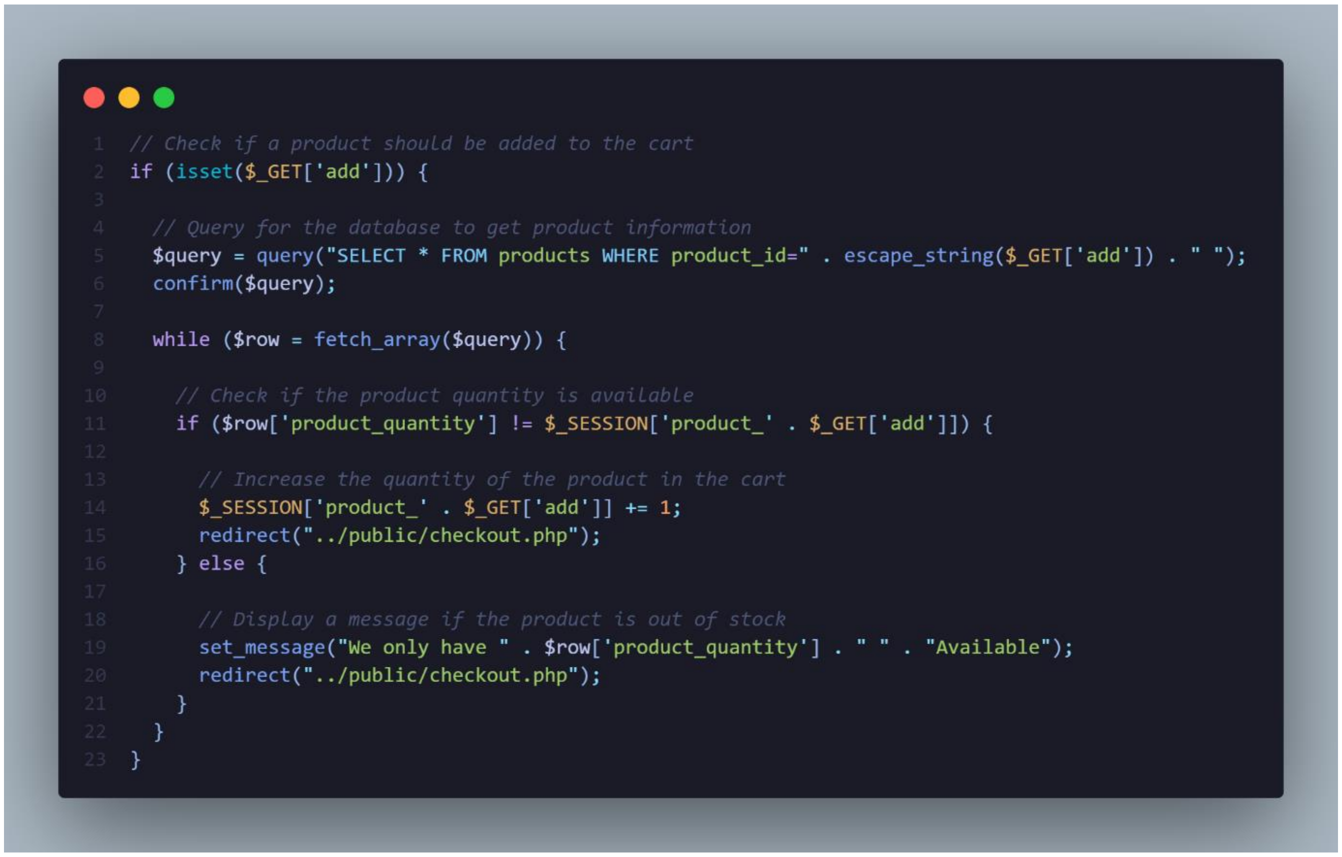
BACK END | Let's take a closer look at the shopping cart management functionality (How the functionality was implemented):

1. Adding Products to the Cart:

The PHP script displayed in figure 67 below manages the adding products to a user's cart functionality. This script is activated when a user clicks the "Add To Cart" button on a product detail page. It initially checks the URL's query string for the presence of the 'add' parameter, which indicates that a product defined by its ID is to be added to the cart.

When the script receives confirmation that a product is being added, it uses the product ID from the URL to query the database to obtain details about that particular product. In order to clean up the input and stop SQL injection, the escape_string method is utilized. The script handles the retrieved product data once it has run the query and verified its success.

The script then determines if the chosen product's quantity is in stock. The amount of each product in the cart is tracked using a session variable ($_SESSION['product_'. $_GET['add']]). If the product is available, the customer is redirected to the checkout page and the amount of that product is increased in the session (basically adding it to the cart). In addition to redirecting to the payment page, the product sets a message to notify the user of its restricted availability if it is out of stock.

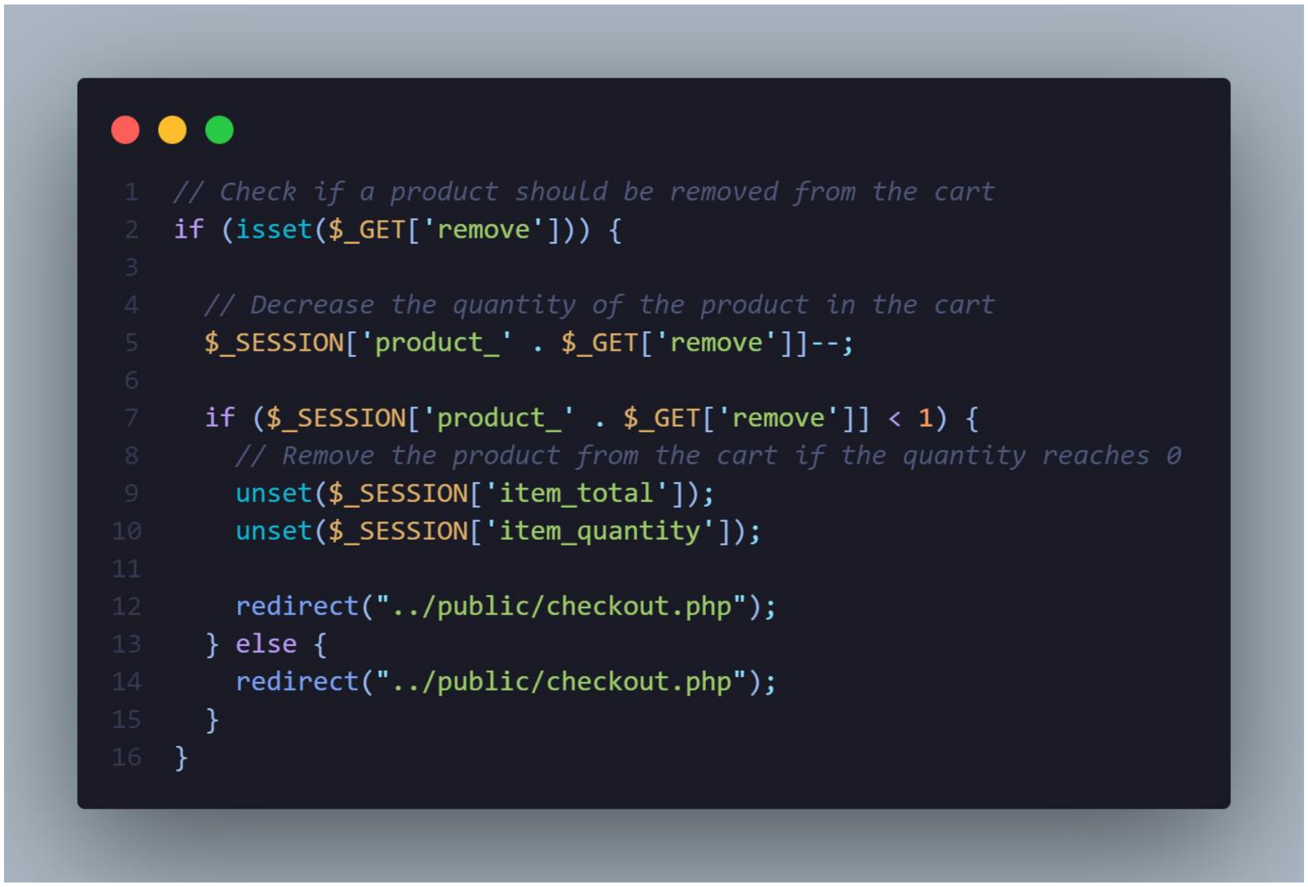


```php
// Check if a product should be added to the cart
if (isset($_GET['add'])) {

  // Query for the database to get product information
  $query = query("SELECT * FROM products WHERE product_id=" . escape_string($_GET['add']) . " ");
  confirm($query);

  while ($row = fetch_array($query)) {

    // Check if the product quantity is available
    if ($row['product_quantity'] != $_SESSION['product_' . $_GET['add']]) {

      // Increase the quantity of the product in the cart
      $_SESSION['product_' . $_GET['add']] += 1;
      redirect("../public/checkout.php");
    } else {

      // Display a message if the product is out of stock
      set_message("We only have " . $row['product_quantity'] . " " . "Available");
      redirect("../public/checkout.php");
    }
  }
}
```

Figure 67: Displaying add to cart functionality.

2. **Removing Items from the Cart:**

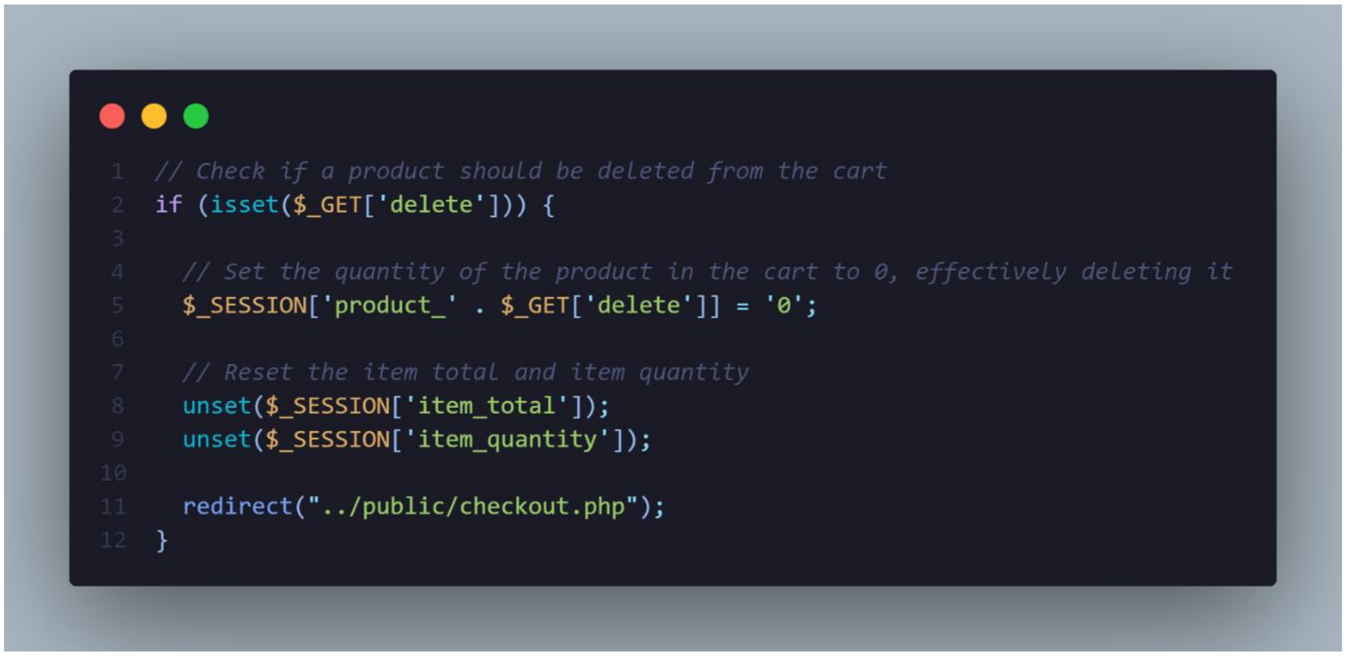
This PHP script displayed in figure 68 below is intended to handle the process of removing items from a user's shopping cart. This script runs whenever a user chooses to remove a product from their cart. The first thing it does is see if the URL's 'remove' parameter is set, indicating that a particular product—identified by its ID—is to be taken out of the cart. Next, using a session variable ($_SESSION['product_'. $_GET['remove']]), the script reduces the amount of the designated product in the cart. There are no more units of that product in the cart if the amount of that product in the cart is less than one. In response, the script sends the user to the checkout page after removing the total number of items and total price from the session data. Without deleting the session data, it only reroutes the user to the checkout page if the quantity is still greater than zero. By doing this, it is made sure that the user's intended amount and selection of products are reflected in the shopping cart.



```php
// Check if a product should be removed from the cart
if (isset($_GET['remove'])) {

  // Decrease the quantity of the product in the cart
  $_SESSION['product_' . $_GET['remove']]--;

  if ($_SESSION['product_' . $_GET['remove']] < 1) {
    // Remove the product from the cart if the quantity reaches 0
    unset($_SESSION['item_total']);
    unset($_SESSION['item_quantity']);

    redirect("../public/checkout.php");
  } else {
    redirect("../public/checkout.php");
  }
}
```

Figure 68: Displaying remove from cart functionality.

The PHP script displayed in figure 69 below is in charge of totally removing a product from a user's shopping basket. This script runs when the user chooses to remove a product from their cart, as indicated by the 'delete' parameter that is specified in the URL query string. It sets the amount of that product in the session to zero ($_SESSION['product_'. $_GET['delete']] = '0') and uses its ID (retrieved from $_GET['delete']) to identify the exact product to be deleted. The product basically gets taken out of the cart with this operation. Next, because the contents of the cart have changed, the script removes any session variables that pertain to the total cost ('item_total') and total quantity ('item_quantity') of the goods in the cart. Lastly, the user is sent to the "../public/checkout.php" checkout page. In order to guarantee that the user sees the current status of their cart upon deletion, this redirection modifies the cart display to reflect the removal of the item.



```php
// Check if a product should be deleted from the cart
if (isset($_GET['delete'])) {

  // Set the quantity of the product in the cart to 0, effectively deleting it
  $_SESSION['product_' . $_GET['delete']] = '0';

  // Reset the item total and item quantity
  unset($_SESSION['item_total']);
  unset($_SESSION['item_quantity']);

  redirect("../public/checkout.php");
}
```

Figure 69: Displaying the delete from cart functionality.

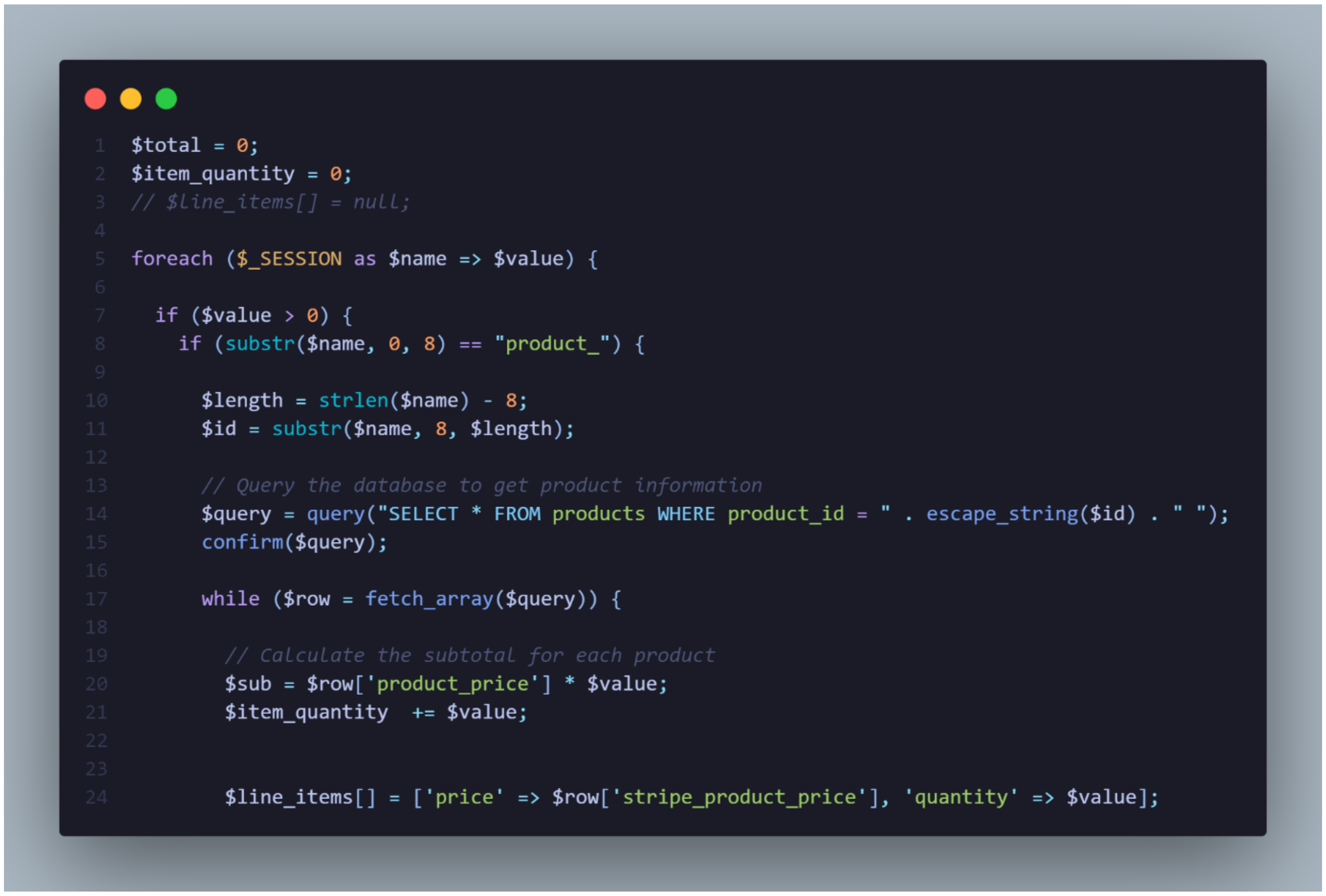
3. **Total Cost Calculations**

The purpose of the PHP script displayed in figure 70 is to manage the products in a shopping cart for an online store. In order to track the total cost and total number of products in the cart, it initializes two variables: $total and $item_quantity. The script then uses a foreach loop to traverse over each session variable. In order to determine which session variables are associated with the products in the cart, it is necessary to determine whether the variable name begins with "product_" and whether its value, which represents quantity, is more than zero.

The script extracts the product ID that is stored in the session variable name for each product that has been identified in the session. It then uses the retrieved ID to query the database for comprehensive details about that product. To avoid SQL injection issues, escape_string is used to sanitize the ID.

The script multiplies the product's price (retrieved from the database) by the quantity in the session as it processes each product within the loop to determine the subtotal for that product. The running total cost of the cart is increased by this subtotal. The total number of products in the cart is likewise added up by the script.

The script also creates an array called $line_items, which appears to be used for processing payments (using Stripe, as the array key 'stripe_product_price' indicates). The quantity of each product in the cart as well as its Stripe pricing ID are stored in this array. This feature is necessary to determine the total cost and number of goods in the basket, which are necessary for processing payments and completing the checkout process.



```php
$total = 0;
$item_quantity = 0;
// $line_items[] = null;

foreach ($_SESSION as $name => $value) {

  if ($value > 0) {
    if (substr($name, 0, 8) == "product_") {

      $length = strlen($name) - 8;
      $id = substr($name, 8, $length);

      // Query the database to get product information
      $query = query("SELECT * FROM products WHERE product_id = " . escape_string($id) . " ");
      confirm($query);

      while ($row = fetch_array($query)) {

        // Calculate the subtotal for each product
        $sub = $row['product_price'] * $value;
        $item_quantity  += $value;


        $line_items[] = ['price' => $row['stripe_product_price'], 'quantity' => $value];
```

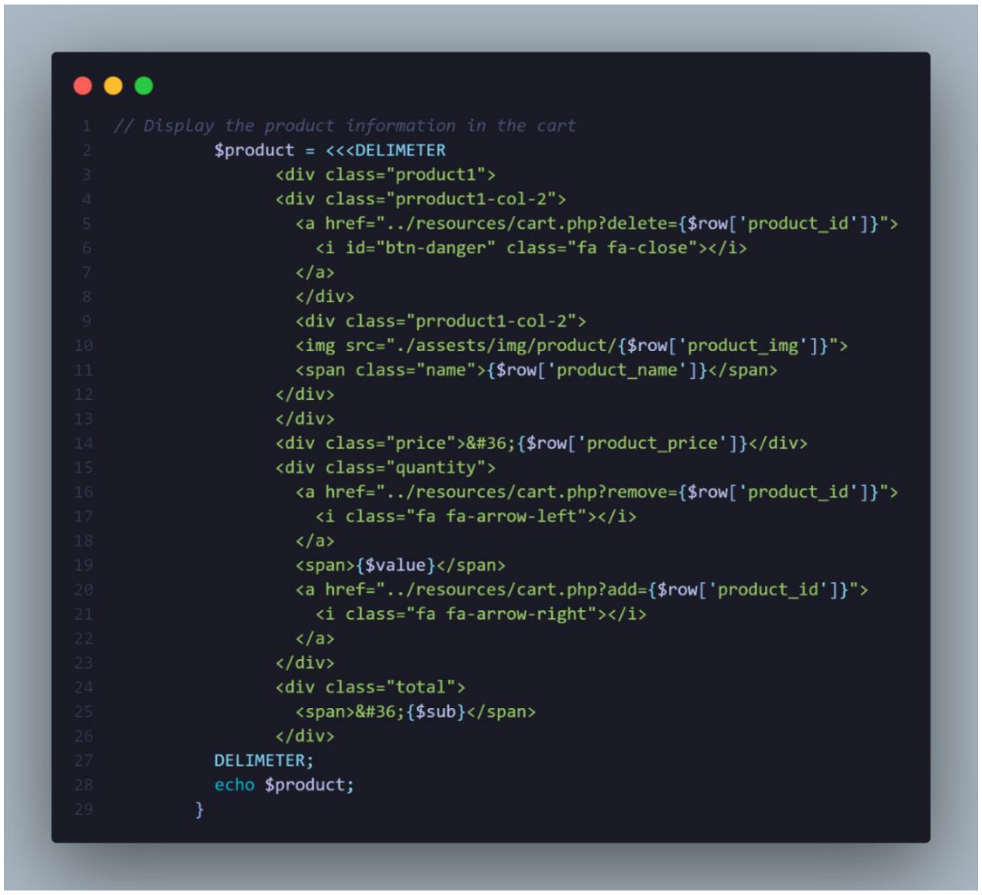
Figure 70: Displaying code to calculate totals.

4. **Item Display and Interaction:**

The purpose of the PHP code in figure 71 below is to show every item that a customer has placed to their shopping cart. For every item in the cart, it creates an HTML block using the heredoc syntax (<\<DELIMETER). A section dedicated to the product is included in this HTML, along with a picture of the item, its name, price, and the user-selected quantity.

Important interactive components are also included: a close-icon delete button that links to the cart page via a delete parameter that contains the product's ID. This enables customers to take the item out of their cart. The user can also change the product amount right from the cart thanks to the addition of increment and decrement buttons (arrow icons) that link to the cart page with add and remove parameters.

These controls are accompanied with a display of the product quantity that is currently in the basket ({$value}) and the calculated subtotal for each product ({$sub}). Before checking out, customers may manage their product selections, view prices, and adjust quantities as necessary thanks to this setup's dynamic and user-friendly cart interface.

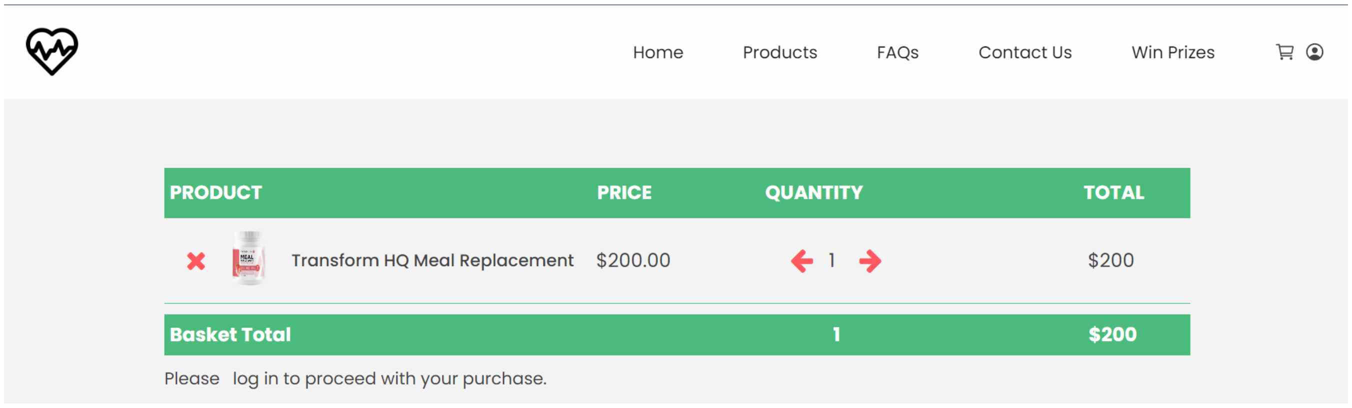


```
// Display the product information in the cart
          $product = <<<DELIMETER
                <div class="product1">
                <div class="prroduct1-col-2">
                  <a href="../resources/cart.php?delete={$row['product_id']}">
                    <i id="btn-danger" class="fa fa-close"></i>
                  </a>
                  </div>
                  <div class="prroduct1-col-2">
                  <img src="./assests/img/product/{$row['product_img']}">
                  <span class="name">{$row['product_name']}</span>
                </div>
                </div>
                <div class="price">&#36;{$row['product_price']}</div>
                <div class="quantity">
                  <a href="../resources/cart.php?remove={$row['product_id']}">
                    <i class="fa fa-arrow-left"></i>
                  </a>
                  <span>{$value}</span>
                  <a href="../resources/cart.php?add={$row['product_id']}">
                    <i class="fa fa-arrow-right"></i>
                  </a>
                </div>
                <div class="total">
                  <span>&#36;{$sub}</span>
                </div>
          DELIMETER;
          echo $product;
        }
```

Figure 71: Displaying code to display products in cart.

FRONT END | Let's take a closer look at how the user experiences the cart management functionality:

1. **Product in Cart:** The image below displays a product in the shopping cart. It displays an attractive and well-framed image of the product, as well as its name, price, quantity, and total.

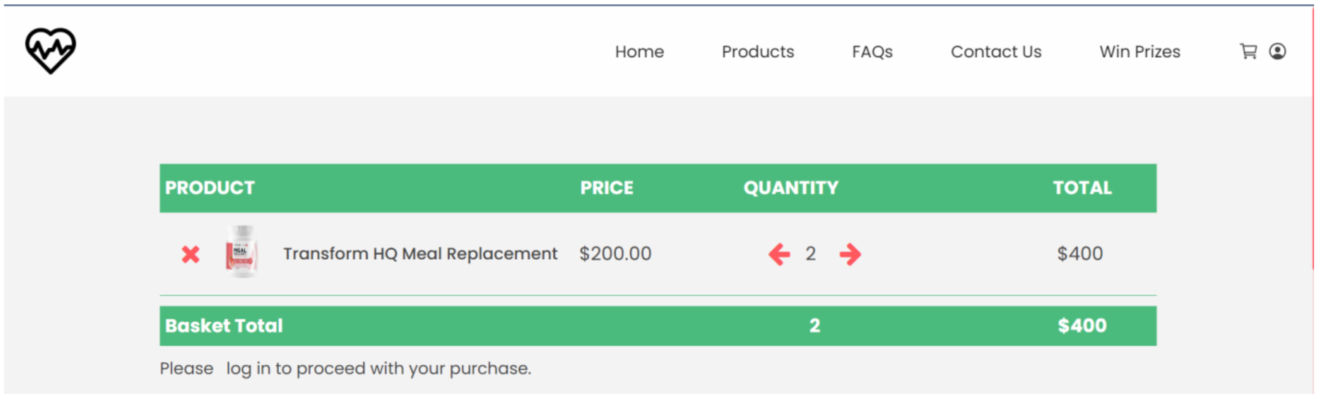


PRODUCT	PRICE	QUANTITY	TOTAL
Transform HQ Meal Replacement	$200.00	1	$200
Basket Total		1	$200

Please log in to proceed with your purchase.

Figure 72: Displaying product in cart.

2. **Quantity Increased to 2:** In this image, displayed is the same product in the cart, but now the quantity has been increased to 2.

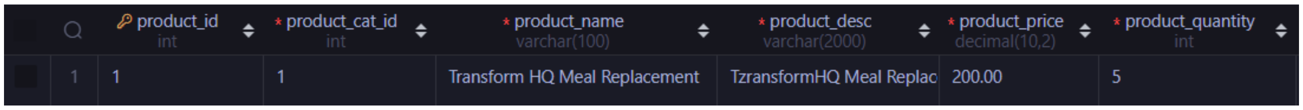


PRODUCT	PRICE	QUANTITY	TOTAL
Transform HQ Meal Replacement	$200.00	2	$400
Basket Total		2	$400

Please log in to proceed with your purchase.

Figure 73: Displaying product in cart with the quantity of two (2).

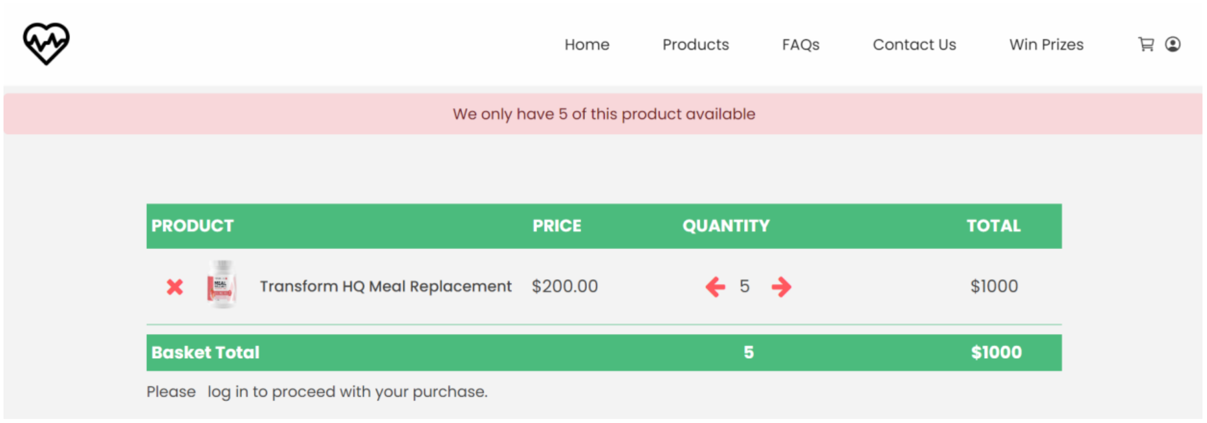
3. **Database Quantity (5):** The image below displays a representation of the product in the database, specifically indicating that there are 5 units of this product in stock.



	product_id int	product_cat_id int	product_name varchar(100)	product_desc varchar(2000)	product_price decimal(10,2)	product_quantity int
1	1	1	Transform HQ Meal Replacement	TzransformHQ Meal Replac	200.00	5

Figure 74: Displaying database quantity.

4. **Product Availability Warning:** This image below conveys to the user a message stating that there are only 5 products available.



We only have 5 of this product available

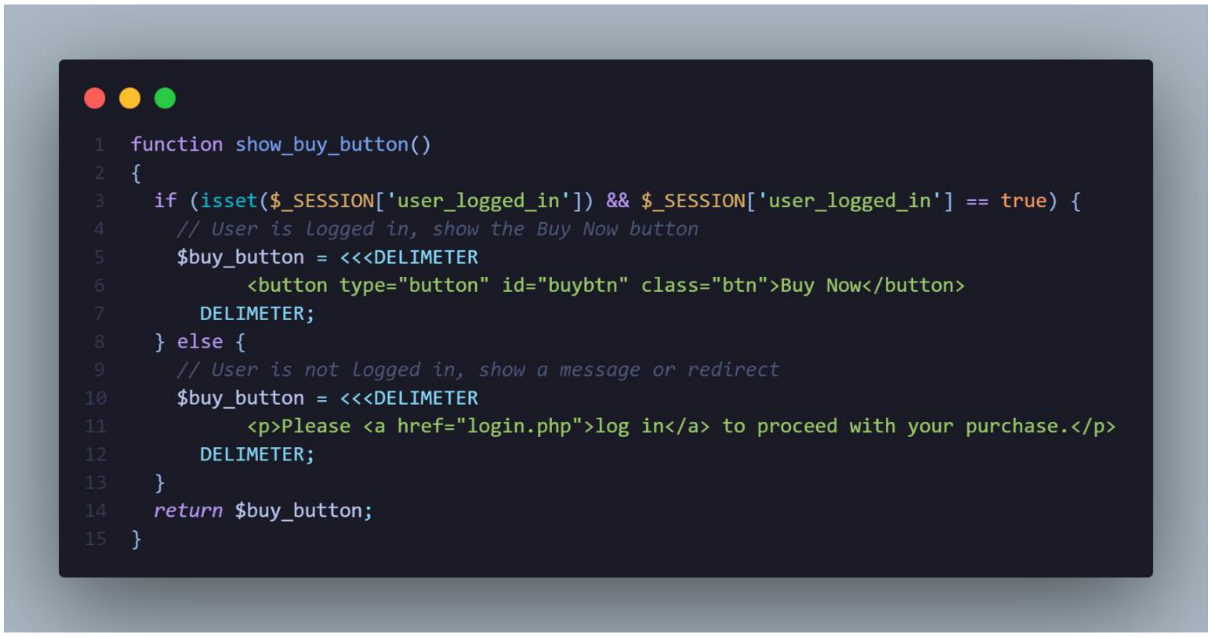
PRODUCT	PRICE	QUANTITY	TOTAL
Transform HQ Meal Replacement	$200.00	5	$1000
Basket Total		5	$1000

Please log in to proceed with your purchase.

Figure 75: Displaying 4. Product Availability Warning message.

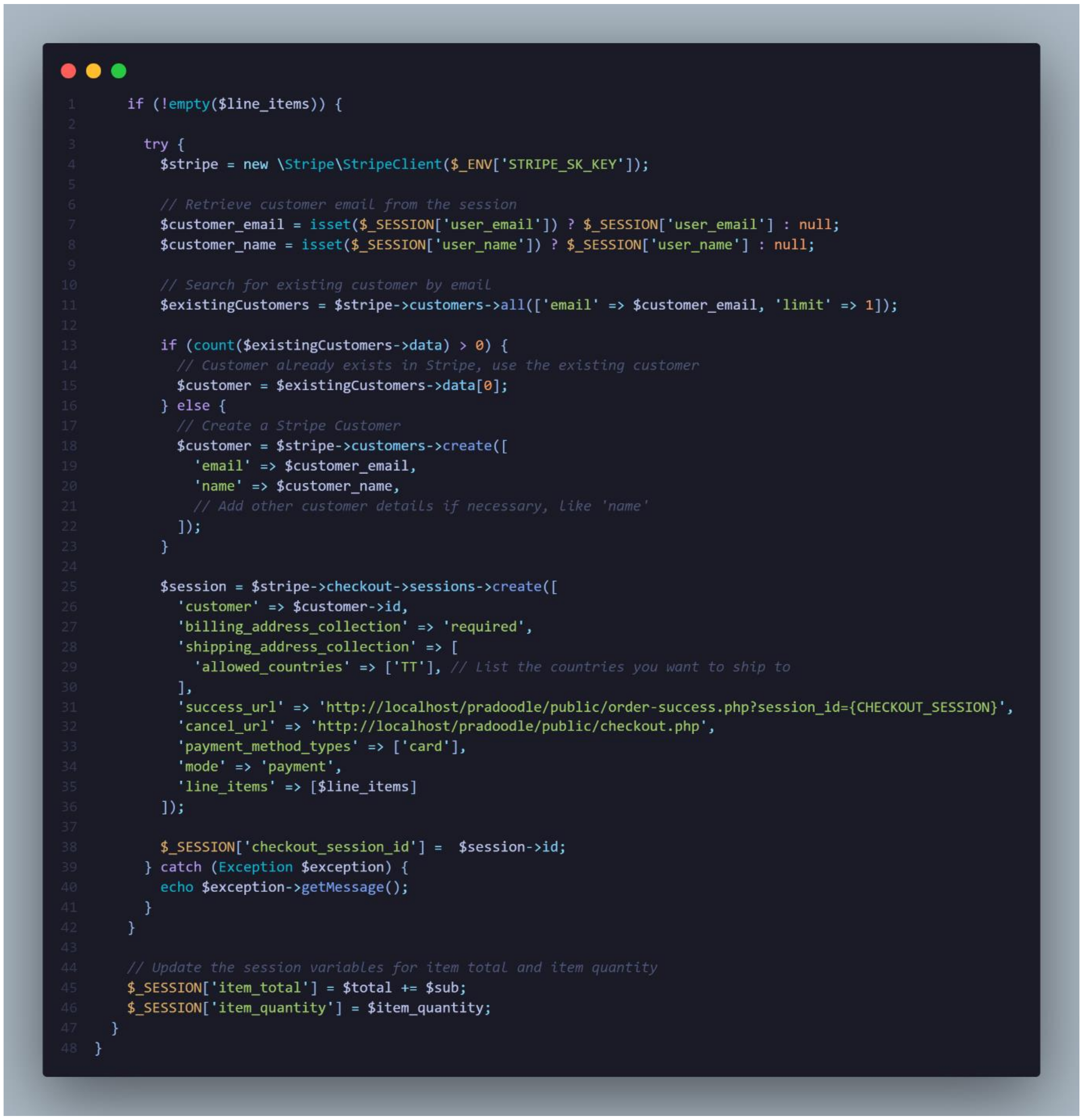
FR10 – Checkout Process

Depending on the user's login status, the show_buy_button PHP method dynamically displays alternative content. When the user is logged in, as indicated by the session variable $_SESSION['user_logged_in']; otherwise, the method uses the heredoc syntax (<<<DELIMETER) to create a "Buy Now" button. With the ID buybtn, this button is designed to be used for purchases and is built as an HTML button element. In contrast, the function creates an HTML paragraph with a message asking the user to log in and a link to the login page (login.php) if the user is not logged in. With this method, purchases can only be made by logged-in users, encouraging others to log in for the full shopping experience.



```php
function show_buy_button()
{
  if (isset($_SESSION['user_logged_in']) && $_SESSION['user_logged_in'] == true) {
    // User is logged in, show the Buy Now button
    $buy_button = <<<DELIMETER
          <button type="button" id="buybtn" class="btn">Buy Now</button>
      DELIMETER;
  } else {
    // User is not logged in, show a message or redirect
    $buy_button = <<<DELIMETER
          <p>Please <a href="login.php">log in</a> to proceed with your purchase.</p>
      DELIMETER;
  }
  return $buy_button;
}
```

Figure 76: Displaying function to display buy button.



```php
    if (!empty($line_items)) {

      try {
        $stripe = new \Stripe\StripeClient($_ENV['STRIPE_SK_KEY']);

        // Retrieve customer email from the session
        $customer_email = isset($_SESSION['user_email']) ? $_SESSION['user_email'] : null;
        $customer_name = isset($_SESSION['user_name']) ? $_SESSION['user_name'] : null;

        // Search for existing customer by email
        $existingCustomers = $stripe->customers->all(['email' => $customer_email, 'limit' => 1]);

        if (count($existingCustomers->data) > 0) {
          // Customer already exists in Stripe, use the existing customer
          $customer = $existingCustomers->data[0];
        } else {
          // Create a Stripe Customer
          $customer = $stripe->customers->create([
            'email' => $customer_email,
            'name' => $customer_name,
            // Add other customer details if necessary, like 'name'
          ]);
        }

        $session = $stripe->checkout->sessions->create([
          'customer' => $customer->id,
          'billing_address_collection' => 'required',
          'shipping_address_collection' => [
            'allowed_countries' => ['TT'], // List the countries you want to ship to
          ],
          'success_url' => 'http://localhost/pradoodle/public/order-success.php?session_id={CHECKOUT_SESSION}',
          'cancel_url' => 'http://localhost/pradoodle/public/checkout.php',
          'payment_method_types' => ['card'],
          'mode' => 'payment',
          'line_items' => [$line_items]
        ]);

        $_SESSION['checkout_session_id'] =  $session->id;
      } catch (Exception $exception) {
        echo $exception->getMessage();
      }
    }

    // Update the session variables for item total and item quantity
    $_SESSION['item_total'] = $total += $sub;
    $_SESSION['item_quantity'] = $item_quantity;
  }
}
```

Figure 77: Displaying checkout code.

This PHP code, which mostly manages the preparation for the last payment step, is an essential component of a shopping cart system's checkout procedure. The first step is to confirm if the cart contains merchandise, or line items. The script works with customer data for the checkout process if the cart is not empty.

The script verifies whether the client's name and email are kept on record for the session. It looks up an existing customer in the system using this information. It reuses the client's information if it already exists; if not, a new customer record is made. This section makes sure that every transaction, whether from a new or returning consumer, is associated with the right person.

The script then establishes a checkout session. Important data such as the customer ID, shipping and invoicing information, URLs for redirection following a successful payment or cancellation, and the payment method are all included in this session. The products in the cart, which indicate what the customer is buying, are also included in the session.

The script creates the checkout session and saves its ID in the session variable for potential future use—for example, retrieving or verifying the payment status. Any mistakes made throughout this process are detected and shown.

The program ends by updating session variables that keep track of the total amount spent and the total number of products in the cart. Maintaining the accuracy of the cart's contents while the user completes the checkout procedure depends on this update. Here, the main objective is to arrange and get ready the user's cart information for the last stages of their transaction.

FR11 – Payment Integration

Overall Explanation of the functionality

This section will go over the payment integration within the web application in great depth. Payment integration is a vital component of e-commerce websites, allowing customers to make secure purchases. Our focus will be on the payment integration area, which is powered by the Stripe payment gateway. You can find pictures of the stripe API and checkout page at Appendix G.

Note To Reader: The subject at hand focuses solely on the integration of the payment gateway into the web application. Other sections of the application documentation cover the checkout and order processing procedure, including the interaction between the cart and payment gateway.

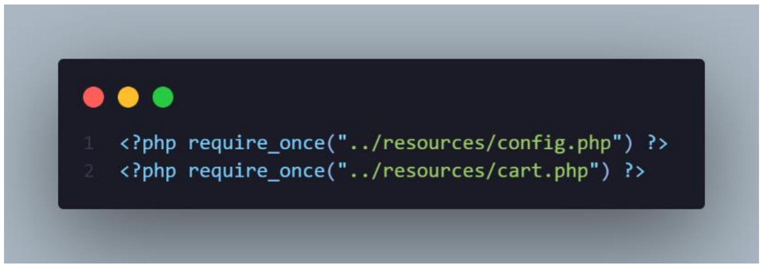
Payment Gateway Selection:

Stripe, a highly recognized and trustworthy platform for processing online payments, was chosen as the payment gateway for this web application. Stripe provides a secure and seamless transaction processing solution, preserving the security and confidentiality of users' payment information.

Let's now dissect the payment integration code into its component parts:

Including Configuration and Cart Files:

The first step in integrating payments is to include the necessary files for configuration settings and cart management. These files define functions for the shopping cart's management and create database connections.

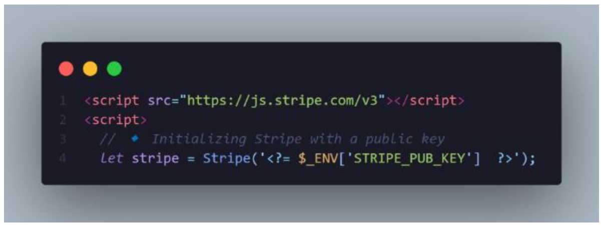


```php
<?php require_once("../resources/config.php") ?>
<?php require_once("../resources/cart.php") ?>
```

Figure 78: Displaying code for Configuration and Cart Files.

Initializing Stripe:

The public API key is used to initialize Stripe, which is essential for securely communicating with Stripe's services. By initializing the application, Stripe's infrastructure may be communicated with by the application.



```
<script src="https://js.stripe.com/v3"></script>
<script>
  //  ◆  Initializing Stripe with a public key
  let stripe = Stripe('<?= $_ENV['STRIPE_PUB_KEY']  ?>');
```

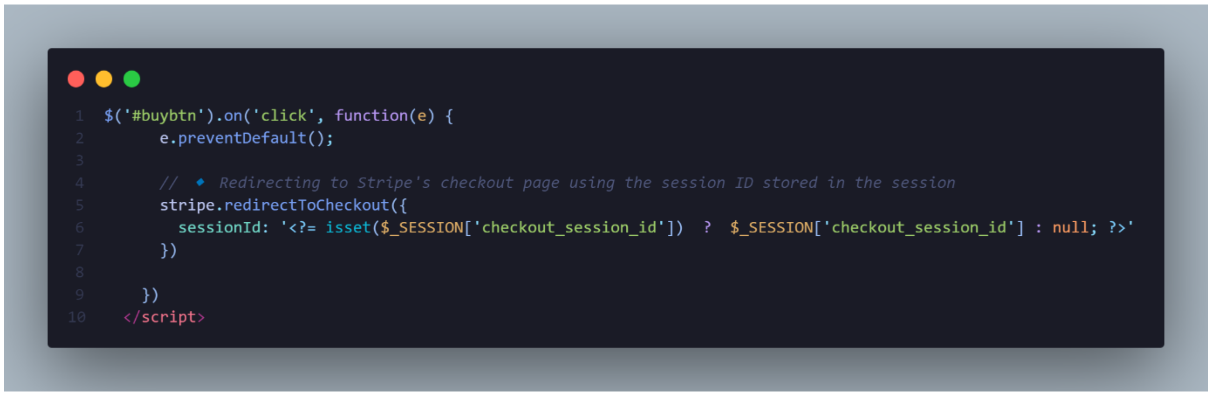
Figure 79: Displaying code to initialize stripe Api.

Event Listener for Buy Button:

The "Buy Now" button, which enables users to finalize their purchases, marks the end of the user experience. When the button is pressed, the following actions take place:

a. Stop the behavior of the form submission by default.

b. Using the session ID contained in the application's session data, reroute the user to Stripe's checkout page.



```
$('#buybtn').on('click', function(e) {
      e.preventDefault();

      //  ◆  Redirecting to Stripe's checkout page using the session ID stored in the session
      stripe.redirectToCheckout({
        sessionId: '<?= isset($_SESSION['checkout_session_id'])  ?  $_SESSION['checkout_session_id'] : null; ?>'
      })

    })
  </script>
```

Figure 80: Displaying event listener for buy now button.

Conclusion

This web application's payment integration section demonstrates a strong solution of the Stripe payment gateway. By selecting the "Buy Now" button, customers may easily add items to their shopping cart, see the total quantity and price, and start the payment process. Users can make purchases safely and conveniently online thanks to Stripe's secure API and transaction processing capabilities.

FR12 – Order Confirmation

Introduction

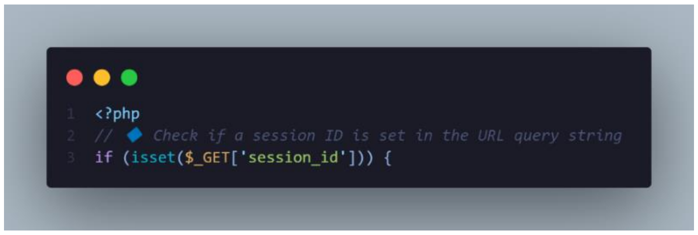
In order to give users a concise overview of their completed transactions, a web application's order confirmation part is essential. We will get into the specifics of how the order confirmation procedure is implemented in this section. This is an important stage that comes after a successful payment processing.

Order Confirmation Page - The order confirmation page acknowledges and validates the user's purchase while giving them vital transactional information. It includes facts on the items ordered, the total amount paid, and the dispatch.

Key Components of Order Confirmation:

1. Retrieving Session ID:

The system verifies that a session ID is present in the URL query string before starting the order confirmation process. An essential point of reference for obtaining transaction details is this session ID.

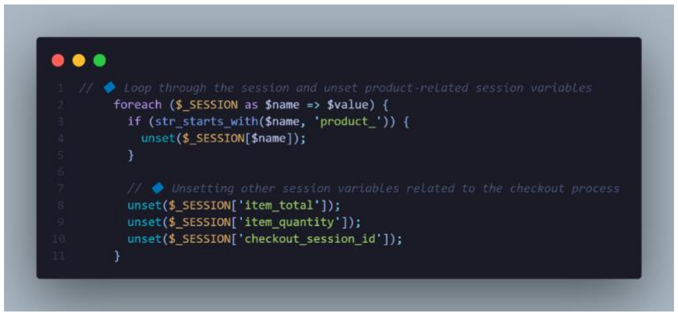


```php
<?php
// 🔷 Check if a session ID is set in the URL query string
if (isset($_GET['session_id'])) {
```

Figure 81: Displaying code to retrieve session ID.

2. Clearing Session Data:

The program then clears session variables linked to the product after verifying that the session ID is legitimate. By doing this step, you may be sure that the cart has been emptied and avoid errors in future transactions.

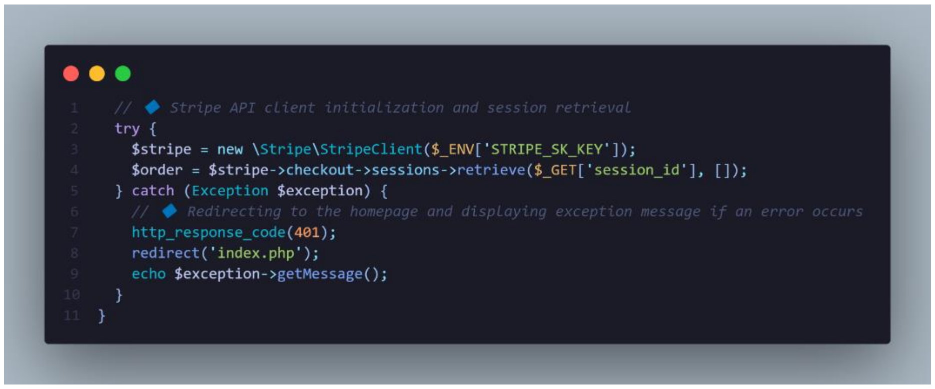


```php
// 🔷 Loop through the session and unset product-related session variables
    foreach ($_SESSION as $name => $value) {
      if (str_starts_with($name, 'product_')) {
        unset($_SESSION[$name]);
      }

      // 🔷 Unsetting other session variables related to the checkout process
      unset($_SESSION['item_total']);
      unset($_SESSION['item_quantity']);
      unset($_SESSION['checkout_session_id']);
    }
```

Figure 82: Display code to clear session data.

3. Stripe API Interaction:

During the order confirmation procedure, the Stripe API is used. The application uses the session ID to get customer information and other transaction details from Stripe.

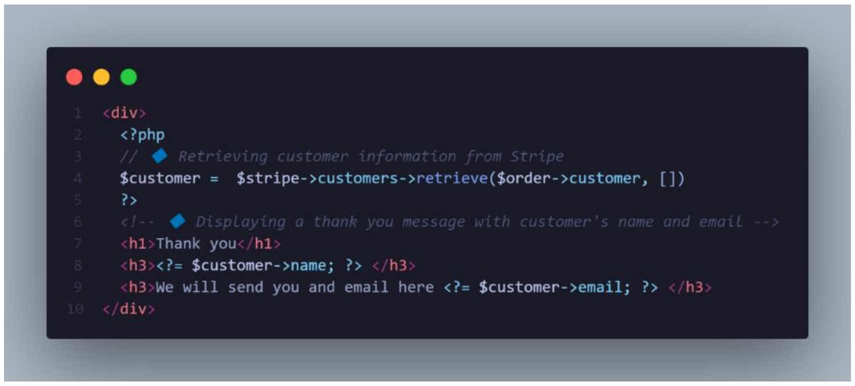


```php
// 🔷 Stripe API client initialization and session retrieval
try {
  $stripe = new \Stripe\StripeClient($_ENV['STRIPE_SK_KEY']);
  $order = $stripe->checkout->sessions->retrieve($_GET['session_id'], []);
} catch (Exception $exception) {
  // 🔷 Redirecting to the homepage and displaying exception message if an error occurs
  http_response_code(401);
  redirect('index.php');
  echo $exception->getMessage();
 }
}
```

Figure 83: Displaying code to interact with stripe API.

4. Customer Information Display:

The application first gets transaction data from Stripe, then it gets more client data and shows a customized thank-you note. The customer's name and email address are usually included in this message.



```php
<div>
  <?php
  // 🔷 Retrieving customer information from Stripe
  $customer =  $stripe->customers->retrieve($order->customer, [])
  ?>
  <!-- 🔷 Displaying a thank you message with customer's name and email -->
  <h1>Thank you</h1>
  <h3><?= $customer->name; ?> </h3>
  <h3>We will send you and email here <?= $customer->email; ?> </h3>
</div>
```

Figure 84: Displaying code for checkout success message.

Conclusion:

After a successful payment transaction, the order confirmation part of the web application makes sure that users receive a tailored and unambiguous confirmation of their purchase. The program improves the user experience and gives users peace of mind about their finished order by retrieving transaction details, deleting session data, and displaying a thank-you message. As a vital part of the entire e-commerce process, this sector makes sure that customers have a smooth and fulfilling buying experience.

FR13 – Order History

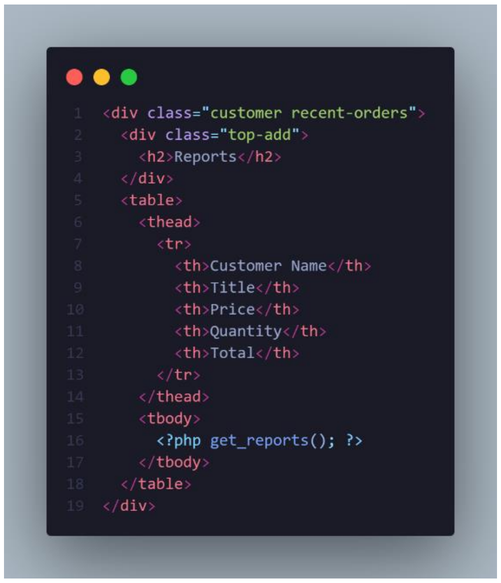
Overall Explanation of the functionality

In this section the function that enables users to see their order history will be implemented. A table on a webpage on a the "Reports" section will contain information about the customer's orders. The order information is arranged in headers such as "Customer Name," "Title," "Price," "Quantity," and "Total". The details of each order are displayed in this table, which is dynamically populated with data from the database's'reports' table via the PHP function get_reports(). The script retrieves customer information for orders associated with Stripe customers in order to match it with the session of the logged-in user. With the help of this tool, customers may see an organized, detailed history of their orders.

BACK END | Let's take a closer look at the order history functionality (How the functionality was implemented):

1. HTML Table Structure:

This HTML code displayed in figure 85 below adds a table of the customer's orders. The header for this section is "Reports." The table is arranged with headers that list the essential information about each order, including "Customer Name," "Title," "Price," "Quantity," and "Total." A PHP function named get_reports() is used inside the table body (<tbody>) to dynamically retrieve and show the customer's latest order details from the database. With this configuration, users may see all of their previous orders' product names, prices, quantities, and total amounts in an easy-to-understand format.

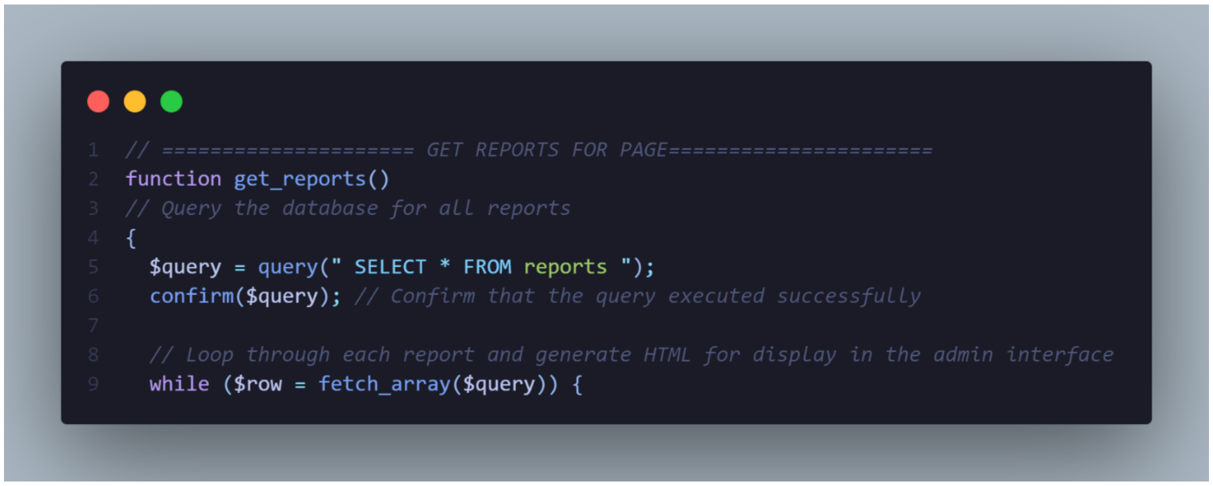


```html
<div class="customer recent-orders">
  <div class="top-add">
    <h2>Reports</h2>
  </div>
  <table>
    <thead>
      <tr>
        <th>Customer Name</th>
        <th>Title</th>
        <th>Price</th>
        <th>Quantity</th>
        <th>Total</th>
      </tr>
    </thead>
    <tbody>
      <?php get_reports(); ?>
    </tbody>
  </table>
</div>
```

Figure 85: Displaying HTML code for table structure.

2. PHP Function get_reports() | Database Query | Iterative Processing:

The purpose of the PHP function get_reports is to retrieve and provide client order reports, enhancing the HTML table structure that we previously covered. The query is executed successfully and returns all entries from the'reports' table by querying the database. The function then goes through each report iteratively, extracting pertinent information such as amount, total, and pricing of the commodity.

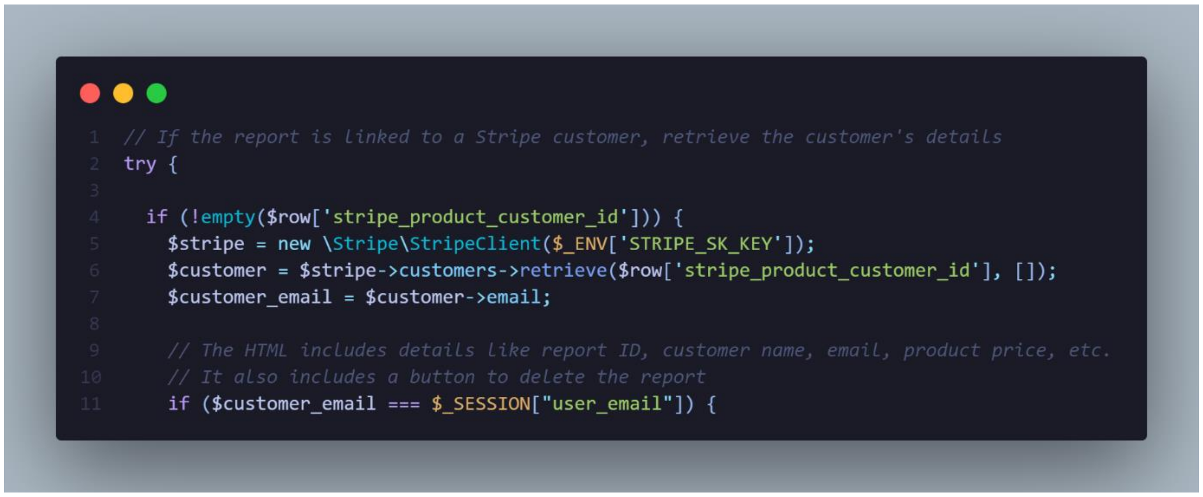


```php
// ===================== GET REPORTS FOR PAGE======================
function get_reports()
// Query the database for all reports
{
  $query = query(" SELECT * FROM reports ");
  confirm($query); // Confirm that the query executed successfully

  // Loop through each report and generate HTML for display in the admin interface
  while ($row = fetch_array($query)) {
```

Figure 86: Displaying code for get_reports() function.

3. Stripe Integration:

Based on the Stripe customer ID, reports associated with Stripe customers are retrieved via the Stripe API. The function verifies whether the email address of the Stripe client and the user who is currently logged in match (stored in the session).

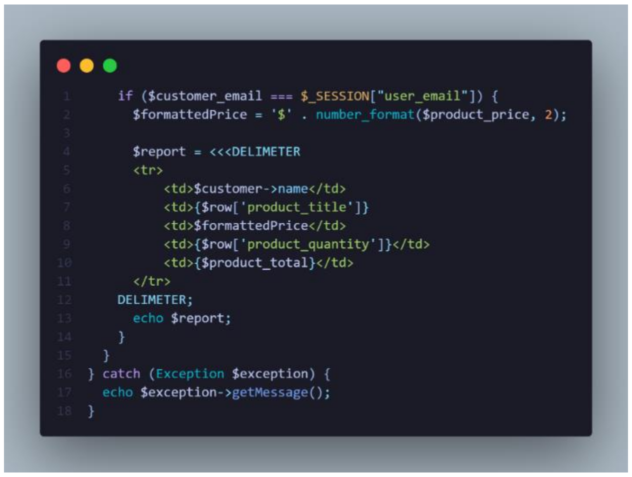


```php
// If the report is linked to a Stripe customer, retrieve the customer's details
try {

  if (!empty($row['stripe_product_customer_id'])) {
    $stripe = new \Stripe\StripeClient($_ENV['STRIPE_SK_KEY']);
    $customer = $stripe->customers->retrieve($row['stripe_product_customer_id'], []);
    $customer_email = $customer->email;

    // The HTML includes details like report ID, customer name, email, product price, etc.
    // It also includes a button to delete the report
    if ($customer_email === $_SESSION["user_email"]) {
```

Figure 87: Display code for STRIPE integration.

4. Dynamic HTML Rows Creation:

In the event that a match is found, each matching report generates an HTML row (<tr>) with the customer's name, product title, price, quantity, and total shown. The user can see a customized overview of their most recent orders thanks to this dynamic report detail creation, which guarantees the information in the table is relevant to their transactions.

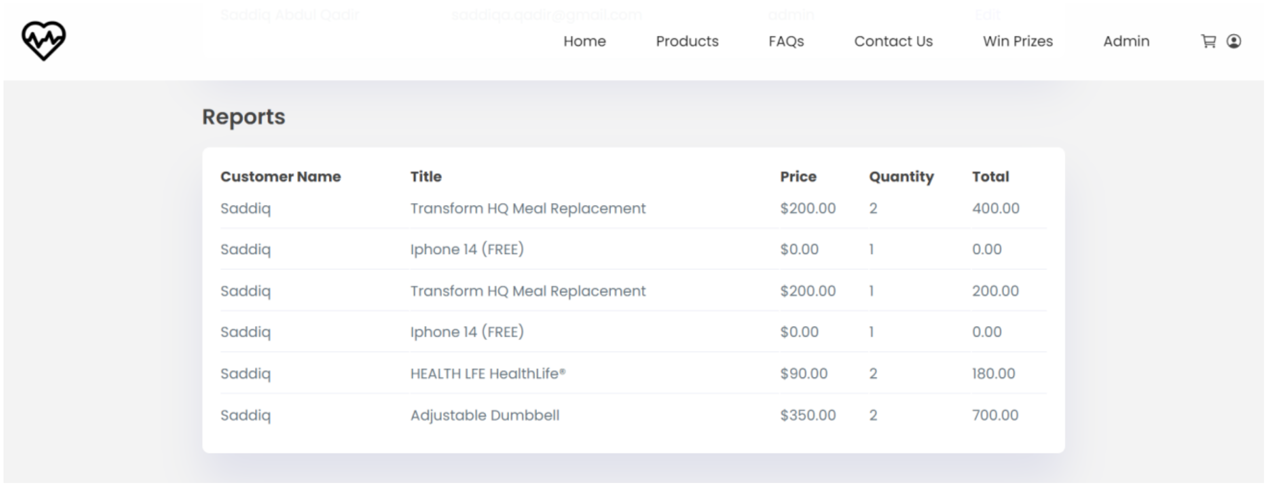


```php
    if ($customer_email === $_SESSION["user_email"]) {
      $formattedPrice = '$' . number_format($product_price, 2);

      $report = <<<DELIMETER
      <tr>
          <td>$customer->name</td>
          <td>{$row['product_title']}
          <td>$formattedPrice</td>
          <td>{$row['product_quantity']}</td>
          <td>{$product_total}</td>
      </tr>
    DELIMETER;
      echo $report;
    }
  }
} catch (Exception $exception) {
  echo $exception->getMessage();
}
```

Figure 88: Displaying code to dynamically show HTML rows.

FRONT END | Let's take a closer look at how the user experiences the cart management functionality:

The image below demonstrates how users can easily see their order history, which displays the products they've purchased as well as their pricing, quantities, and final totals.



Reports

Customer Name	Title	Price	Quantity	Total
Saddiq	Transform HQ Meal Replacement	$200.00	2	400.00
Saddiq	Iphone 14 (FREE)	$0.00	1	0.00
Saddiq	Transform HQ Meal Replacement	$200.00	1	200.00
Saddiq	Iphone 14 (FREE)	$0.00	1	0.00
Saddiq	HEALTH LFE HealthLife®	$90.00	2	180.00
Saddiq	Adjustable Dumbbell	$350.00	2	700.00

Figure 89: Display image with customer order history.

FR14 – User Reviews and Ratings

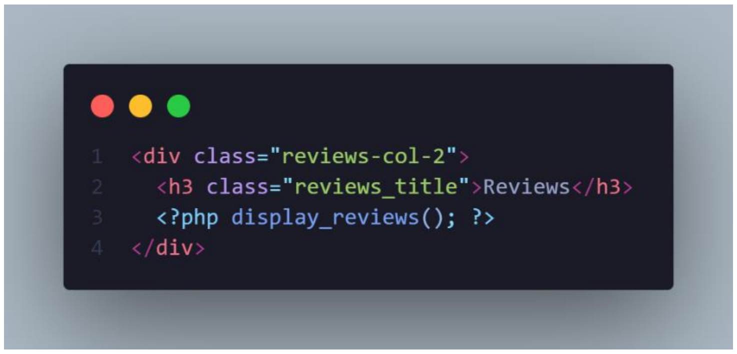
Overall Explanation of the functionality

This section will cover the user review and rating functionality. This section manages the review function on a website's product page. It uses display_reviews() to show reviews that already exist, presents a form for users to post new reviews, and handles these submissions. The add_review() function handles information like as the reviewer's name, email address, and review content when a user publishes a review. It also updates the product's comment count. By enabling users to write and read product reviews, the system promotes a community-driven purchasing experience and increases user engagement.

BACK END | Let's take a closer look at the user reviews and rating functionality (How the functionality was implemented):

1. **Displaying Reviews:**

This code displayed below is figure 90 adds a section to a webpage where user reviews can be shown. Customers can get insights and feedback from other users by using the PHP method display_reviews() to fetch and display review content inside a specific region, all under the "Reviews" title.

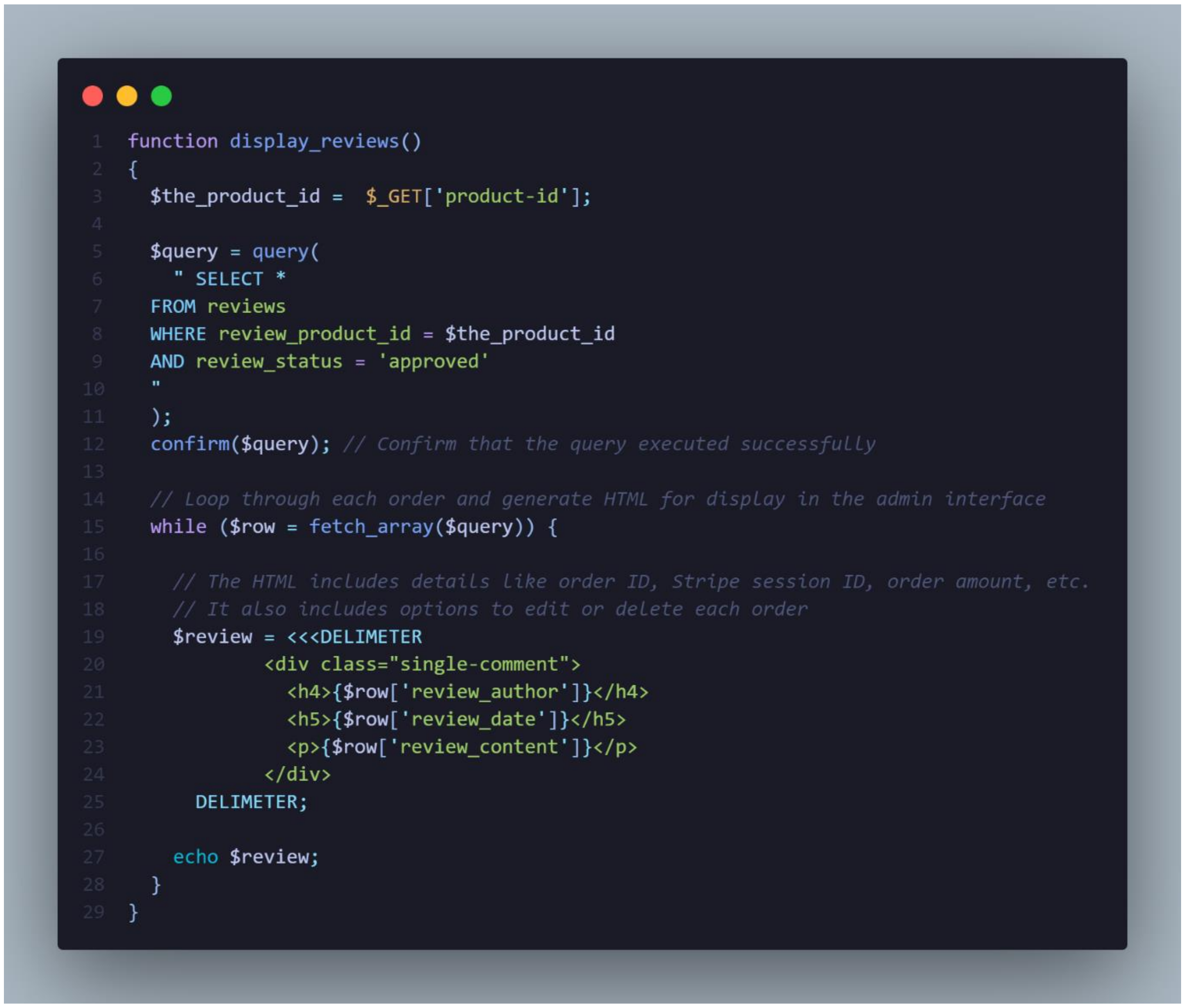


```
<div class="reviews-col-2">
  <h3 class="reviews_title">Reviews</h3>
  <?php display_reviews(); ?>
</div>
```

Figure 90: Displaying code that creates the section to display reviews.

2. **Function - display_reviews():**

The purpose of the PHP function display_reviews is to retrieve and show product reviews on a webpage. After extracting the product ID from the query string of the URL, it runs a database query to choose reviews related to this product ID, limiting the selection to approved reviews. Following verification of the query's successful execution, the function iterates through each retrieved review. It creates a block of HTML for every review, organized to show the name of the reviewer, the review date, and the review content itself, all contained in a div element with a single-comment style. With this configuration, user evaluations should be presented in an easily legible and structured manner, providing other consumers with insightful information and input regarding the product.

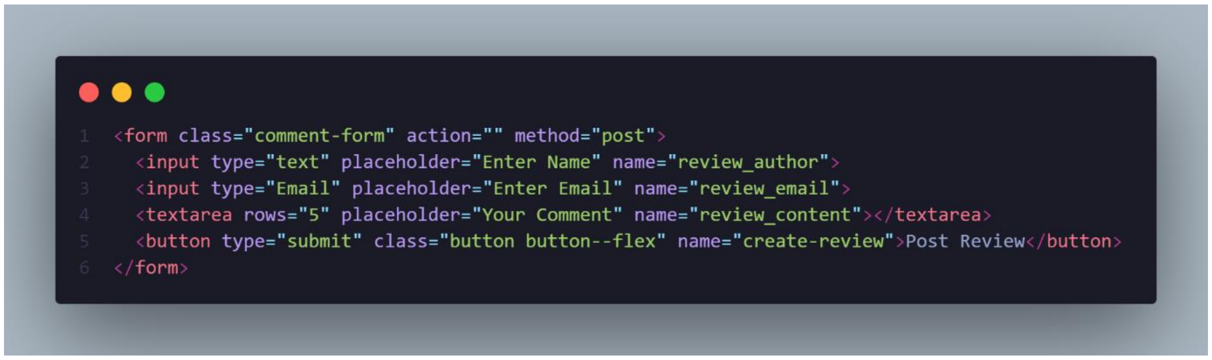


```
function display_reviews()
{
  $the_product_id =  $_GET['product-id'];

  $query = query(
    " SELECT *
  FROM reviews
  WHERE review_product_id = $the_product_id
  AND review_status = 'approved'
  "
  );
  confirm($query); // Confirm that the query executed successfully

  // Loop through each order and generate HTML for display in the admin interface
  while ($row = fetch_array($query)) {

    // The HTML includes details like order ID, Stripe session ID, order amount, etc.
    // It also includes options to edit or delete each order
    $review = <<<DELIMETER
            <div class="single-comment">
              <h4>{$row['review_author']}</h4>
              <h5>{$row['review_date']}</h5>
              <p>{$row['review_content']}</p>
            </div>
      DELIMETER;

    echo $review;
  }
}
```

Figure 91: Displaying function to display reviews.

3. Review Submission Form

The HTML code displayed in the figure 92 below is responsible for the development of the form that allows customers to post product reviews. It has a wider text space for the review content and input fields for the reviewer's name and email. The POST method is used to submit the form. After entering their information and leaving a remark, users can submit their reviews by clicking the "Post Review" submit button. The purpose of this form is to facilitate customers' effortless provision of comments or thoughts regarding products, hence augmenting the interactive and community elements of the website.

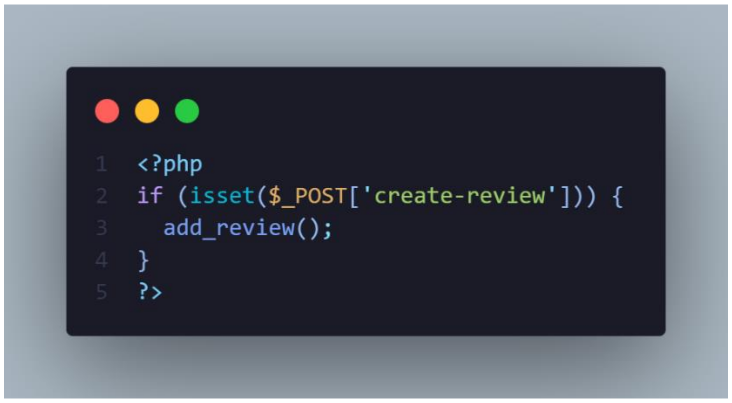


```
<form class="comment-form" action="" method="post">
  <input type="text" placeholder="Enter Name" name="review_author">
  <input type="Email" placeholder="Enter Email" name="review_email">
  <textarea rows="5" placeholder="Your Comment" name="review_content"></textarea>
  <button type="submit" class="button button--flex" name="create-review">Post Review</button>
</form>
```

Figure 92: Displaying review submission form.

4. Handling Review Submissions:

This PHP script verifies whether a website's review form has been submitted. It recognizes this action through the 'create-review' POST variable when the user clicks the "Post Review" button. The script invokes the add_review() function in the event that the form is submitted. This function is in charge of handling and keeping track of the reviewer's name, email address, and comments that are entered by the user.



```
<?php
if (isset($_POST['create-review'])) {
  add_review();
}
?>
```

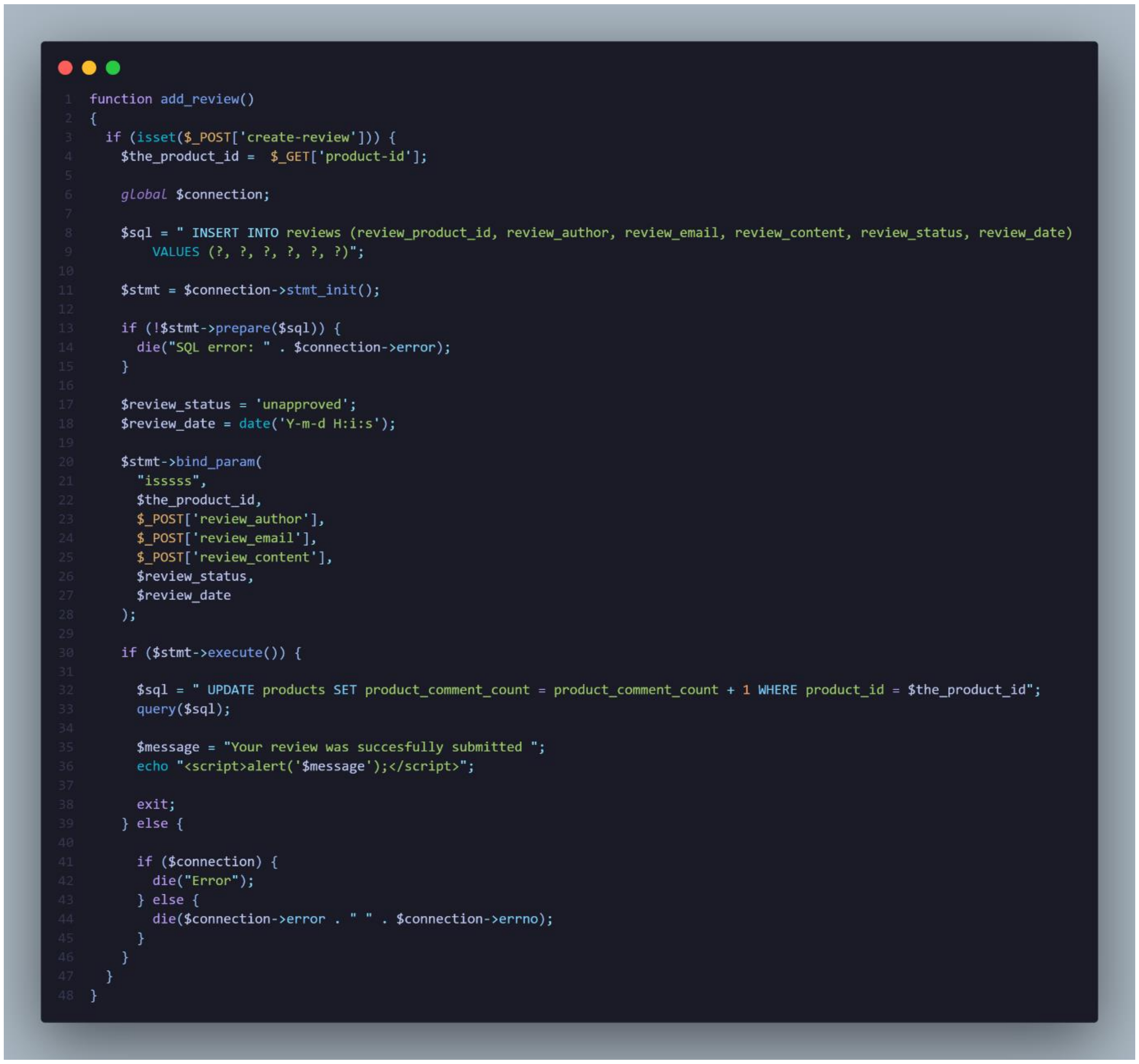
Figure 93: Displaying code that verifies if the form was submitted.

5. Function - add_review():

This PHP script's add_review() method is meant to handle and keep track of reviews that users write. Upon receiving a review from a user, the function imports the review into the'reviews' database table by preparing a SQL statement and retrieving the product ID from the URL. In addition to the product ID, author, email, and text, the review also contains the date, time, and default status of "unapproved."

To prevent SQL injection, the function makes advantage of prepared statements for security. Following the binding and execution of the user-provided data to the statement, the script increases the product's comment count in the 'products' table. JavaScript's alert function is used to deliver a confirmation message to the user upon successful execution.

The script manages errors during this process by displaying error notifications. This feature makes sure that user reviews are safely entered into the database and update the product's comment count to reflect the new review, subject to approval.



```php
function add_review()
{
  if (isset($_POST['create-review'])) {
    $the_product_id =  $_GET['product-id'];

    global $connection;

    $sql = " INSERT INTO reviews (review_product_id, review_author, review_email, review_content, review_status, review_date)
        VALUES (?, ?, ?, ?, ?, ?)";

    $stmt = $connection->stmt_init();

    if (!$stmt->prepare($sql)) {
      die("SQL error: " . $connection->error);
    }

    $review_status = 'unapproved';
    $review_date = date('Y-m-d H:i:s');

    $stmt->bind_param(
      "isssss",
      $the_product_id,
      $_POST['review_author'],
      $_POST['review_email'],
      $_POST['review_content'],
      $review_status,
      $review_date
    );

    if ($stmt->execute()) {

      $sql = " UPDATE products SET product_comment_count = product_comment_count + 1 WHERE product_id = $the_product_id";
      query($sql);

      $message = "Your review was succesfully submitted ";
      echo "<script>alert('$message');</script>";

      exit;
    } else {

      if ($connection) {
        die("Error");
      } else {
        die($connection->error . " " . $connection->errno);
      }
    }
  }
}
```

Figure 94: Displaying add review section.

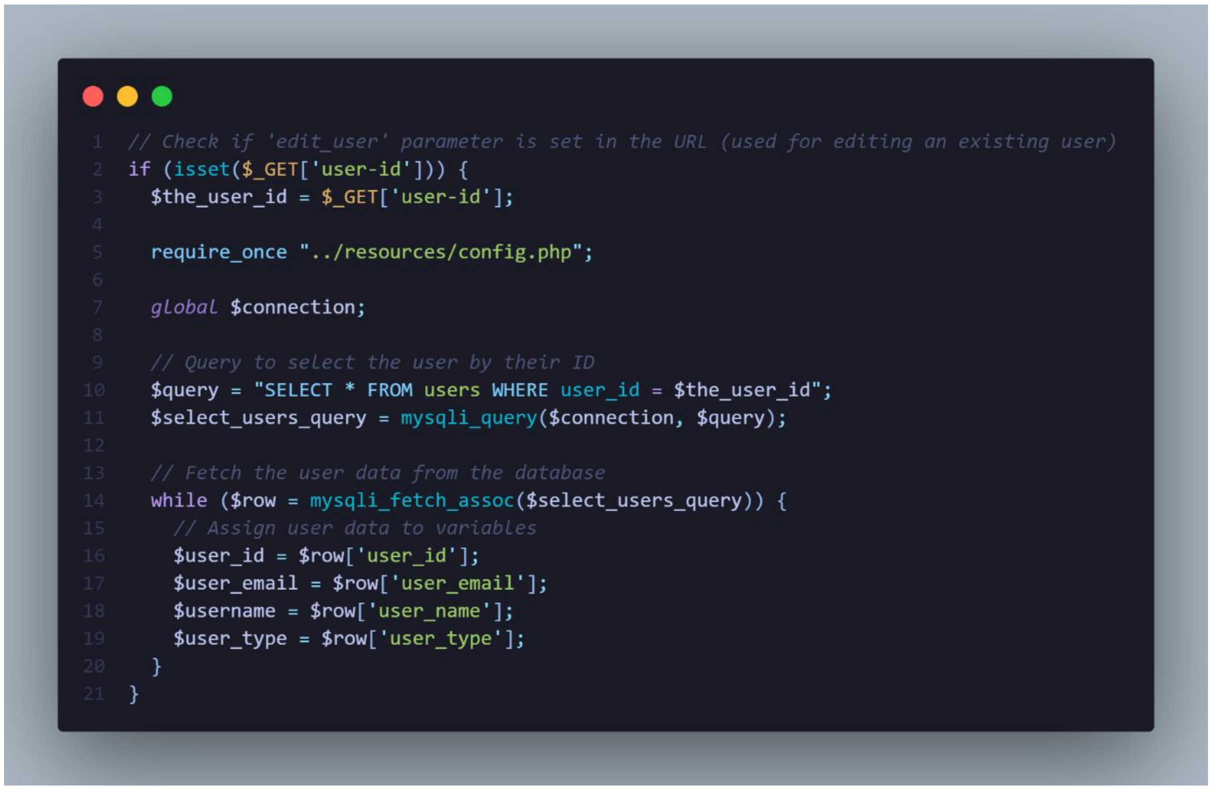
FR15 – Account Management

The application's account management feature gives users control over their account details in an effort to empower them. Users can view and modify their personal information in this section. It improves the user experience by offering an easy-to-use interface for managing accounts. We will look at the main features and specifics of this section's implementation below.

BACK END | Let's take a closer look at Account Management feature (How the functionality was implemented):

Data Retrieval:

The system first verifies that the 'edit_user' option is included in the URL before granting users the ability to manage their accounts. This option indicates that the user wants to change their account information if it is present. The unique ID of the user is taken from the URL to make it easier to identify the particular user. The user's data is then retrieved by establishing a connection to the database. The user's ID, email address, username, and user type are among the vital details in this data that are essential for efficient account management.

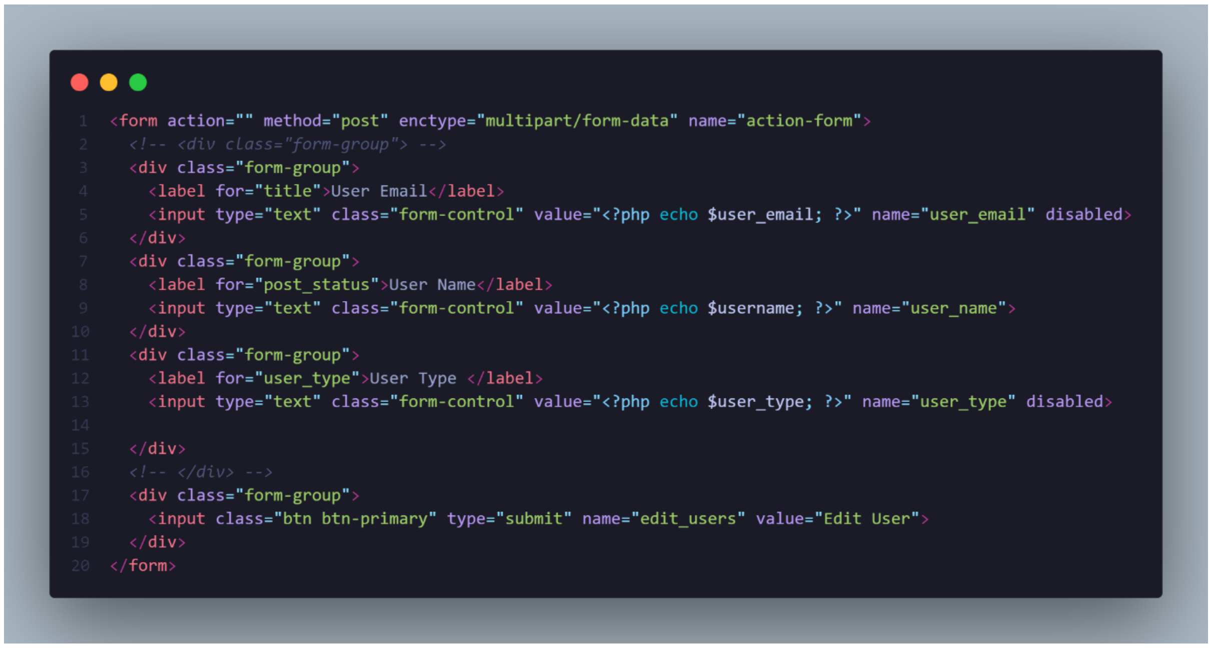


```php
// Check if 'edit_user' parameter is set in the URL (used for editing an existing user)
if (isset($_GET['user-id'])) {
  $the_user_id = $_GET['user-id'];

  require_once "../resources/config.php";

  global $connection;

  // Query to select the user by their ID
  $query = "SELECT * FROM users WHERE user_id = $the_user_id";
  $select_users_query = mysqli_query($connection, $query);

  // Fetch the user data from the database
  while ($row = mysqli_fetch_assoc($select_users_query)) {
    // Assign user data to variables
    $user_id = $row['user_id'];
    $user_email = $row['user_email'];
    $username = $row['user_name'];
    $user_type = $row['user_type'];
  }
}
```

Figure 95: Displaying code to retrieve the data.

Edit User Form:

Users are provided with an HTML form that is easy to use, enabling them to make changes to their account details. 'User_name' and 'User_email' are input fields on this form. The 'user_email' field is marked as disabled, meaning that editing it is not possible on this specific page. Similarly, the 'user_type' field is visible but not editable because it is disabled. The form fields are pre-populated with the user's current data to expedite the user experience and provide a point of reference for any changes the user may want to make.

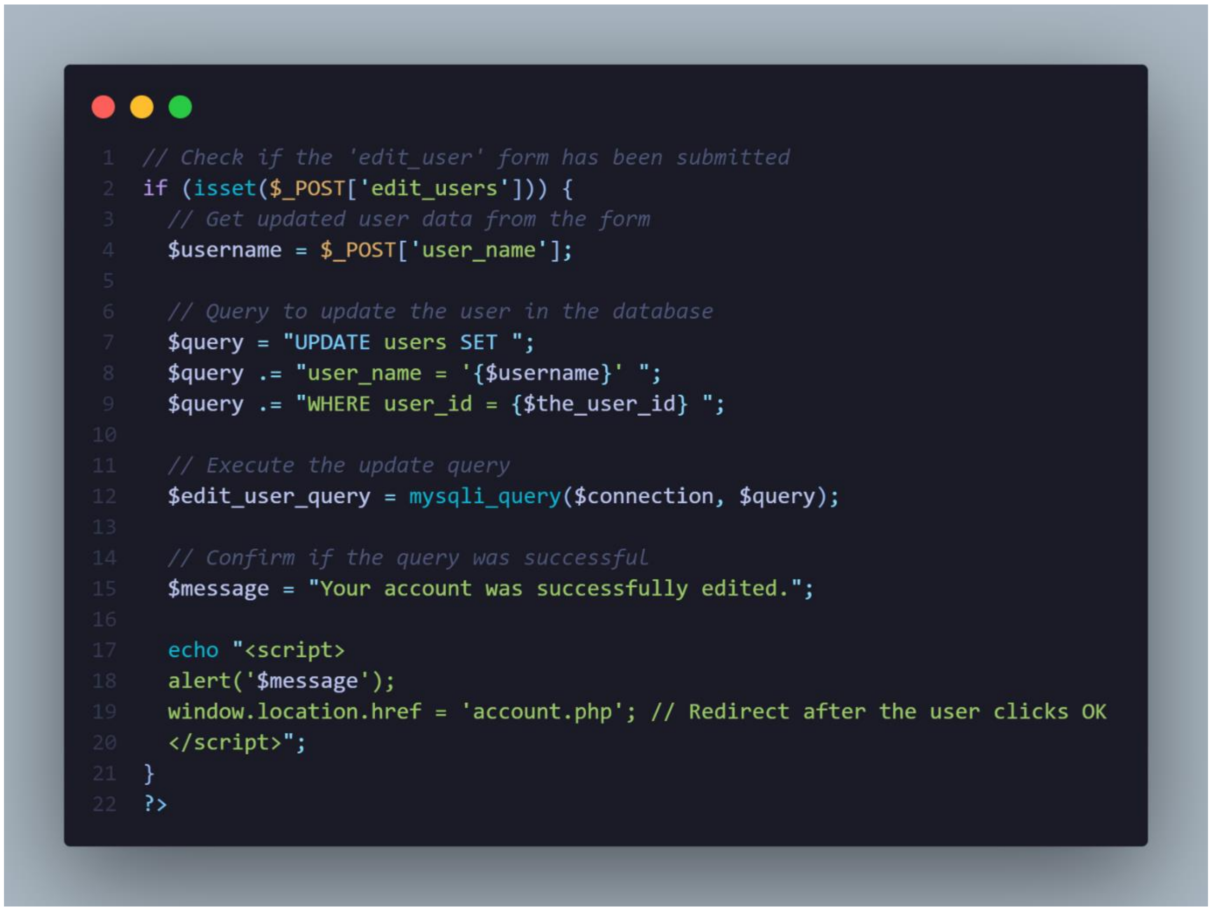


```html
<form action="" method="post" enctype="multipart/form-data" name="action-form">
  <!-- <div class="form-group"> -->
  <div class="form-group">
    <label for="title">User Email</label>
    <input type="text" class="form-control" value="<?php echo $user_email; ?>" name="user_email" disabled>
  </div>
  <div class="form-group">
    <label for="post_status">User Name</label>
    <input type="text" class="form-control" value="<?php echo $username; ?>" name="user_name">
  </div>
  <div class="form-group">
    <label for="user_type">User Type </label>
    <input type="text" class="form-control" value="<?php echo $user_type; ?>" name="user_type" disabled>

  </div>
  <!-- </div> -->
  <div class="form-group">
    <input class="btn btn-primary" type="submit" name="edit_users" value="Edit User">
  </div>
</form>
```

Figure 96: Displaying the edit user form.

Editing User Data:

A POST request is sent to the very same page when the "Edit User" button on the form is clicked to submit it. The PHP script that goes with it evaluates if the 'edit_users' POST option is set, indicating that the form has been submitted. Should the submission of the form be identified, the updated 'user_name' value is obtained from the form. Then, in order to update the user's data in the database, a SQL query is dynamically built with a focus on the 'user_name' field. If the update process is successful, the user is automatically sent to their account management page and a success message is displayed.



```php
// Check if the 'edit_user' form has been submitted
if (isset($_POST['edit_users'])) {
  // Get updated user data from the form
  $username = $_POST['user_name'];

  // Query to update the user in the database
  $query = "UPDATE users SET ";
  $query .= "user_name = '{$username}' ";
  $query .= "WHERE user_id = {$the_user_id} ";

  // Execute the update query
  $edit_user_query = mysqli_query($connection, $query);

  // Confirm if the query was successful
  $message = "Your account was successfully edited.";

  echo "<script>
  alert('$message');
  window.location.href = 'account.php'; // Redirect after the user clicks OK
  </script>";
}
?>
```

Figure 97: Displaying the code to edit a user.

Conclusion:

An essential part of the application that gives users full control is the account administration section. It greatly enhances customer pleasure and convenience by allowing users to view and change personal information. This feature, which provides a simple and effective interface for account administration, is a prime example of user-centric design.

FR16 – Admin Dashboard

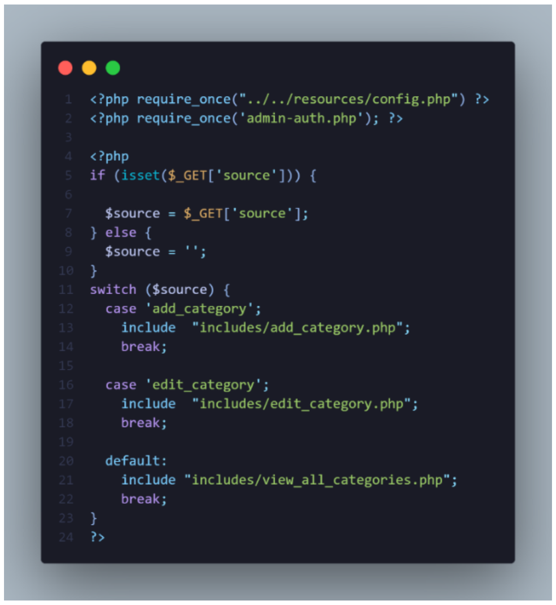
We'll look at the admin dashboard in this section. This is a central location where administrators may manage every aspect of the website's operations and content. This covers handling user accounts, adding, modifying, and removing products, as well as controlling user comments by accepting or rejecting them. Admins can view and control the spinning game winners from the dashboard as well. It also incorporates with the tools for advanced data analytics and visualization, making it possible to obtain a comprehensive summary of website metrics. Administrators get access to all the tools they need for efficient site management, including the ability to manage product categories, see and process orders, and access comprehensive reports.

BACK END | Let's take a closer look at Admin Dashboard (How the functionality was implemented):

Implementing CRUD operations for Users, Reports, Orders, Products, Spinner Winners, and Categories:

This section will examine the implementation of the CRUD (Create, Read, Update, Delete) operations for categories. This is how it works: all of the categories are displayed by default when an admin selects the select categories on the navbar. The admin then gets the option to edit an individual category record or add a new category when on the "View All Categories" page.

The PHP script below illustrates how the category, different pages are accessed. The script starts with two important PHP files, 'config.php' for configuration settings and 'admin-auth.php' for authentication logic. It then uses isset($_GET['source']) to check for the availability of a'source' parameter in the URL's query string, assigning its value to the $source variable if one exists; otherwise, $source is set to an empty string. The code then enters a switch statement and includes and executes particular PHP files based on the value of $source. If $source is 'add_category,' for example, it includes 'add_category.php,' and if it is 'edit_category,' it includes 'edit_category.php.' If no match is discovered (which is the default scenario), the file 'view_all_categories.php' is included. This method enables control over alternative admin area views or actions based on the URL's'source' argument.

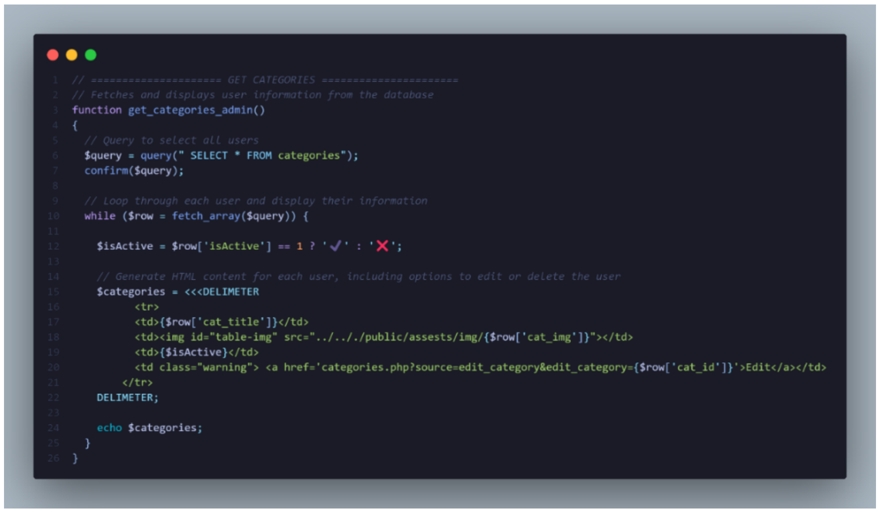


```php
<?php require_once("../../resources/config.php") ?>
<?php require_once('admin-auth.php'); ?>

<?php
if (isset($_GET['source'])) {

  $source = $_GET['source'];
} else {
  $source = '';
}
switch ($source) {
  case 'add_category';
    include  "includes/add_category.php";
    break;

  case 'edit_category';
    include  "includes/edit_category.php";
    break;

  default:
    include "includes/view_all_categories.php";
    break;
}
?>
```

Figure 98: Displaying the code to control access to pages.

View Categories (Default Options)

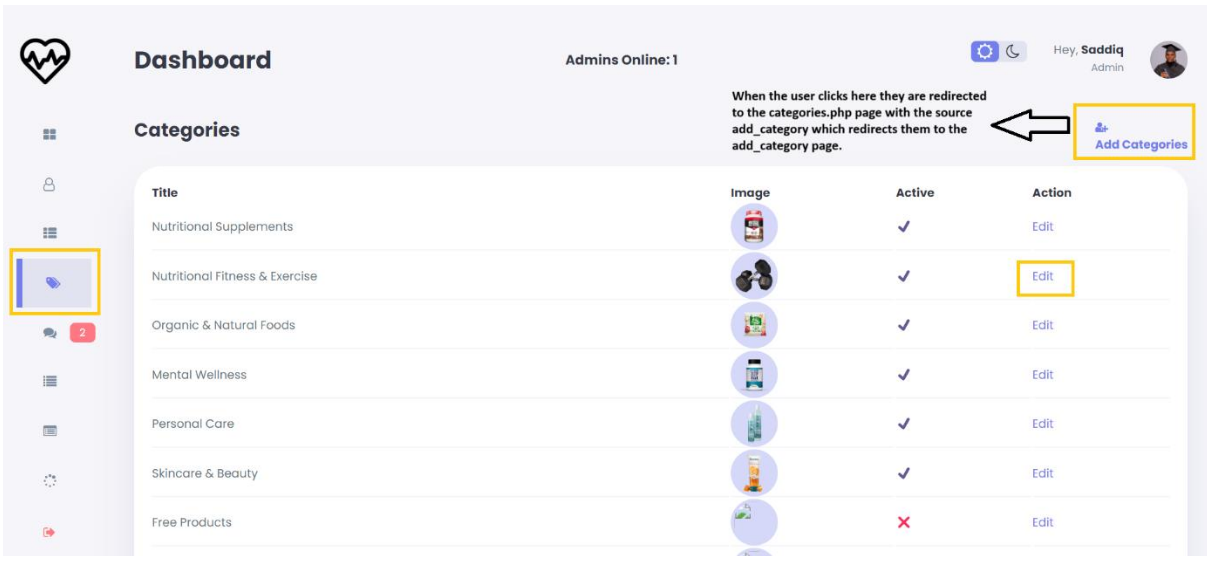
The script below displays a PHP function that is used to display the categories on the view all categories page. The PHP method get_categories_admin() returns category information from the database table "categories." It generates HTML text for each category that includes the title, an accompanying image, and an indicator of the category's active status (represented by checkmark or cross symbols). The feature also adds a "Edit" link to each category, allowing administrators to change the details of each category. The created HTML content is subsequently repeated to the webpage, allowing administrators to easily examine and change category data.



```php
// ===================== GET CATEGORIES ======================
// Fetches and displays user information from the database
function get_categories_admin()
{
  // Query to select all users
  $query = query(" SELECT * FROM categories");
  confirm($query);

  // Loop through each user and display their information
  while ($row = fetch_array($query)) {

    $isActive = $row['isActive'] == 1 ? '✓' : '❌';

    // Generate HTML content for each user, including options to edit or delete the user
    $categories = <<<DELIMETER
          <tr>
          <td>{$row['cat_title']}</td>
          <td><img id="table-img" src="../../../public/assests/img/{$row['cat_img']}"></td>
          <td>{$isActive}</td>
          <td class="warning"> <a href='categories.php?source=edit_category&edit_category={$row['cat_id']}'>Edit</a></td>
        </tr>
    DELIMETER;

    echo $categories;
  }
}
```

Figure 99: Displaying code to view categories.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 100 below, which also demonstrates how the added features improve the user experience.


Dashboard
Admins Online: 1
Hey, Saddiq
Admin
Categories
When the user clicks here they are redirected to the categories.php page with the source add_category which redirects them to the add_category page.
Add Categories
2

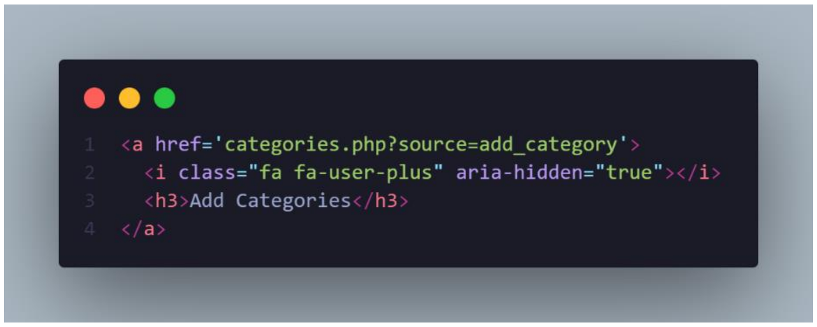
Title	Image	Active	Action
Nutritional Supplements		✓	Edit
Nutritional Fitness & Exercise		✓	Edit
Organic & Natural Foods		✓	Edit
Mental Wellness		✓	Edit
Personal Care		✓	Edit
Skincare & Beauty		✓	Edit
Free Products		✗	Edit

Figure 100: Displaying categories.

As seen in the above image, the categories are displayed successfully with the option to add a product and edit individual records.

Add Category

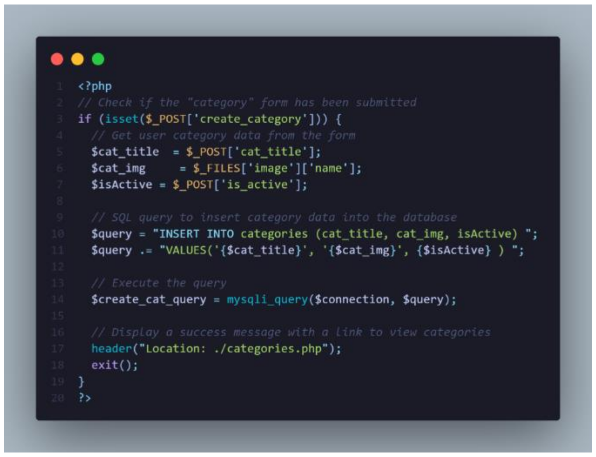
The code displayed in figure 101 below is included in the view all categories page. Whiles on this page we can click the link displayed in the image below. When this link labelled 'Add Categories' is clicked by admins they are redirected to the 'categories.php' page with the query parameter "source=add_category." Which then takes them to the 'add_ category page.



```
<a href='categories.php?source=add_category'>
  <i class="fa fa-user-plus" aria-hidden="true"></i>
  <h3>Add Categories</h3>
</a>
```

Figure 101: Displaying add to category button.

When users land on the add category page and fills out the form by entering the category name, status and uploading the image. The PHP code displayed in figure 102 below then checks if the form has been submitted. If it has, it retrieves information from the form fields on the form such as the category title, image name, and whether or not it is active. Then, it constructs an SQL query to insert this category data into a database table named "categories." Following query execution, it redirects the admin to the "categories.php" page and closes the script. In essence, when a form is submitted, this code handles the creation of a new category and then redirects the user to explore the categories.



```php
<?php
// Check if the "category" form has been submitted
if (isset($_POST['create_category'])) {
  // Get user category data from the form
  $cat_title  = $_POST['cat_title'];
  $cat_img     = $_FILES['image']['name'];
  $isActive = $_POST['is_active'];

  // SQL query to insert category data into the database
  $query = "INSERT INTO categories (cat_title, cat_img, isActive) ";
  $query .= "VALUES('{$cat_title}', '{$cat_img}', {$isActive} ) ";

  // Execute the query
  $create_cat_query = mysqli_query($connection, $query);

  // Display a success message with a link to view categories
  header("Location: ./categories.php");
  exit();
}
?>
```

Figure 102: Displaying code to create a category.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 103 below, which also demonstrates how the added features improve the user experience.

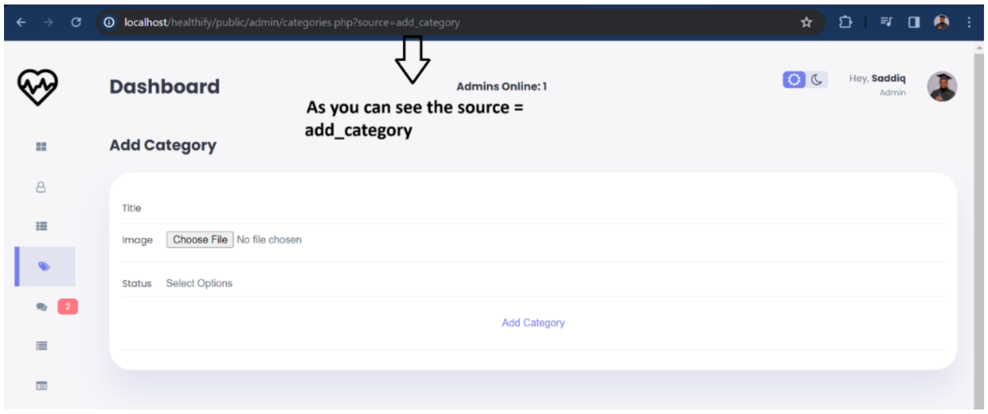
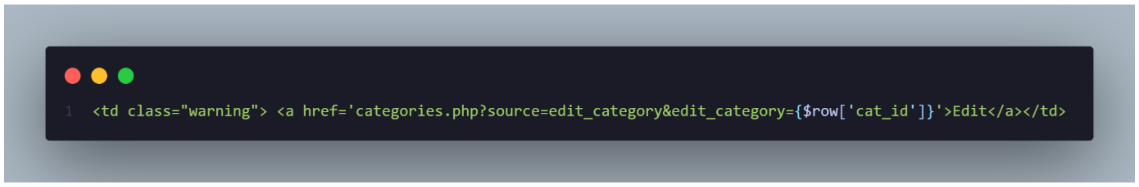


Figure 103: Displaying add to category button.

As seen in the above image, the add category page is displayed successfully.

Edit Category

The code displayed in figure 104 below is included in the view all categories page. Whiles on this page we can click the link displayed in the image below. When this link labelled 'Edit' is clicked by admins they are redirected to the 'categories.php' page with the query parameter "source=edit_category." Which then takes them to the 'edit_ category' page.



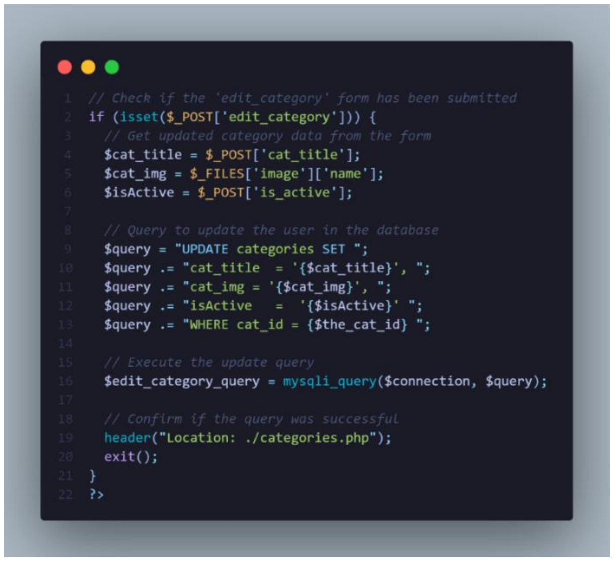
```
<td class="warning"> <a href='categories.php?source=edit_category&edit_category={$row['cat_id']}'>Edit</a></td>
```

Figure 104: Displaying edit category button.

When users navigate to the "edit_category" page and complete the form by providing category details such as the category title, image upload, and specifying its status, the PHP code, as displayed in the provided figure, checks if the form has been submitted. If it has been submitted, the code proceeds to extract the information entered into the form fields, including the category title, image name, and the active status of the category.

Following this, the code constructs an SQL query tailored to update the category's data within the database table, which is named "categories." This query ensures that any modifications made to the category's information are reflected and saved in the database.

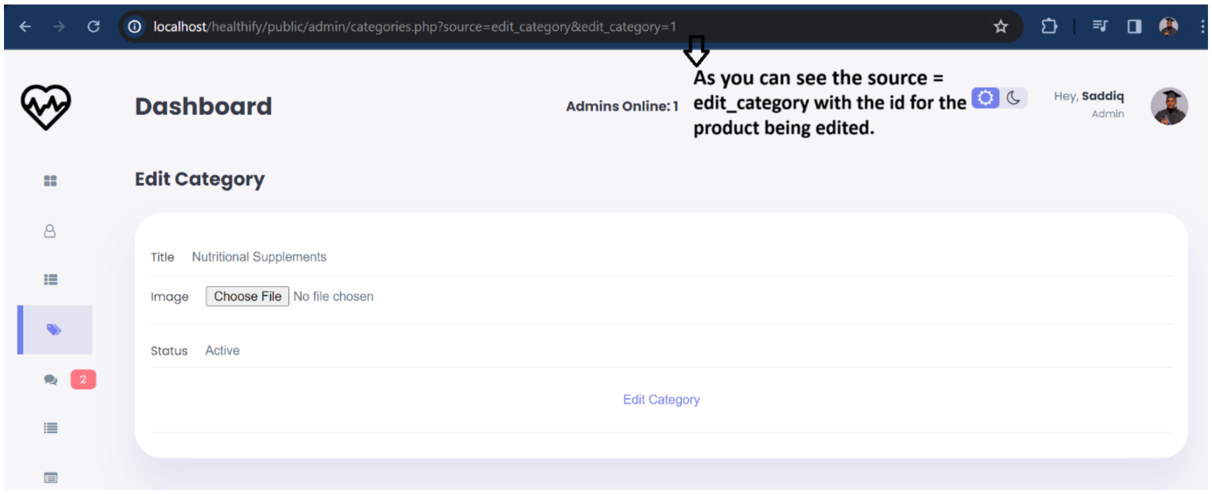
After successfully executing the query, the code performs a redirection action. This redirection sends the admin back to the "categories.php" page, where they can view the updated category information and explore the list of categories.



```
// Check if the 'edit_category' form has been submitted
if (isset($_POST['edit_category'])) {
  // Get updated category data from the form
  $cat_title = $_POST['cat_title'];
  $cat_img = $_FILES['image']['name'];
  $isActive = $_POST['is_active'];

  // Query to update the user in the database
  $query = "UPDATE categories SET ";
  $query .= "cat_title  = '{$cat_title}', ";
  $query .= "cat_img = '{$cat_img}', ";
  $query .= "isActive   =  '{$isActive}' ";
  $query .= "WHERE cat_id = {$the_cat_id} ";

  // Execute the update query
  $edit_category_query = mysqli_query($connection, $query);

  // Confirm if the query was successful
  header("Location: ./categories.php");
  exit();
}
?>
```

Figure 105: Displaying code to edit category.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 106 below, which also demonstrates how the added features improve the user experience.


localhost/healthify/public/admin/categories.php?source=edit_category&edit_category=1

As you can see the source = edit_category with the id for the product being edited.

Dashboard

Admins Online: 1

Hey, Saddiq
Admin

Edit Category

Title Nutritional Supplements

Image Choose File No file chosen

Status Active

Edit Category

Figure 106: Displaying edit category form.

As seen in the above image, the edit category page is displayed successfully.

<u>Conclusion of the Admin Dashboard feature.</u>

In this section, we've delved into the implementation of CRUD (Create, Read, Update, Delete) operations for various aspects of the website, including Users, Reports, Orders, Products, Spinner Winners, and Categories. While Categories served as the example, the same coding style and concept were applied to the other components.

The adoption of CRUD activities enables administrators to execute critical tasks such as creating, reading, updating, and removing data across multiple portions of the website. This ensures flexibility and ease of management, allowing the website's content and functionality to be tailored to changing needs.

To summarize, the Admin Dashboard provides a centralized and efficient interface for administrators to control and monitor all aspects of the website, with the ability to apply CRUD operations to various data categories, such as Users, Reports, Orders, Products, Spinner Winners, and Categories, ensuring a seamless and user-friendly administrative experience.

Advanced Features Implemented

FR17 – Chatbot Integration

Collect.chat was chosen for the website's chatbot feature because of its effective and user-friendly design. With the platform, a personalized chatbot would be possible without requiring complex technological knowledge, all it takes is a few drag-and-drop sentence templates.

The main advantage of utilizing Collect.chat is its simplicity of installation—achieved with a straightforward copy-paste technique—which guarantees that the chatbot is always ready to engage with users around-the-clock. Instant email notifications are sent out whenever a visitor engages with the chatbot and responds, providing real-time updates to the business.

Additionally, Collect.chat gives you the freedom to build chatbots that can speak to a variety of audiences by supporting several languages. To maximize visitor interaction, it also offers capabilities that allow you to customize the chatbot's appearance based on time and place. Important benefits include the ability to integrate with different marketing platforms and the inclusion of built-in visualization tools. With the use of these technologies, companies may monitor chatbot interactions, examine client feedback, and learn more about the preferences and behavior of their customers. Businesses can improve consumer engagement and spur corporate success by refining their services and plans with the aid of this invaluable information.

BACK END | Let's take a closer look at the chat bot functionality (How the functionally was implemented):

The essential stage of design marked the starting point of our Collect.chat chatbot development process. Choosing an appropriate avatar and theme that complement the brand identity and website aesthetics was the first step in this process. Making sure the chatbot's visual components were captivating and appealing to the target market was the main goal of the design process. The goal in personalizing the look was to produce a chatbot that was not only useful but also aesthetically pleasing, improving the user experience from the very beginning. See figure 107 below for the user interface used to design the chatbot.

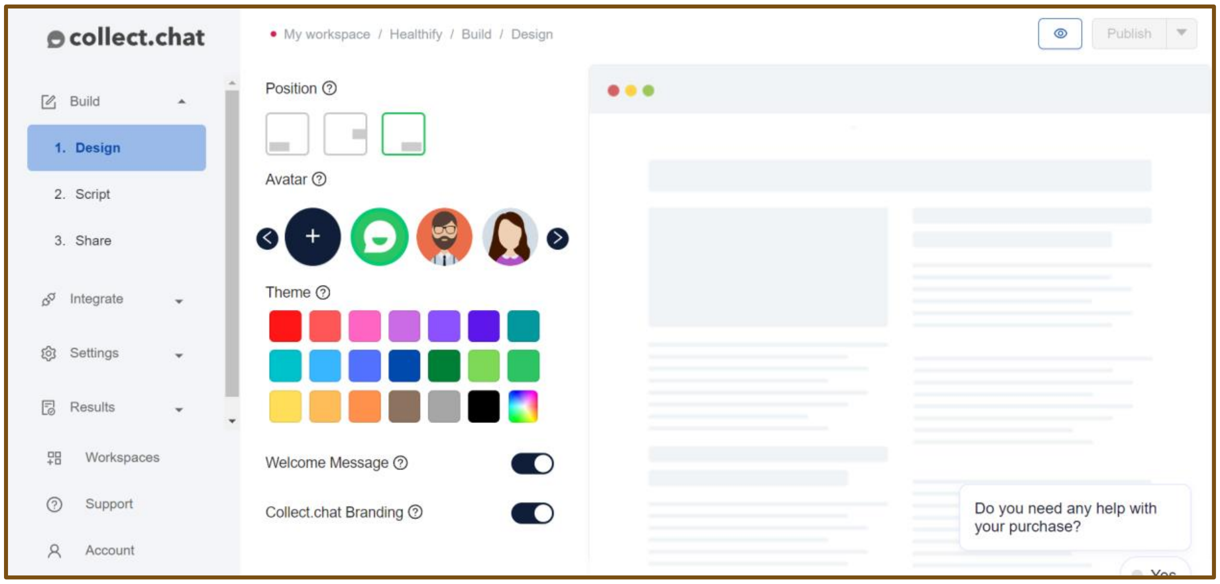
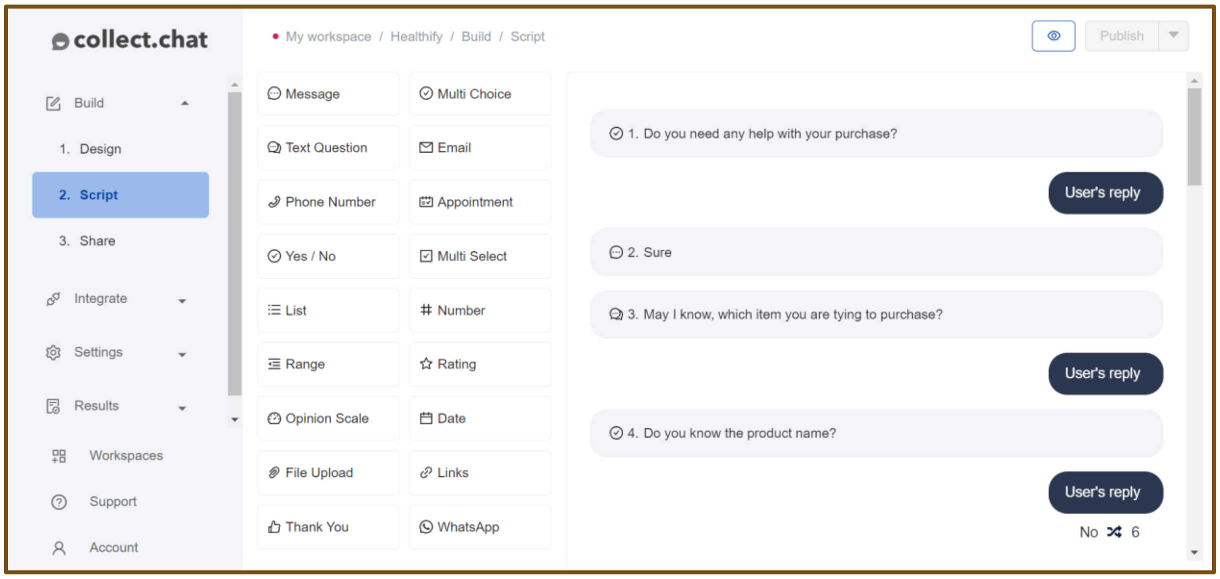


Figure 107: Displaying chatbot design page.

The next stage after the design phase was to script the chatbot. This entailed creating conversation flows and responses that would be used by the chatbot to interact with website visitors. The screenplay was written to answer a wide range of probable questions in an instructive and helpful manner. This stage necessitated careful consideration of vocabulary and tone to ensure that the chatbot communicated effectively, accurately portraying our brand voice, and addressing the needs of our audience. See figure 108 below for the user interface used to script the chatbot.



1. Do you need any help with your purchase?

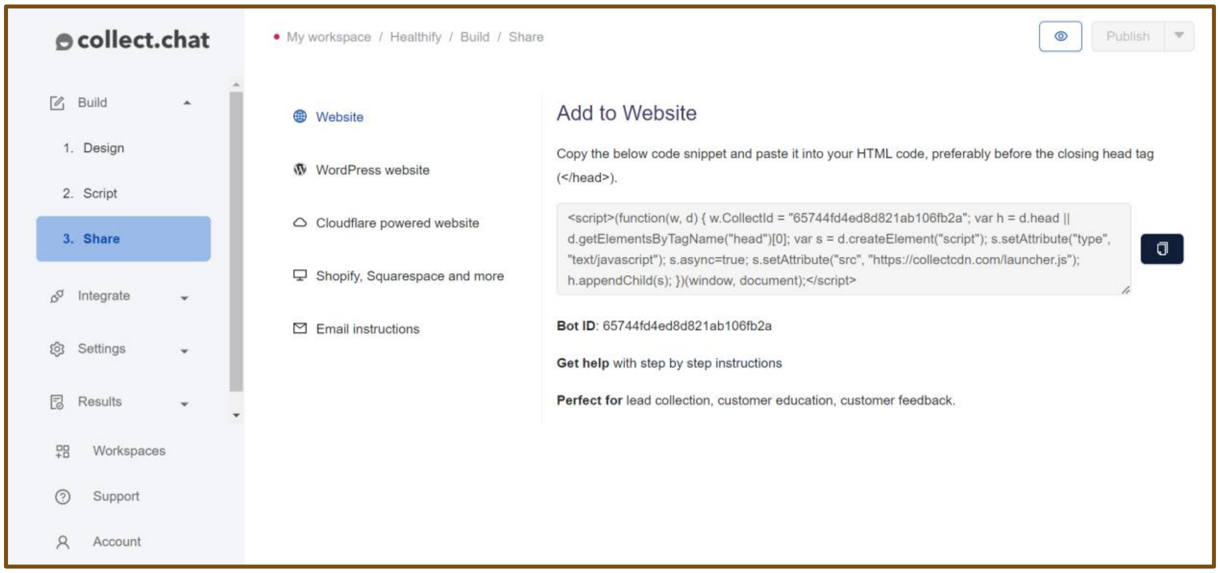
2. Sure

3. May I know, which item you are tying to purchase?

4. Do you know the product name?

Figure 108: Displaying chatbot script design page.

The last stage in bringing our chatbot into action was to share and integrate it with our website. This entailed a simple procedure of embedding the chatbot into our web pages, ensuring that users could access it at critical touchpoints. The connection supposed to be seamless, providing visitors with fast support and improving their overall experience on our website. This chatbot will boost user engagement, provide immediate service, and gain useful data from user interactions by making the chatbot easily accessible. See figure 109 below to see how we got the script to link the chatbot with our website.



Add to Website

Copy the below code snippet and paste it into your HTML code, preferably before the closing head tag (</head>).

```
<script>(function(w, d) { w.CollectId = "65744fd4ed8d821ab106fb2a"; var h = d.head ||
d.getElementsByTagName("head")[0]; var s = d.createElement("script"); s.setAttribute("type",
"text/javascript"); s.async=true; s.setAttribute("src", "https://collectcdn.com/launcher.js");
h.appendChild(s); })(window, document);</script>
```

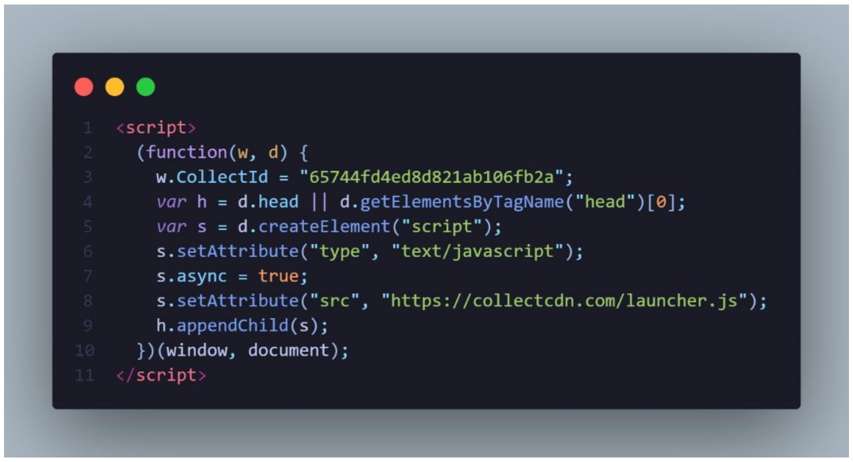
Bot ID: 65744fd4ed8d821ab106fb2a

Get help with step by step instructions

Perfect for lead collection, customer education, customer feedback.

Figure 109: Displaying chatbot share page.

The script was simply inserted into the head section of the header on each page to make the chatbot available on the necessary pages of the website. This guarantees that the chatbot is integrated and available to users who visit these specific locations of our website. See figure 110 below for the code that was inserted in the head section.

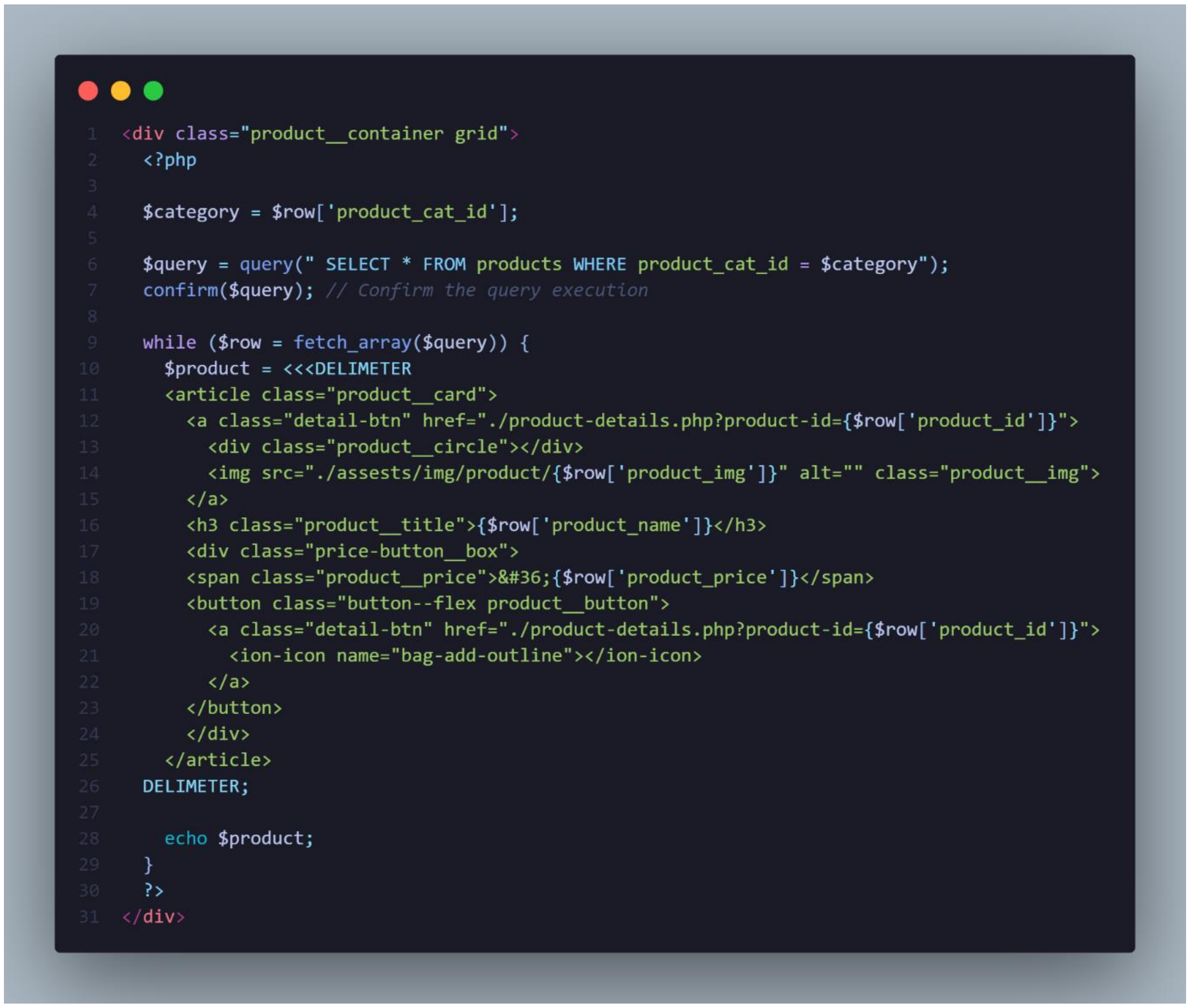


```
<script>
  (function(w, d) {
    w.CollectId = "65744fd4ed8d821ab106fb2a";
    var h = d.head || d.getElementsByTagName("head")[0];
    var s = d.createElement("script");
    s.setAttribute("type", "text/javascript");
    s.async = true;
    s.setAttribute("src", "https://collectcdn.com/launcher.js");
    h.appendChild(s);
  })(window, document);
</script>
```

Figure 110: Displaying chatbot script.

FR18 – Recommendation Engine

The provided code below in figure 111 is used for the recommendation engine on the website's product details page. It shows products in the same category as the one the user is now viewing. This feature improves the purchasing experience by recommending relevant products to clients. The PHP script initially obtains the category ID of the currently displayed product. It then searches the database for further products in the same category, iterating through the results. The script outputs a block of HTML for each product identified, which includes a clickable image going to the product's detail page, the product name, and the price. This is presented in a 'product__card' layout, which offers a visually appealing and unified user experience. The utilization of the 'bag-add-outline' and 'detail-btn' icons on every product card improves user engagement by providing effortless access to comprehensive product details and the ability to add goods to the shopping cart.



```
<div class="product__container grid">
  <?php

  $category = $row['product_cat_id'];

  $query = query(" SELECT * FROM products WHERE product_cat_id = $category");
  confirm($query); // Confirm the query execution

  while ($row = fetch_array($query)) {
    $product = <<<DELIMETER
    <article class="product__card">
      <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
        <div class="product__circle"></div>
        <img src="./assests/img/product/{$row['product_img']}" alt="" class="product__img">
      </a>
      <h3 class="product__title">{$row['product_name']}</h3>
      <div class="price-button__box">
      <span class="product__price">&#36;{$row['product_price']}</span>
      <button class="button--flex product__button">
        <a class="detail-btn" href="./product-details.php?product-id={$row['product_id']}">
          <ion-icon name="bag-add-outline"></ion-icon>
        </a>
      </button>
      </div>
    </article>
  DELIMETER;

    echo $product;
  }
  ?>
</div>
```

Figure 111: Displaying code for recommendation engine.

FR19 – Gamification Features

Overall Explanation of the functionality

In this section, we delve into the exciting gamification feature implemented on the platform, offering a fun and interactive way for users to engage. Our feature is centred around a captivating spinner game, where participants get the chance to spin a vibrant wheel marked with numbers from 1 to 20. The game's thrill lies in the unpredictability of the spinner's landing point, creating an atmosphere of anticipation and excitement.

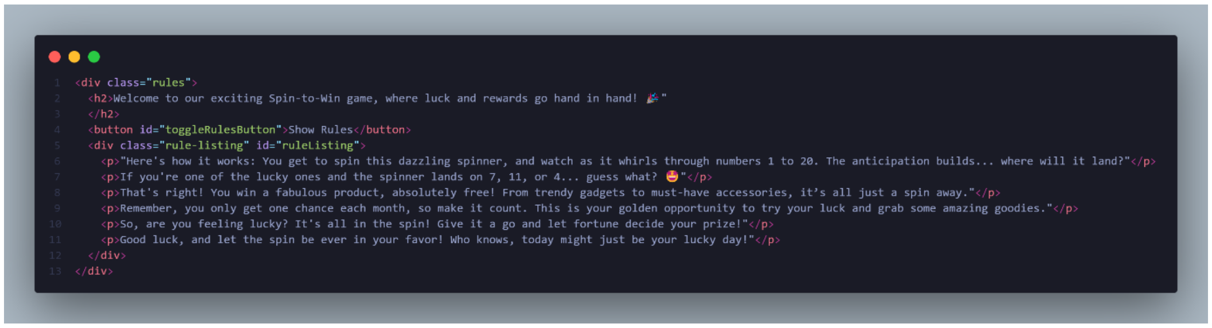
The rules are straightforward yet thrilling: if the spinner lands on the numbers 7, 11, or 4, the participant wins an amazing product for free! These prizes range from the latest gadgets to essential accessories, making each spin a potential gateway to a delightful surprise.

Importantly, participants are granted one opportunity each month to spin the wheel, adding a layer of strategy to the game. This limitation encourages users to cherish their monthly chance and adds a special significance to each spin. The game not only enhances user engagement but also enriches their experience on the platform, fostering a sense of community and excitement. As they spin the wheel, they're left wondering: will today be their lucky day? This feature not only entertains but also deepens user involvement, showcasing the commitment to innovative and enjoyable user experiences.

BACK END | Let's take a closer look at the gamification process process (How the functionally was implemented):

Before the user can spin the can read the rules:

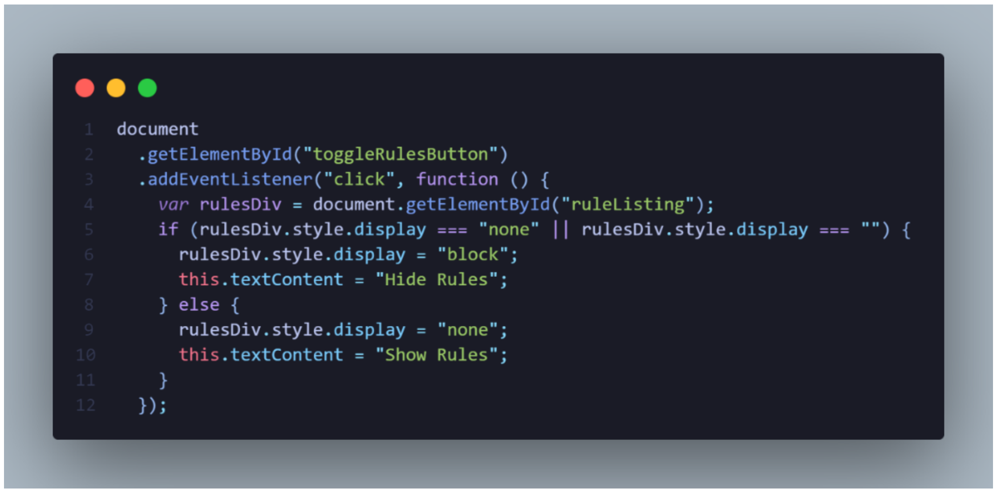
Users are provided with the chance to read and understand the rules of the spinner game before participating in the gamification feature. This is critical for ensuring a thorough understanding of how the game operates. The HTML framework that displays the game rules is depicted in Figure 112. This configuration allows users to become acquainted with the game mechanics, such as the winning numbers and the awards linked with them.



```html
<div class="rules">
  <h2>Welcome to our exciting Spin-to-Win game, where luck and rewards go hand in hand! 🎉"
  </h2>
  <button id="toggleRulesButton">Show Rules</button>
  <div class="rule-listing" id="ruleListing">
    <p>"Here's how it works: You get to spin this dazzling spinner, and watch as it whirls through numbers 1 to 20. The anticipation builds... where will it land?"</p>
    <p>If you're one of the lucky ones and the spinner lands on 7, 11, or 4... guess what? 🤩"</p>
    <p>That's right! You win a fabulous product, absolutely free! From trendy gadgets to must-have accessories, it’s all just a spin away."</p>
    <p>Remember, you only get one chance each month, so make it count. This is your golden opportunity to try your luck and grab some amazing goodies."</p>
    <p>So, are you feeling lucky? It's all in the spin! Give it a go and let fortune decide your prize!"</p>
    <p>Good luck, and let the spin be ever in your favor! Who knows, today might just be your lucky day!"</p>
  </div>
</div>
```

Figure 112: Displaying code for spinner game rules.

Figure 113 shows the JavaScript code responsible for the rules section's toggle functionality in addition to the HTML structure. This rational addition allows users to simply open, read, and close the rules as needed. It improves the user experience by giving them an interactive and user-friendly approach to access game information without bombarding them with constant on-screen text. By implementing this toggle mechanism, we ensure that users have a smooth and delightful experience when playing the spinning game, are well informed about the rules, and feel confident about their chances of winning.

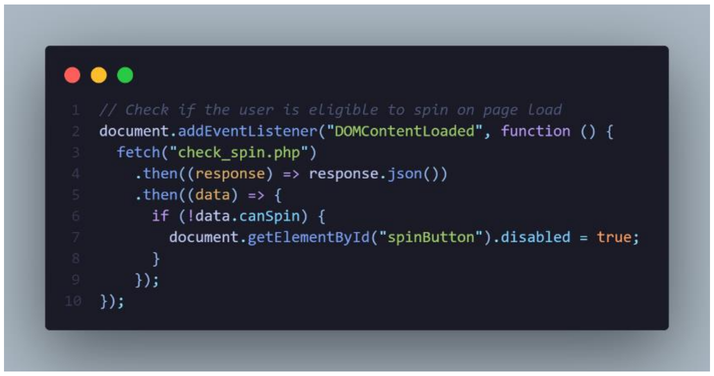


```
document
  .getElementById("toggleRulesButton")
  .addEventListener("click", function () {
    var rulesDiv = document.getElementById("ruleListing");
    if (rulesDiv.style.display === "none" || rulesDiv.style.display === "") {
      rulesDiv.style.display = "block";
      this.textContent = "Hide Rules";
    } else {
      rulesDiv.style.display = "none";
      this.textContent = "Show Rules";
    }
  });
```

Figure 113: Displaying code of the spinner rules toggle button.

After the user read the rules (optional) the system decides whether the user is eligible to spin:

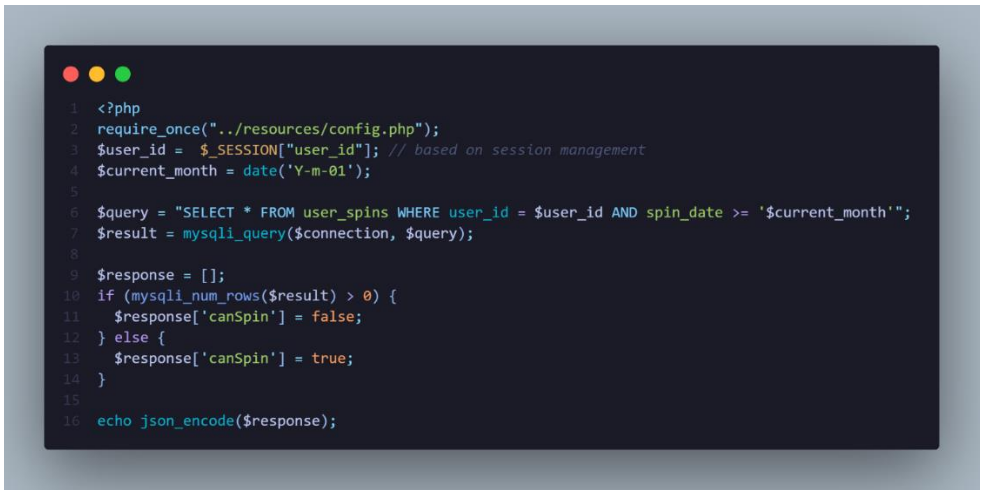
The JavaScript snippet shown in figure 114 below depicts a function that runs when the page loads. This function is critical for the spinner game since it determines whether or not a user is eligible to spin. This is accomplished by submitting a request to "check_spin.php". If the user is ineligible to play (for example, they have already played this month), the function disables the spin button, ensuring that the game rules are followed and preventing additional spins within the timeframe.



```
// Check if the user is eligible to spin on page load
document.addEventListener("DOMContentLoaded", function () {
  fetch("check_spin.php")
    .then((response) => response.json())
    .then((data) => {
      if (!data.canSpin) {
        document.getElementById("spinButton").disabled = true;
      }
    });
});
```

Figure 114: Displaying code that decide eligibility.

The PHP code in "check_spin.php" that goes with it is critical in assessing user eligibility. It obtains the user's ID from the session and searches the 'user_spins' table for any spins recorded during the current month. If the user has already spun the wheel in the current month, the script changes the value of 'canSpin' to false, indicating that the user is not eligible for another spin. If no spins are detected for the user in the current month, the variable 'canSpin' is set to true, allowing them to play the game. This PHP code ensures that the game's one-spin-per-month regulation is followed. Figure 115 below displays the code in the “check_spin.php”



```php
<?php
require_once("../resources/config.php");
$user_id =  $_SESSION["user_id"]; // based on session management
$current_month = date('Y-m-01');

$query = "SELECT * FROM user_spins WHERE user_id = $user_id AND spin_date >= '$current_month'";
$result = mysqli_query($connection, $query);

$response = [];
if (mysqli_num_rows($result) > 0) {
  $response['canSpin'] = false;
} else {
  $response['canSpin'] = true;
}

echo json_encode($response);
```

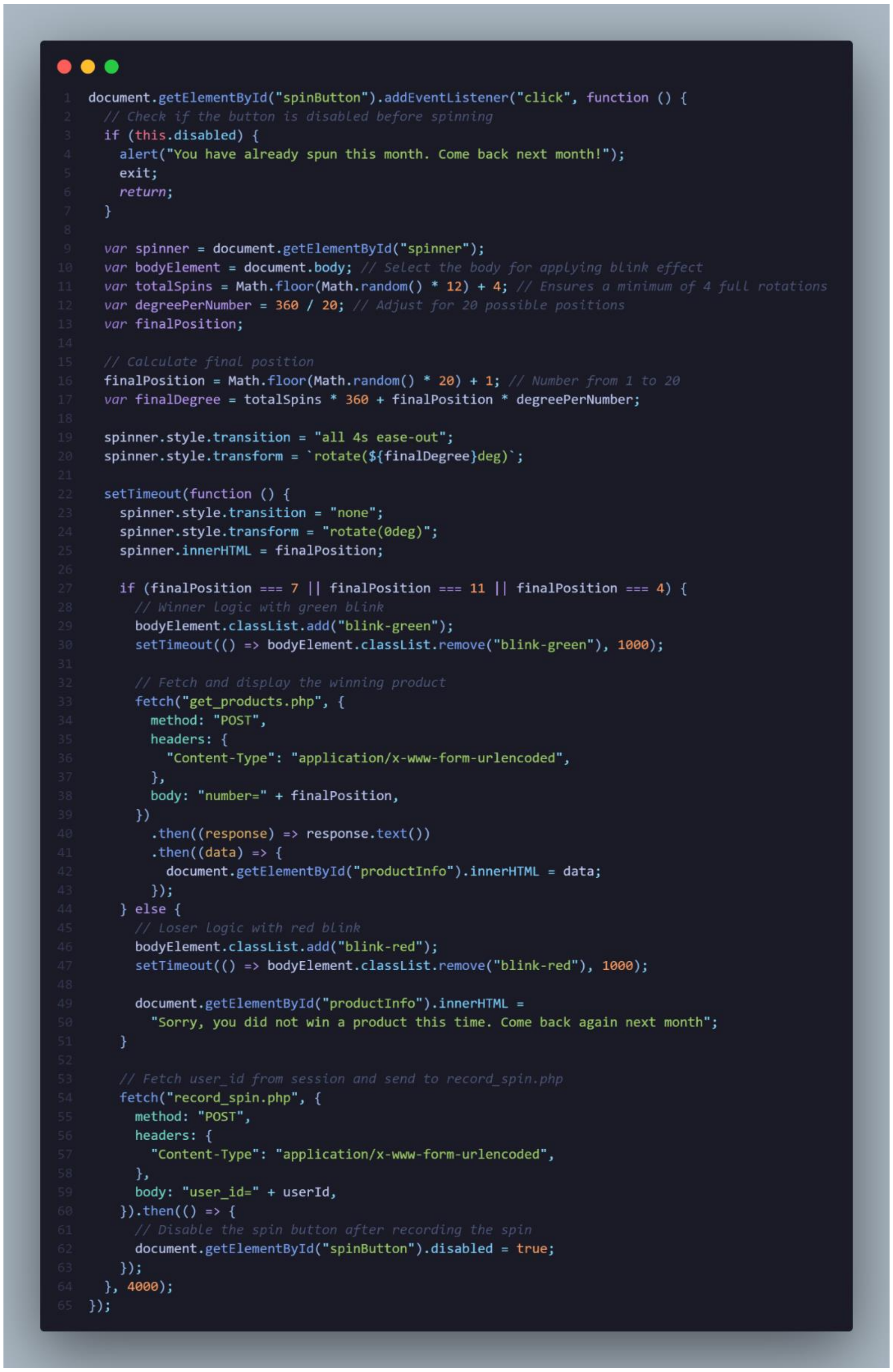
Figure 115: Displaying code in the check_spin.php file.

Core functionality of the spinner:

The JavaScript code displayed in image 116 below controls the "Spin the Wheel" game's fundamental feature. When a user presses the "spin" button, the code first determines whether the button is disabled, indicating that the user has already played this month. If this is the case, an alert alerts them to return the following month. Assuming the user is eligible to participate, the script activates the spinner, generating a random final position between 1 and 20, containing the winning numbers 7, 11, and 4. Because of the computed degree of rotation and a 4-second transition, the spinner animates smoothly.

If the spinner falls on a winning number after the spin, a "green blink" effect is activated to indicate victory. The script then retrieves information from "get_products.php" in order to present the winning product. If a non-winning number is selected, a "red blink" effect emerges, and the user is informed of their non-winning status.

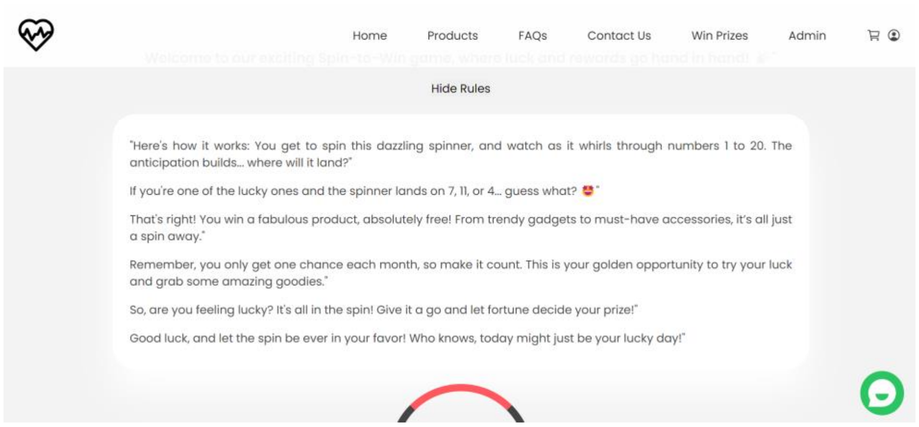
Lastly an HTTP POST request is made to "record_spin.php", which stores the user's spin in the database. This request updates the user's eligibility and disables the spin button to prevent future spins within the current month. This code contains the game's mechanics, giving users a compelling and fair experience.



```
document.getElementById("spinButton").addEventListener("click", function () {
  // Check if the button is disabled before spinning
  if (this.disabled) {
    alert("You have already spun this month. Come back next month!");
    exit;
    return;
  }

  var spinner = document.getElementById("spinner");
  var bodyElement = document.body; // Select the body for applying blink effect
  var totalSpins = Math.floor(Math.random() * 12) + 4; // Ensures a minimum of 4 full rotations
  var degreePerNumber = 360 / 20; // Adjust for 20 possible positions
  var finalPosition;

  // Calculate final position
  finalPosition = Math.floor(Math.random() * 20) + 1; // Number from 1 to 20
  var finalDegree = totalSpins * 360 + finalPosition * degreePerNumber;

  spinner.style.transition = "all 4s ease-out";
  spinner.style.transform = `rotate(${finalDegree}deg)`;

  setTimeout(function () {
    spinner.style.transition = "none";
    spinner.style.transform = "rotate(0deg)";
    spinner.innerHTML = finalPosition;

    if (finalPosition === 7 || finalPosition === 11 || finalPosition === 4) {
      // Winner logic with green blink
      bodyElement.classList.add("blink-green");
      setTimeout(() => bodyElement.classList.remove("blink-green"), 1000);

      // Fetch and display the winning product
      fetch("get_products.php", {
        method: "POST",
        headers: {
          "Content-Type": "application/x-www-form-urlencoded",
        },
        body: "number=" + finalPosition,
      })
        .then((response) => response.text())
        .then((data) => {
          document.getElementById("productInfo").innerHTML = data;
        });
    } else {
      // Loser logic with red blink
      bodyElement.classList.add("blink-red");
      setTimeout(() => bodyElement.classList.remove("blink-red"), 1000);

      document.getElementById("productInfo").innerHTML =
        "Sorry, you did not win a product this time. Come back again next month";
    }

    // Fetch user_id from session and send to record_spin.php
    fetch("record_spin.php", {
      method: "POST",
      headers: {
        "Content-Type": "application/x-www-form-urlencoded",
      },
      body: "user_id=" + userId,
    }).then(() => {
      // Disable the spin button after recording the spin
      document.getElementById("spinButton").disabled = true;
    });
  }, 4000);
});
```

Figure 116: Displaying code for the core functionality of the spinner game.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 117 below, which also demonstrates how the added features improve the user experience.



"Here's how it works: You get to spin this dazzling spinner, and watch as it whirls through numbers 1 to 20. The anticipation builds... where will it land?"

If you're one of the lucky ones and the spinner lands on 7, 11, or 4... guess what? 😍"

That's right! You win a fabulous product, absolutely free! From trendy gadgets to must-have accessories, it's all just a spin away."

Remember, you only get one chance each month, so make it count. This is your golden opportunity to try your luck and grab some amazing goodies."

So, are you feeling lucky? It's all in the spin! Give it a go and let fortune decide your prize!"

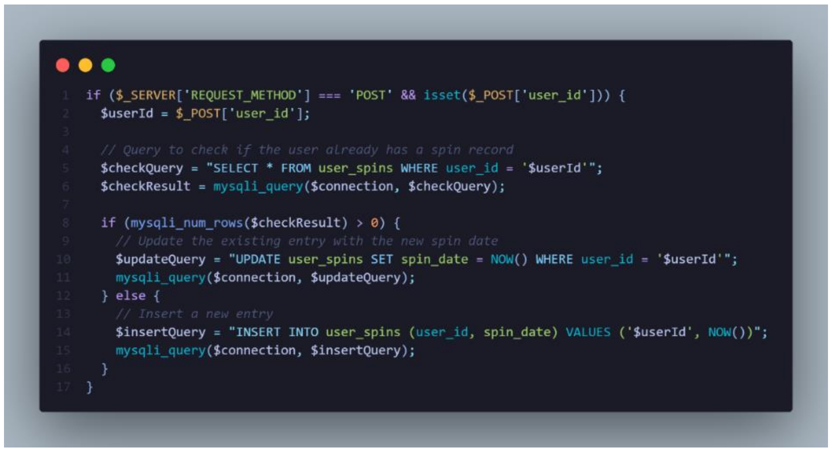
Good luck, and let the spin be ever in your favor! Who knows, today might just be your lucky day!"

Figure 117: Displaying rules for the spinner game.

As seen in the above image, the rules of the spin game are display successfully.

As mentioned in the core functionality section above, when a user spins, it is recorded to prevent another spin for the next month -- How is the spin recorded:

When a POST request is received with the 'user_id' the script provided below first the user_spins table is checked to see if there's an existing record for the given user-id. If there's an existing 'user_id' this indicates the user have played before therefore, the script then updates the existing record for the user by setting the spin-date to the current time (using 'NOW()'). If there's no record found this indicates it's the user first spin therefore the script creates a new entry in the user_spins table with their user_id and the current date/time. This code effectively logs each spin, ensuring that users can only play once in the given timeframe as per the game's rules.



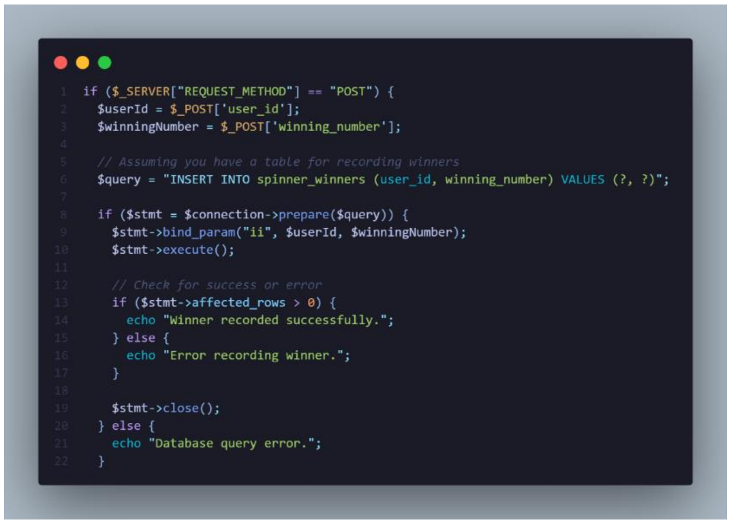
```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['user_id'])) {
  $userId = $_POST['user_id'];

  // Query to check if the user already has a spin record
  $checkQuery = "SELECT * FROM user_spins WHERE user_id = '$userId'";
  $checkResult = mysqli_query($connection, $checkQuery);

  if (mysqli_num_rows($checkResult) > 0) {
    // Update the existing entry with the new spin date
    $updateQuery = "UPDATE user_spins SET spin_date = NOW() WHERE user_id = '$userId'";
    mysqli_query($connection, $updateQuery);
  } else {
    // Insert a new entry
    $insertQuery = "INSERT INTO user_spins (user_id, spin_date) VALUES ('$userId', NOW())";
    mysqli_query($connection, $insertQuery);
  }
}
```

Figure 118: Displaying code that records a spin.

Also mentioned in the core functionality section above, when a user win (gets the correct number), it is recorded in the database -- How is this recorded:

This feature is part of the backend, specifically for recording the winners. Thie winners is recorded using a PHP script displayed in figure 119 below. The script first checks if the request is a POST request, ensuring the data is being sent correctly. The script receives the user ID ($'user_id') and the winning number ('$winningNumber') from the POST request. These variables represent the user who played the game and the number they landed on.

The script then uses prepared statements for secure data binding and SQL injection prevention when inserting a new winning record into the "spinner_winners" table. After running the query, it determines whether the insertion was successful and outputs the relevant message. This code essentially logs each winning spin in the database, crucial for tracking game outcomes and winner analytics.



```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $userId = $_POST['user_id'];
  $winningNumber = $_POST['winning_number'];

  // Assuming you have a table for recording winners
  $query = "INSERT INTO spinner_winners (user_id, winning_number) VALUES (?, ?)";

  if ($stmt = $connection->prepare($query)) {
    $stmt->bind_param("ii", $userId, $winningNumber);
    $stmt->execute();

    // Check for success or error
    if ($stmt->affected_rows > 0) {
      echo "Winner recorded successfully.";
    } else {
      echo "Error recording winner.";
    }

    $stmt->close();
  } else {
    echo "Database query error.";
  }
```

Figure 119: Displaying the code that records when a user wins.

To view how this spin game looks on the website. A screenshot of the feature is displayed in figure 120 below, which also demonstrates how the added features improve the user experience.

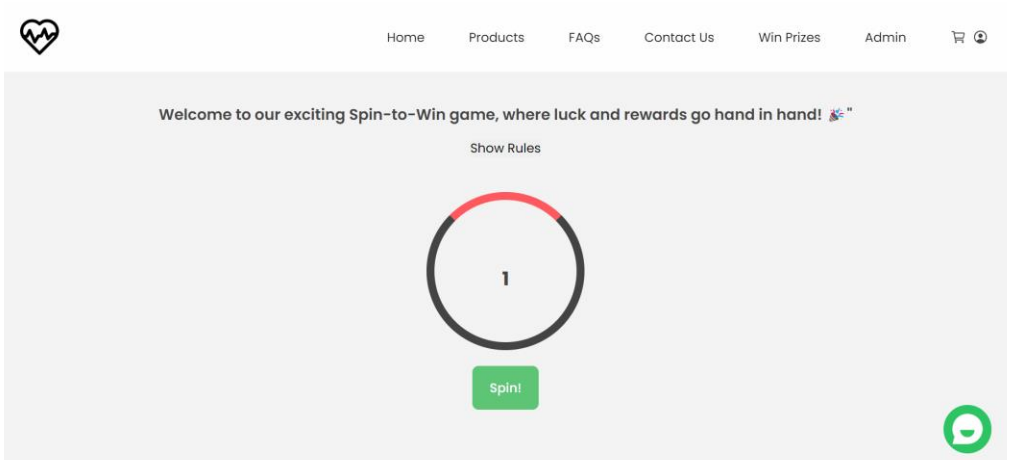
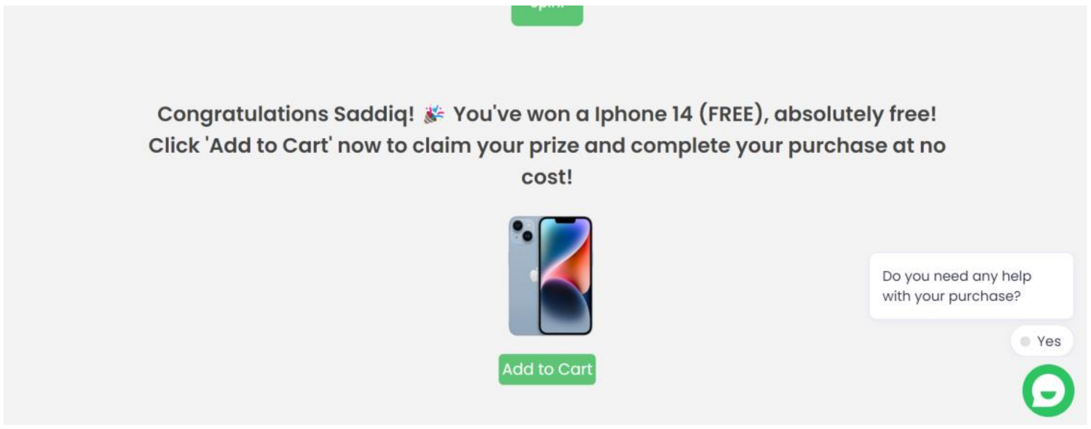

Home
Products
FAQs
Contact Us
Win Prizes
Admin
Welcome to our exciting Spin-to-Win game, where luck and rewards go hand in hand! 🎉"
Show Rules
1
Spin!

Figure 120: Displaying the spin game look.

As seen in the above image, game is displayed successfully.

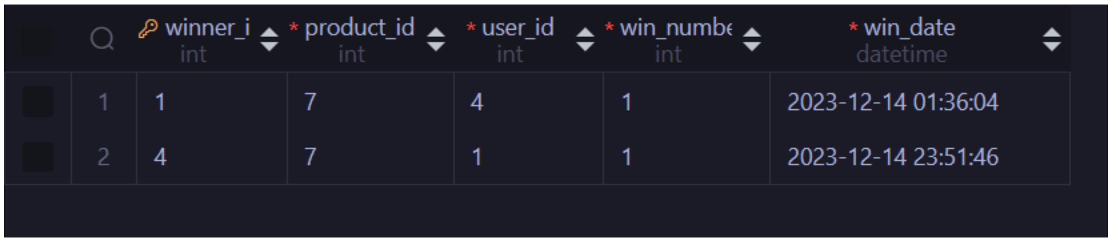
An example of what the user sees when they win the spinner game is displayed below in figure.


Congratulations Saddiq! 🎉 You've won a Iphone 14 (FREE), absolutely free! Click 'Add to Cart' now to claim your prize and complete your purchase at no cost!

Do you need any help with your purchase?

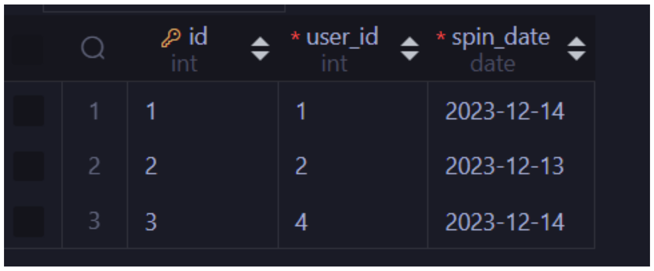
Figure 121: Displaying what the user sees when they win the spinner game.

Figure 122 and 123 displays evidence that the database is updated when the user plays the spin game and when the user wins the game.



	winner_i int	* product_id int	* user_id int	* win_numbe int	* win_date datetime
1	1	7	4	1	2023-12-14 01:36:04
2	4	7	1	1	2023-12-14 23:51:46

Figure 122: Displaying evidence that the database is updated when the user plays the spin game.



	id int	user_id int	spin_date date
1	1	1	2023-12-14
2	2	2	2023-12-13
3	3	4	2023-12-14

Figure 123: Displaying evidence that the database is updated when the user plays the spin game.

FR20 – Advanced Analytics and Data Visualization

Overall Explanation of the functionality

This section delves into the advanced data analytics and visualization functionality that the platform has incorporated. This advanced feature enables, administrators to examine and display a vast range of data that is taken straight from the website's database. A thorough summary of the website's activity is provided by key insights such as the total number of products, user reviews, and registered users.

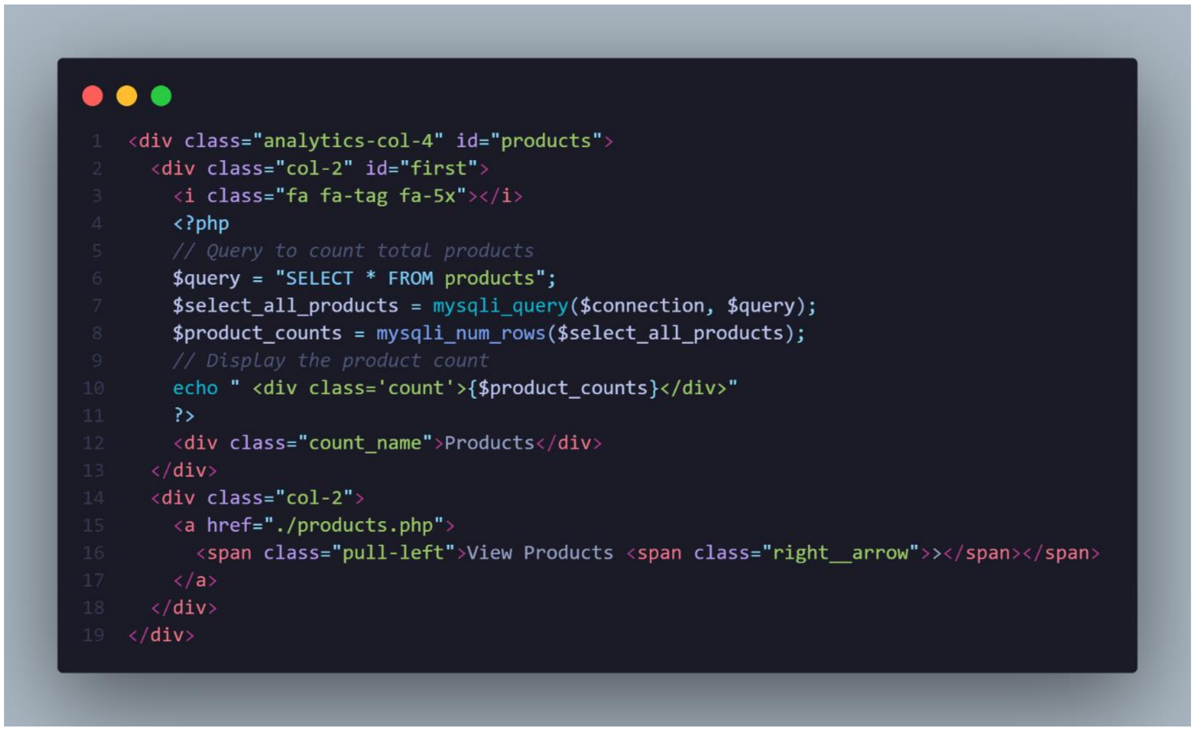
Administrators may also see daily income totals, track the quantity of products sold, and check the number of orders handled. This feature also includes displaying the total amount of money the business has made, the number of new clients it has gained in the last week, and the total number of orders it has received.

This analytics package is further improved with an interactive bar chart that provides a visual comparison of approved versus unapproved reviews and active versus draft products. The tool also ensures that administrators are always aware of the most recent changes and trends on the website by offering up-to-date notifications and insights. This comprehensive analytics capacity allows for efficient maintenance and expansion of the web platform, making it a priceless instrument for strategic planning and decision-making.

Let's take a closer look at the Advanced Analytics and Data Visualization implementation (How the functionality was implemented):

Viewing the total number of products, reviews, users, and orders:

The code displayed in figure 124 below illustrates how this was done. A PHP script was used to execute an SQL query, which retrieves all entries from the products table in the database. This is achieved using the ‘mysqli_query’ function, which runs the query. The total number of products is determined by counting the number of rows returned from the query using the ‘mysql_num_rows’ function and is then displayed on the website. This was then repeated to also get the total number of reviews, users, and orders. See figure 124 below for actual code used to accomplish this.



```
<div class="analytics-col-4" id="products">
  <div class="col-2" id="first">
    <i class="fa fa-tag fa-5x"></i>
    <?php
    // Query to count total products
    $query = "SELECT * FROM products";
    $select_all_products = mysqli_query($connection, $query);
    $product_counts = mysqli_num_rows($select_all_products);
    // Display the product count
    echo " <div class='count'>{$product_counts}</div>"
    ?>
    <div class="count_name">Products</div>
  </div>
  <div class="col-2">
    <a href="./products.php">
      <span class="pull-left">View Products <span class="right__arrow">></span></span>
    </a>
  </div>
</div>
```

Figure 124: Displaying the code for viewing the total number of products, reviews, users, and orders.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in Figure 125 below, which also demonstrates how the added features improve the user experience.

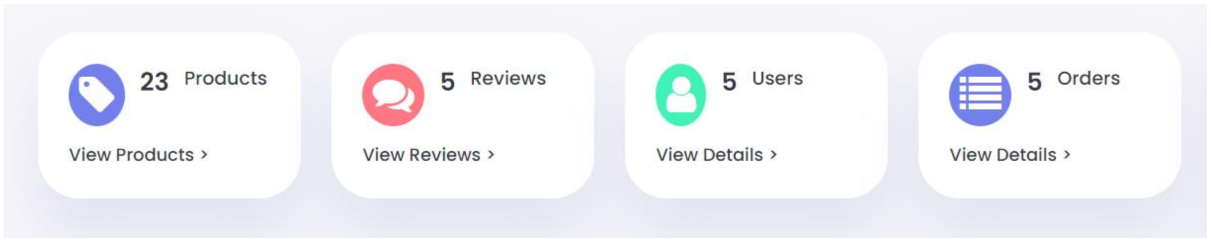
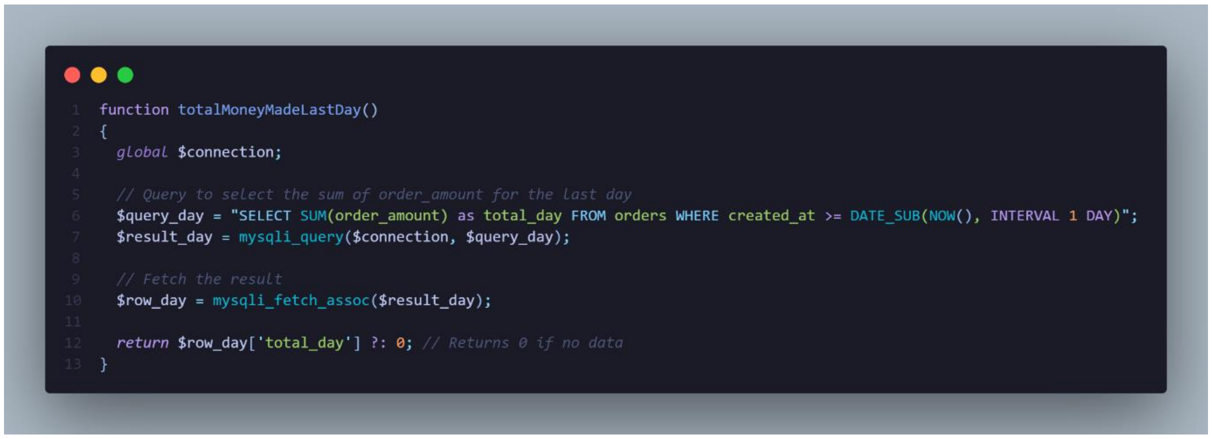


Figure 125: Displaying outcome of this element/function.

As seen in the above image, the total number of products, reviews, users, and orders are displayed successfully.

Viewing the total number (not percentage) of the daily income totals, the quantity of products sold, and total number of orders made:

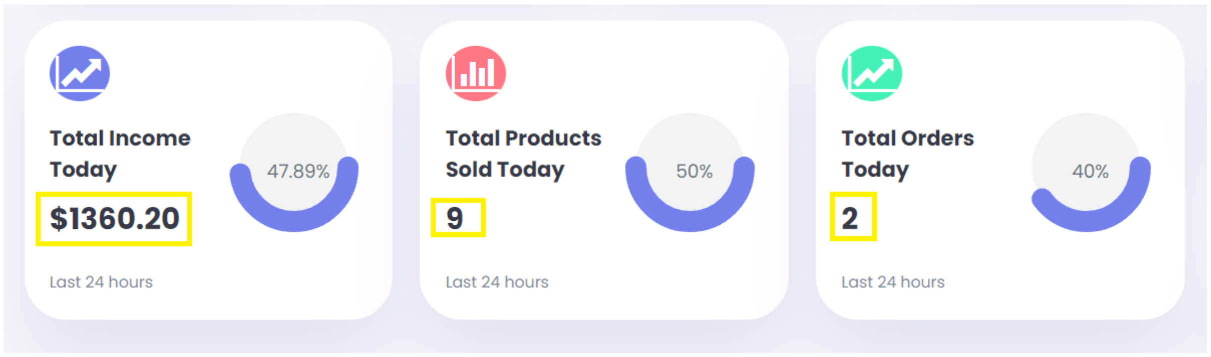
The code displayed in Figure 126 below illustrates how the total number (not percentage) of the daily income totals, the quantity of products sold, and total number of orders made were calculated. A PHP function named 'totalMoneyMadeLastDay' was used, the function calculates the total amount of money made in the last day. A query was used to select the sum of 'order-amount' from the 'orders' table for orders made on the last day using the 'DATE_SUB(NOW(), INTERVAL 1 DAY)' function to filter the data. After running the query with 'mysqli_query', the function fetches the result and returns the total sum. If there is no data (no orders made on the last day) it returns 0. The exact concept of this function was used to get the quantity of products sold, and total number of orders made over the past day.



```php
function totalMoneyMadeLastDay()
{
  global $connection;

  // Query to select the sum of order_amount for the last day
  $query_day = "SELECT SUM(order_amount) as total_day FROM orders WHERE created_at >= DATE_SUB(NOW(), INTERVAL 1 DAY)";
  $result_day = mysqli_query($connection, $query_day);

  // Fetch the result
  $row_day = mysqli_fetch_assoc($result_day);

  return $row_day['total_day'] ?: 0; // Returns 0 if no data
}
```

Figure 126: Displaying how this the total number (not percentage) for the various analytics was calculated.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 127 below, which also demonstrates how the added features improve the user experience.



Total Income Today
$1360.20
47.89%
Last 24 hours

Total Products Sold Today
9
50%
Last 24 hours

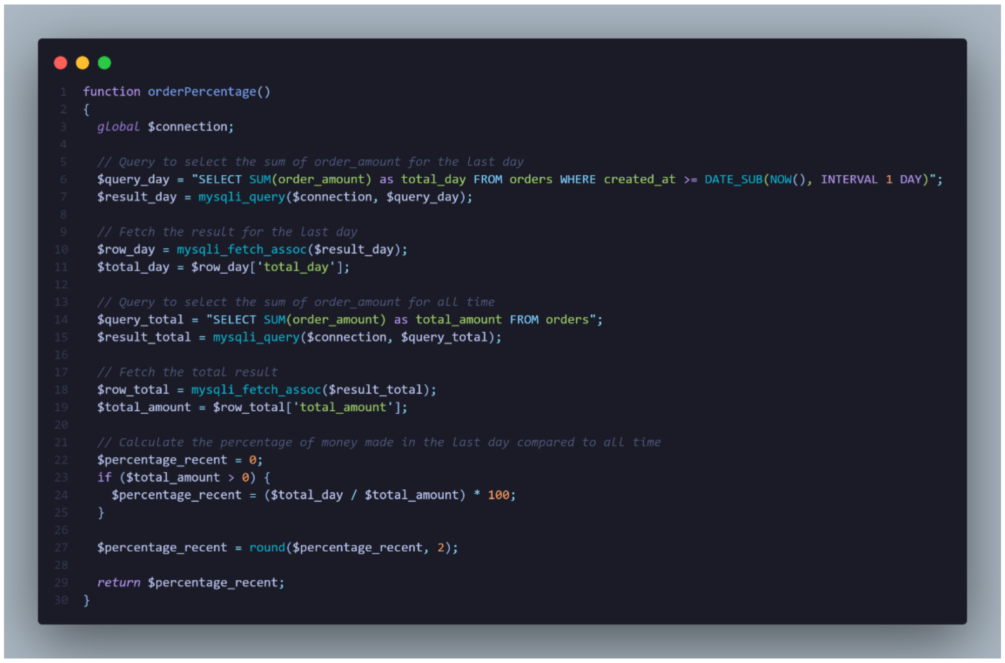
Total Orders Today
2
40%
Last 24 hours

Figure 127: Displaying the outcome of this element/function on the website.

As seen in the above image, total number of the daily income totals, the quantity of products sold, and total number of orders (highlighted in the yellow rectangle) made are display successfully.

Viewing the percentage (not number) of the daily income totals, the quantity of products sold, and total number of orders made:

The code displayed in figure 128 below illustrates how this the percentage (not number) of the daily income totals, the quantity of products sold, and total number of orders made was calculated. A PHP function named 'orderPercentage' was used, to calculate the percentage of the total revenue generated in the last day compared to the overall revenue since the beginning. The function executes two SQL queries: one to sum up the 'order-amount' for orders made in the last day, and another to sum up the 'order amount' for all the orders ever made. After retrieving the two sums from the database, the function calculates the percentage of the day's revenue as a proportion of the total revenue. This percentage is then rounded by two decimal places. The feature allows us to analyse how recent sales compare to the historical performance of the business. The exact concept of this function was used to get the percentage of quantity of products sold, and total number of orders over the past day.

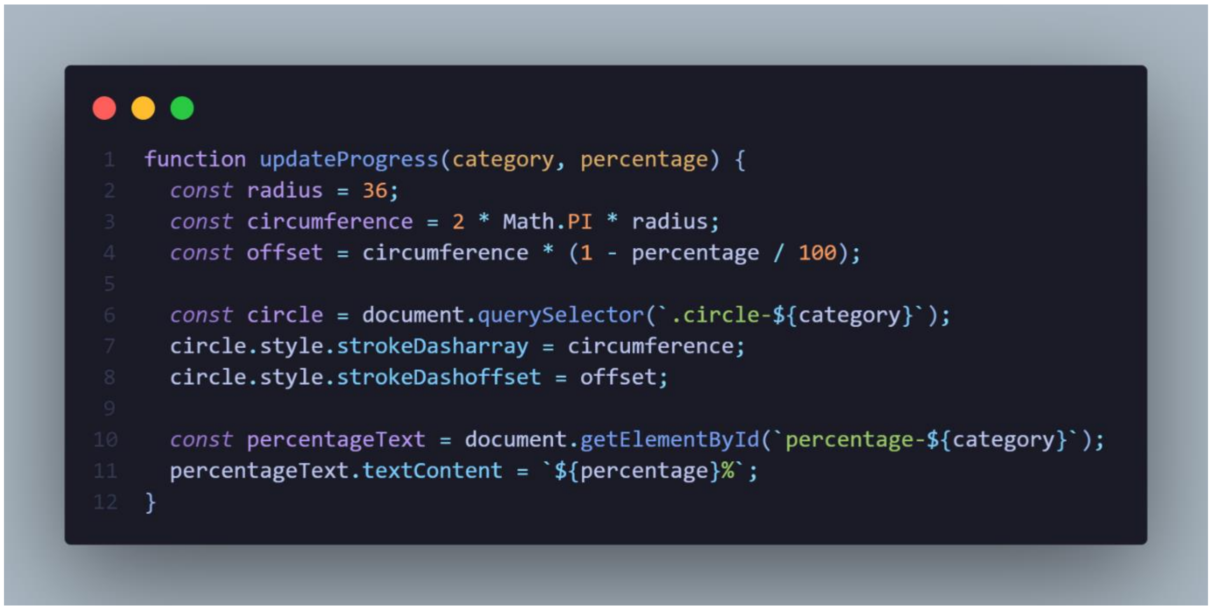


```php
function orderPercentage()
{
  global $connection;

  // Query to select the sum of order_amount for the last day
  $query_day = "SELECT SUM(order_amount) as total_day FROM orders WHERE created_at >= DATE_SUB(NOW(), INTERVAL 1 DAY)";
  $result_day = mysqli_query($connection, $query_day);

  // Fetch the result for the last day
  $row_day = mysqli_fetch_assoc($result_day);
  $total_day = $row_day['total_day'];

  // Query to select the sum of order_amount for all time
  $query_total = "SELECT SUM(order_amount) as total_amount FROM orders";
  $result_total = mysqli_query($connection, $query_total);

  // Fetch the total result
  $row_total = mysqli_fetch_assoc($result_total);
  $total_amount = $row_total['total_amount'];

  // Calculate the percentage of money made in the last day compared to all time
  $percentage_recent = 0;
  if ($total_amount > 0) {
    $percentage_recent = ($total_day / $total_amount) * 100;
  }

  $percentage_recent = round($percentage_recent, 2);

  return $percentage_recent;
}
```

Figure 128: Displaying how the percentage (not number) of the various totals was calculated.

Now that we have the from the code above. We now have to be able to display this on a circular progress bar on the website.

The code in figure 129 below displays how this is done. A function in javascrpt named 'updateProgress' was created to visually update the circular progress bar on the website. It takes two parameters; category (to identify the specific progress bar) and a percentage (to show how much of the progress bar should be filled). The function calculates the circumference of the circle and adjusts the stroke offset to reflect the given percentage, effectively updating the visual representation of the progress bar. Additionally, it updates the text displaying the current percentage, ensuring the displayed value matches the actual progress.



```
function updateProgress(category, percentage) {
  const radius = 36;
  const circumference = 2 * Math.PI * radius;
  const offset = circumference * (1 - percentage / 100);

  const circle = document.querySelector(`.circle-${category}`);
  circle.style.strokeDasharray = circumference;
  circle.style.strokeDashoffset = offset;

  const percentageText = document.getElementById(`percentage-${category}`);
  percentageText.textContent = `${percentage}%`;
}
```

Figure 129: Displaying the code to visually update the circular progress bar on the website.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 130 below, which also demonstrates how the added features improve the user experience.

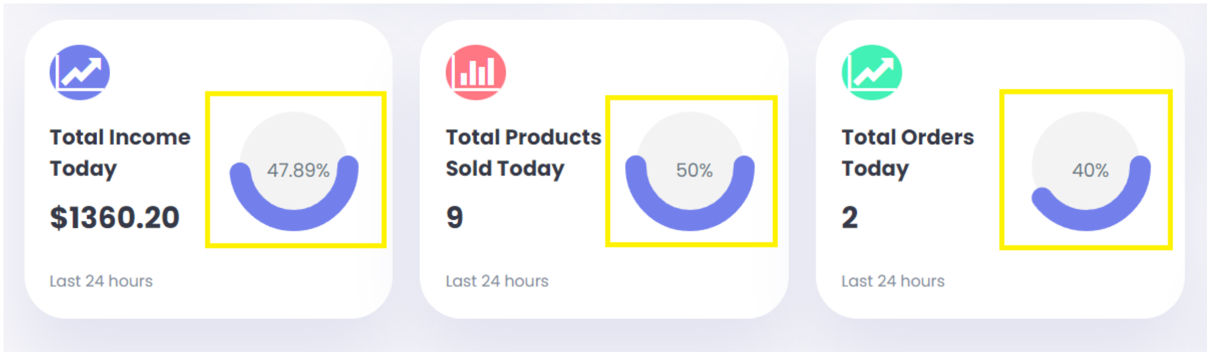
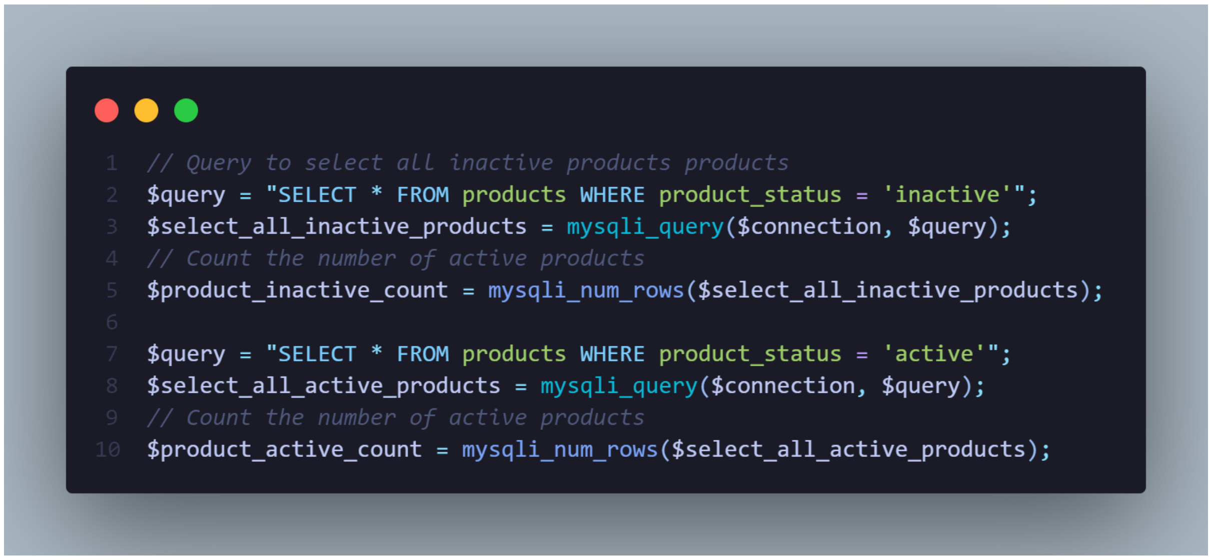


Figure 130: Displaying the outcome of this element/function on the website.

As seen in the above image, the percentage of the number of the daily income totals, the quantity of products sold, and total number of orders (highlighted in the yellow rectangle) made are display successfully.

Viewing the number of active products vs inactive products and the number of approved reviews vs unapproved reviews using a bar chart.

The code displayed in figure 131 below illustrates how the number of active and inactive products are retrieved from the database which must be done before we can display the figures on the bar chart. The PHP code provided below performs the database queries to count the active and inactive products from the 'products' table. It retrieves all the products where the ‘product_status’ is inactive and counts them, storing the result in a variable named ‘$product_inactive_count’. Subsequently ,this process was repeated for the ‘active’ products, saving the active count in a variable named ‘$product_active-count’. This entire process was also done for the approved and unapproved reviews.

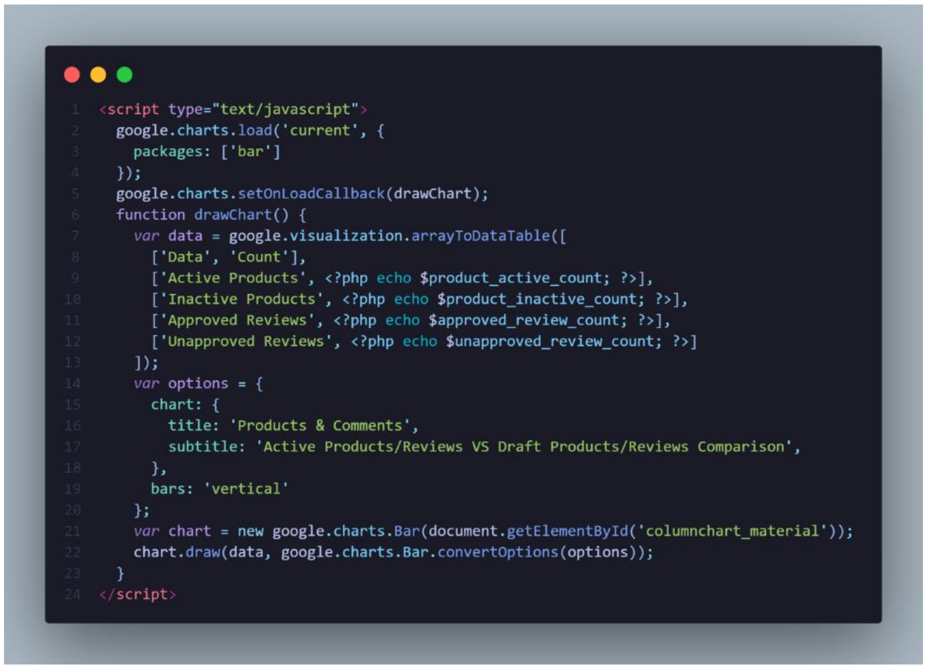


```php
// Query to select all inactive products products
$query = "SELECT * FROM products WHERE product_status = 'inactive'";
$select_all_inactive_products = mysqli_query($connection, $query);
// Count the number of active products
$product_inactive_count = mysqli_num_rows($select_all_inactive_products);

$query = "SELECT * FROM products WHERE product_status = 'active'";
$select_all_active_products = mysqli_query($connection, $query);
// Count the number of active products
$product_active_count = mysqli_num_rows($select_all_active_products);
```

Figure 131: Displaying how the number of active and inactive products are retrieved from the database.

Displaying the number of active and inactive products using the bar chart:

To display the number of active/inactive products and the number of approved/unapproved reviews, JavaScript was used to integrate the numbers (active products etc.) with Google Charts library to create a bar chart visualization. The chart used the variables created in figure 132 (‘$product_inactive_count’ etc.) to dynamically generate the chart data. The chart compares active and inactive products, as well as approved and unapproved reviews. The **draw Chart** function creates a Data Table, sets chart options (including title and type of bars), and finally, renders the chart in an HTML element with the ID 'columnchart_material'. This visualization effectively presents the database's data trends, enhancing the user's understanding of the website's activity and engagement. The code used to implement the features is displayed below.



```
<script type="text/javascript">
  google.charts.load('current', {
    packages: ['bar']
  });
  google.charts.setOnLoadCallback(drawChart);
  function drawChart() {
    var data = google.visualization.arrayToDataTable([
      ['Data', 'Count'],
      ['Active Products', <?php echo $product_active_count; ?>],
      ['Inactive Products', <?php echo $product_inactive_count; ?>],
      ['Approved Reviews', <?php echo $approved_review_count; ?>],
      ['Unapproved Reviews', <?php echo $unapproved_review_count; ?>]
    ]);
    var options = {
      chart: {
        title: 'Products & Comments',
        subtitle: 'Active Products/Reviews VS Draft Products/Reviews Comparison',
      },
      bars: 'vertical'
    };
    var chart = new google.charts.Bar(document.getElementById('columnchart_material'));
    chart.draw(data, google.charts.Bar.convertOptions(options));
  }
</script>
```

Figure 132: Displaying the JS code used to integrate the numbers (active products etc.)

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 133 below, which also demonstrates how the added features improve the user experience.

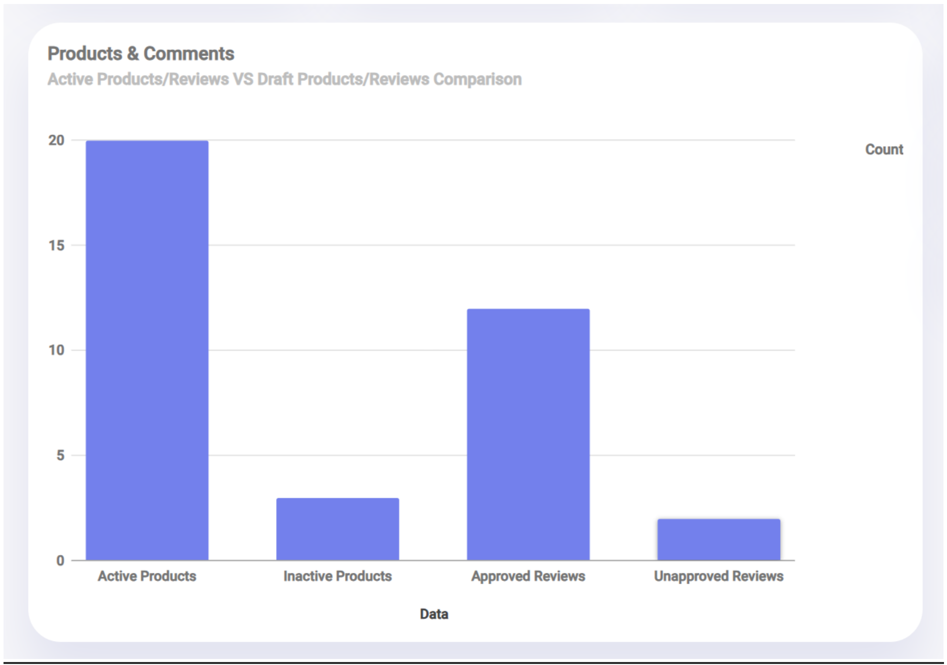


Figure 133: Displaying the outcome of this element/function on the website.

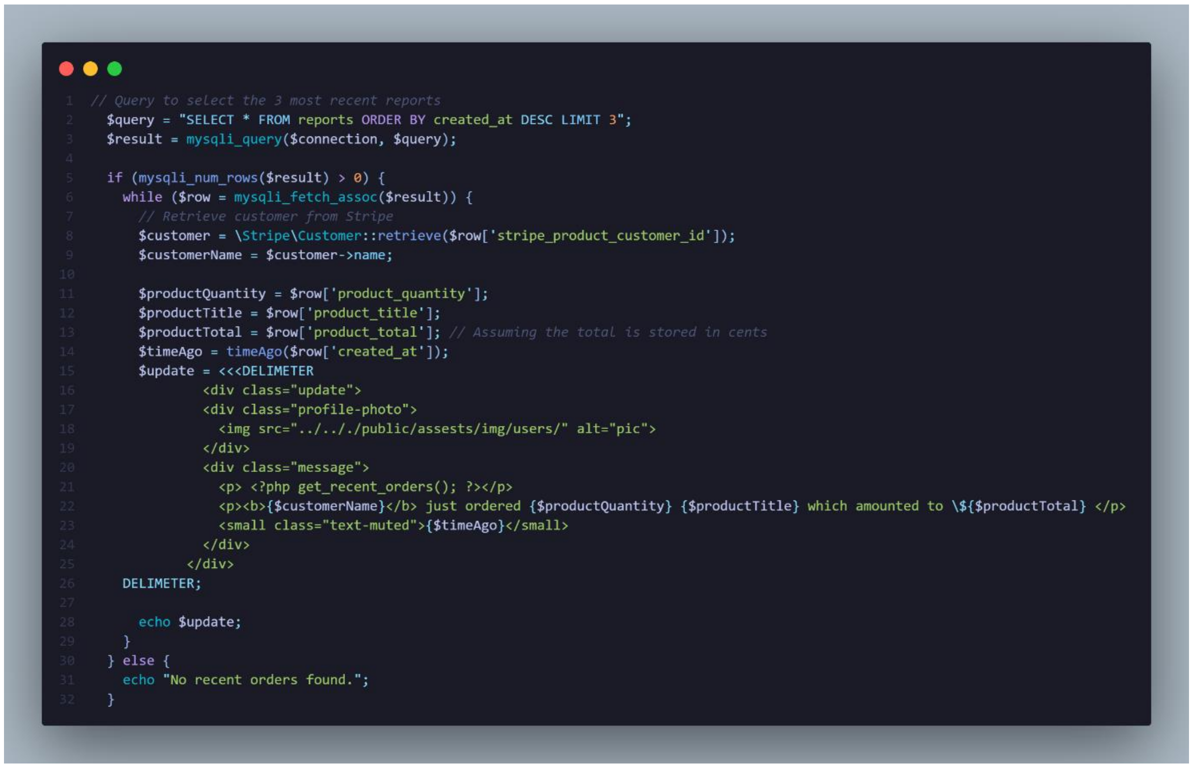
As seen in the above image, the number of active products vs inactive products and the number of approved reviews vs unapproved reviews using a bar chart are display successfully.

Viewing recent orders as messages.

The code displayed in figure 134 below illustrates how the features to let the administrator view recent orders was implemented. The PHP code requires the ‘reports’ table table to retrieve the three most current reports, sorted by their created_at date in descending order. It collects customer information from Stripe using the stripe_product_customer_id for each report and displays the customer's name, the quantity and title of the product ordered, the total amount of the order, and the how long ago the transaction was made.

The script the executes the SQL query with mysqli_query and iterates over the query results with mysqli_fetch_assoc. For each entry, it retrieves the customer's name from Stripe, prepares the product details, and uses a custom timeAgo function to calculate the time since the order. This data is then presented in a structured format, complete with a photo of the customer, a message outlining the order, and a timestamp indicating when the order was placed.

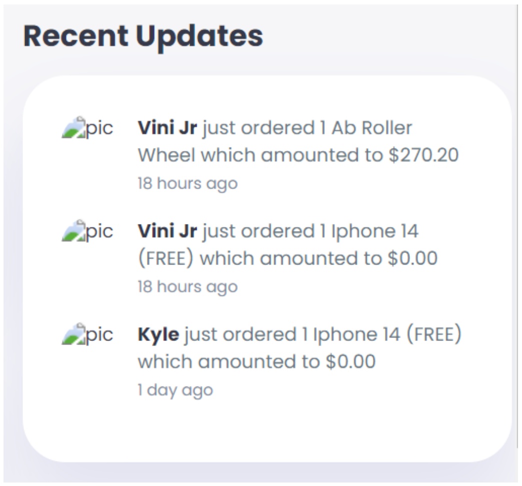
If no recent orders are identified, the message "No recent orders found." Is displayed. This is excellent for showcasing recent client activity.



```php
// Query to select the 3 most recent reports
  $query = "SELECT * FROM reports ORDER BY created_at DESC LIMIT 3";
  $result = mysqli_query($connection, $query);

  if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
      // Retrieve customer from Stripe
      $customer = \Stripe\Customer::retrieve($row['stripe_product_customer_id']);
      $customerName = $customer->name;

      $productQuantity = $row['product_quantity'];
      $productTitle = $row['product_title'];
      $productTotal = $row['product_total']; // Assuming the total is stored in cents
      $timeAgo = timeAgo($row['created_at']);
      $update = <<<DELIMETER
              <div class="update">
              <div class="profile-photo">
                <img src="../../../public/assests/img/users/" alt="pic">
              </div>
              <div class="message">
                <p> <?php get_recent_orders(); ?></p>
                <p><b>{$customerName}</b> just ordered {$productQuantity} {$productTitle} which amounted to \${$productTotal} </p>
                <small class="text-muted">{$timeAgo}</small>
              </div>
            </div>
    DELIMETER;

      echo $update;
    }
  } else {
    echo "No recent orders found.";
  }
```

Figure 134: Displaying the code used to implement the features to let the administrator view recent orders.

To examine the useful outcome of this element/function on the website. A screenshot of the feature is displayed in figure 135 below, which also demonstrates how the added features improve the user experience.


Recent Updates

pic **Vini Jr** just ordered 1 Ab Roller Wheel which amounted to $270.20
18 hours ago

pic **Vini Jr** just ordered 1 Iphone 14 (FREE) which amounted to $0.00
18 hours ago

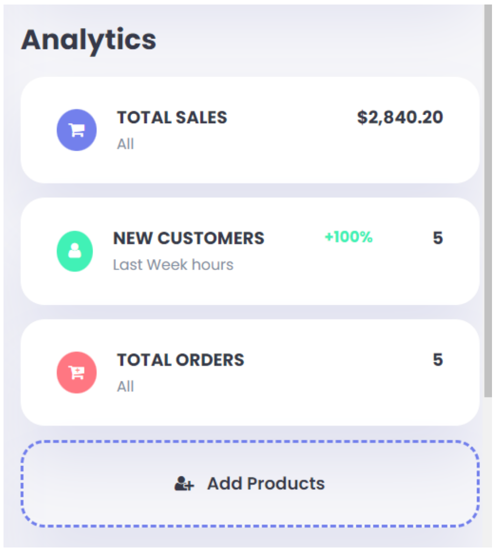
pic **Kyle** just ordered 1 Iphone 14 (FREE) which amounted to $0.00
1 day ago

Figure 135: Displaying the outcome of this element/function on the website.

As seen in the above image, the recent orders are displayed successfully.

Basic Analytics.

These basic analytics displayed below in figure 136 displays the total sales of the business since the beginning, the percentage of new customer that join the platform last week and the total amount of orders made since the beginning. Pieces of code used from the above sections was used to implement these elements.


Analytics

TOTAL SALES $2,840.20
All

NEW CUSTOMERS +100% 5
Last Week hours

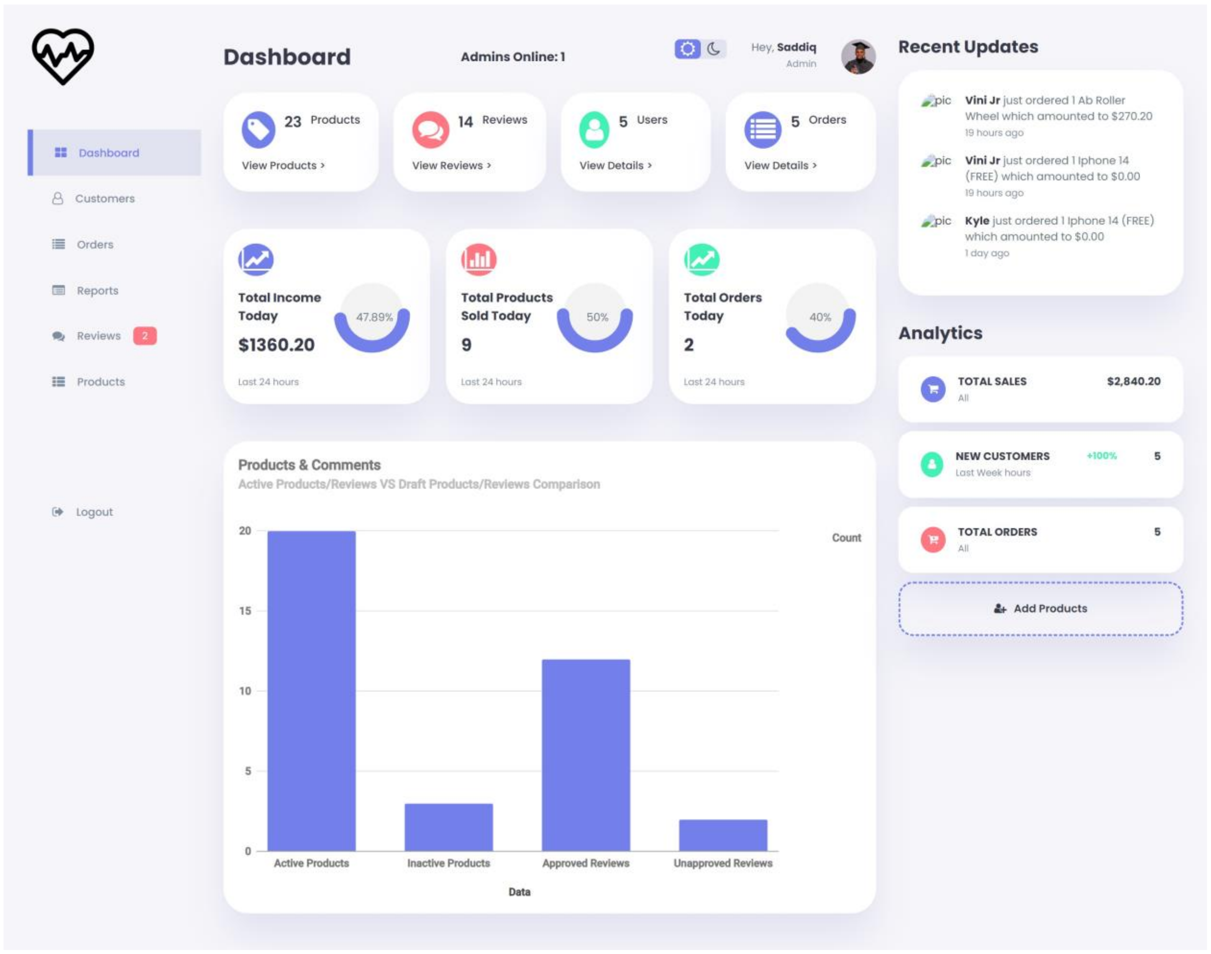
TOTAL ORDERS 5
All

Add Products

Figure 136: Displaying analytics.

Showcase of the Advanced Analytics and Data Visualization feature.

To showcase the implementation and effectiveness of the features discussed, we will now present the results as seen on the webpage. The following image provides a visual representation of how these features come to life in the actual user interface. This visualization not only demonstrates the practical application of the code but also highlights the user experience aspect of the website. The image below captures the essence of the implemented functionalities, offering a glimpse into the real-world outcome of our coding efforts. Let's take a look at the result (full web page):



Vini Jr just ordered 1 Ab Roller Wheel which amounted to $270.20
19 hours ago

Vini Jr just ordered 1 Iphone 14 (FREE) which amounted to $0.00
19 hours ago

Kyle just ordered 1 Iphone 14 (FREE) which amounted to $0.00
1 day ago

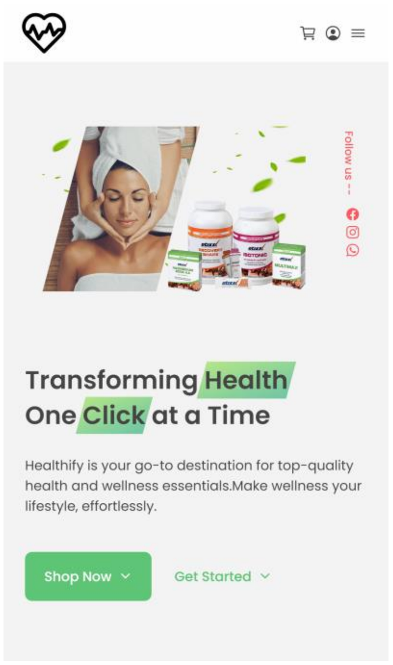
Figure 137: Displaying the full webpage of the section.

FR21 – Responsive Web Design

Media queries were utilized for the website's responsive web design. The website was able to efficiently adjust and react to various screen sizes and devices thanks to these media queries. In order to make sure that the website's layout, font sizes, graphics, and navigation components resized or repositioned itself effectively for each screen, the approach entailed developing precise CSS rules that would apply at different screen widths. This strategy-maintained functionality and aesthetics independent of the device being used to visit the website, ensuring a smooth and user-friendly experience across PCs, tablets, and smartphones.

BACK END | Let's take a closer look at the responsive design implementation (How the functionality was implemented):

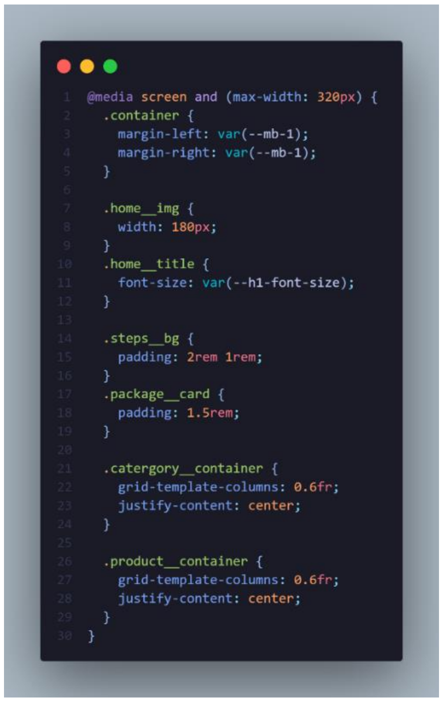
This diagram below offers an illustration of a media query used in responsive web design that targets display up to 320 pixels wide. This CSS snippet optimizes the viewing of different items on smaller displays. It resizes the home image to fit narrower displays, tweaks the container margins for better spacing, and changes the home title's font size for easier reading. For a neater layout, it also modifies padding in specific areas such as'steps__bg' and 'package__card'. The grid structure is modified for categories and goods to a single-column format with text centred, making the user experience consistent and easy to use on smaller screens, such as smartphones.



Transforming Health
One Click at a Time

Healthify is your go-to destination for top-quality health and wellness essentials.Make wellness your lifestyle, effortlessly.

Figure 139: Displaying home page view on a mobile phone.



```css
@media screen and (max-width: 320px) {
  .container {
    margin-left: var(--mb-1);
    margin-right: var(--mb-1);
  }

  .home__img {
    width: 180px;
  }
  .home__title {
    font-size: var(--h1-font-size);
  }

  .steps__bg {
    padding: 2rem 1rem;
  }
  .package__card {
    padding: 1.5rem;
  }

  .catergory__container {
    grid-template-columns: 0.6fr;
    justify-content: center;
  }

  .product__container {
    grid-template-columns: 0.6fr;
    justify-content: center;
  }
}
```

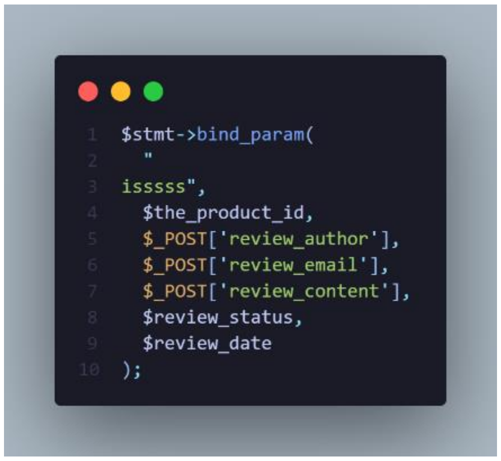
Figure 138: Displaying media query.

Security Consideration

In this section, we will look at the various security features built into the web application. These include secure data processing with prepared statements to prevent SQL injection, URL sanitization, and email sanitization to protect the database from harmful inputs. Session management strategies, such as session ID regeneration, are used to improve security after login. Session variables securely store user-related data, reducing data exposure risks. Passwords are securely hashed, and password composition is reviewed to ensure complexity. Access to certain functionality, such as the "Buy Now" button and admin parts, is also limited based on user login status and role, ensuring that sensitive elements of the website are only viewed by authenticated and authorized users.

Secure Data Handling in Submission Form

This script display below employs prepared statements to handle user input safely during submission processes. By assigning user-supplied data to SQL query parameters, it stops SQL injection.

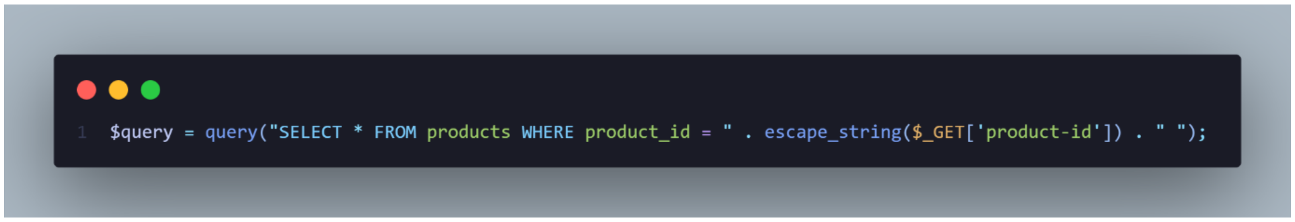


```
$stmt->bind_param(
  "
isssss",
  $the_product_id,
  $_POST['review_author'],
  $_POST['review_email'],
  $_POST['review_content'],
  $review_status,
  $review_date
);
```

Figure 140: Displaying the code that allows for Secure Data Handling in Submission Form.

Sanitization in Fetching Information

Information is safely fetched using this code. To stop SQL injection attacks, it sanitizes the ID retrieved from the URLs using escape_string.

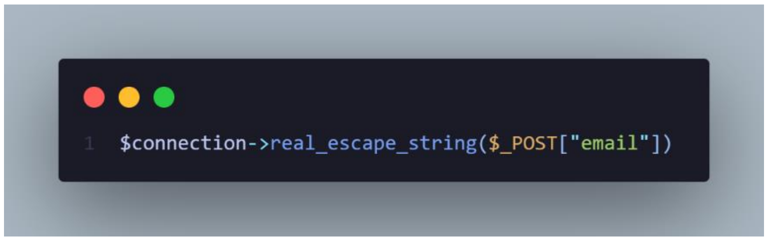


```
$query = query("SELECT * FROM products WHERE product_id = " . escape_string($_GET['product-id']) . " ");
```

Figure 141: Displaying the code that allows for Sanitization in Fetching Information.

Email Sanitization

This code sanitizes user-inputted email addresses to guard against SQL injection, a vital security precaution that protects the database from malicious queries.

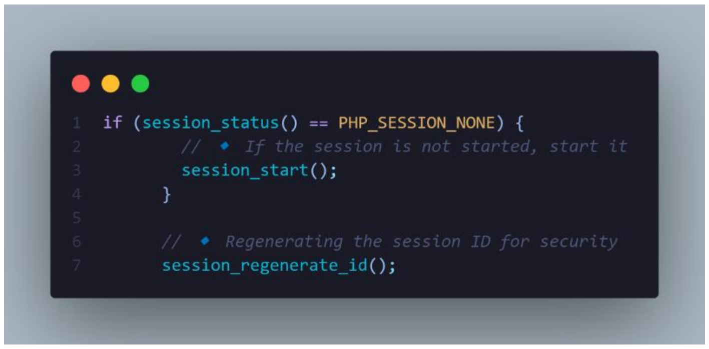


```
$connection->real_escape_string($_POST["email"])
```

Figure 142: Displaying the code that allows for Email Sanitization.

Session Management:

After a successful login, the code starts a new session and regenerates the session ID. This procedure improves security by guarding against hijacking of sessions.



```
if (session_status() == PHP_SESSION_NONE) {
        // 🔹 If the session is not started, start it
        session_start();
    }

    // 🔹 Regenerating the session ID for security
    session_regenerate_id();
```

Figure 143: Displaying the code that handles the session management.

Setting Session Variables:

Session variables are used to store user-related data following a successful login. This safe method lowers the possibility of sensitive data disclosure while maintaining user state across multiple pages.

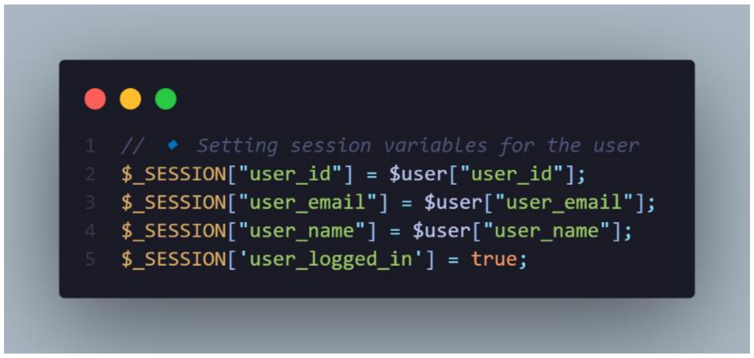

```
// 🔹 Setting session variables for the user
$_SESSION["user_id"] = $user["user_id"];
$_SESSION["user_email"] = $user["user_email"];
$_SESSION["user_name"] = $user["user_name"];
$_SESSION['user_logged_in'] = true;
```

Figure 144: Displaying the code responsible for Setting Session Variables securely.

Password Hashing

The password of the user is safely hashed and then stored in the database by this line. By safeguarding passwords from unwanted access and making them difficult to decipher, using password_hash with PASSWORD_DEFAULT guarantees that a robust hashing technique is used, greatly increasing the security of saved passwords. Keeping user credentials safe requires this important security precaution.

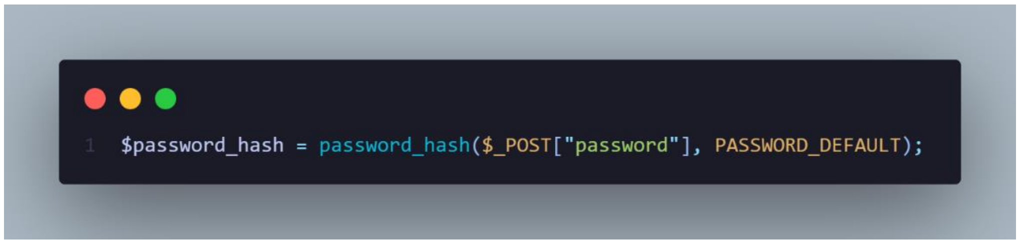
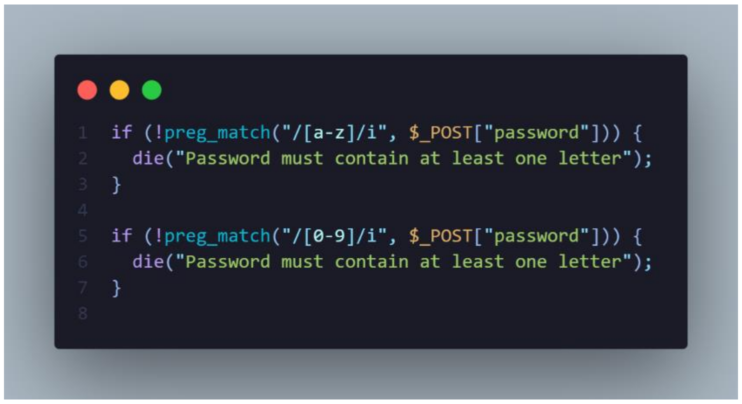

```
$password_hash = password_hash($_POST["password"], PASSWORD_DEFAULT);
```

Figure 145: Displaying the code responsible for password hashing.

Password Composition Check:

enforces a more secure password policy by verifying that the password consists of both letters and digits.

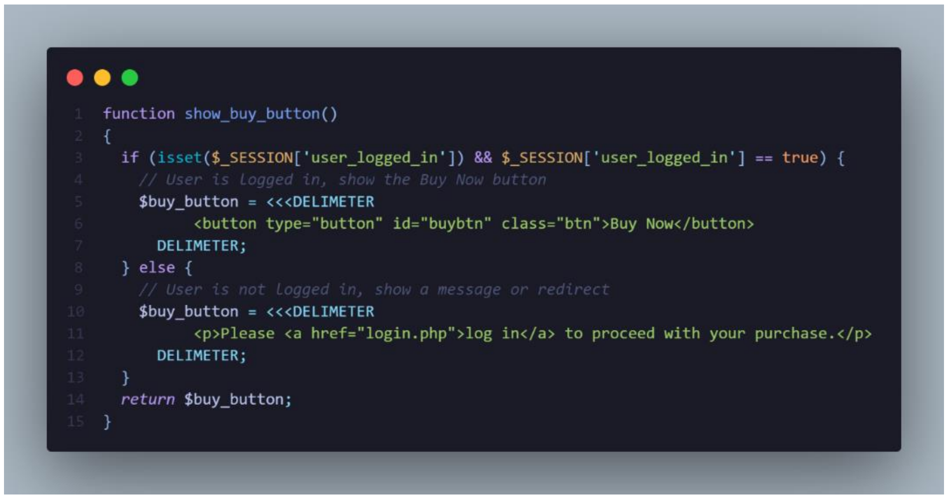


```php
if (!preg_match("/[a-z]/i", $_POST["password"])) {
  die("Password must contain at least one letter");
}

if (!preg_match("/[0-9]/i", $_POST["password"])) {
  die("Password must contain at least one letter");
}
```

Figure 146: Displaying the code responsible for the Password Composition Check.

Login Check for Purchase:

This section makes sure that the "Buy Now" button is only accessible to those who are logged in. It enhances security by limiting purchasing operations to authenticated users and uses the session variable $_SESSION['user_logged_in'] to check the user's login state.

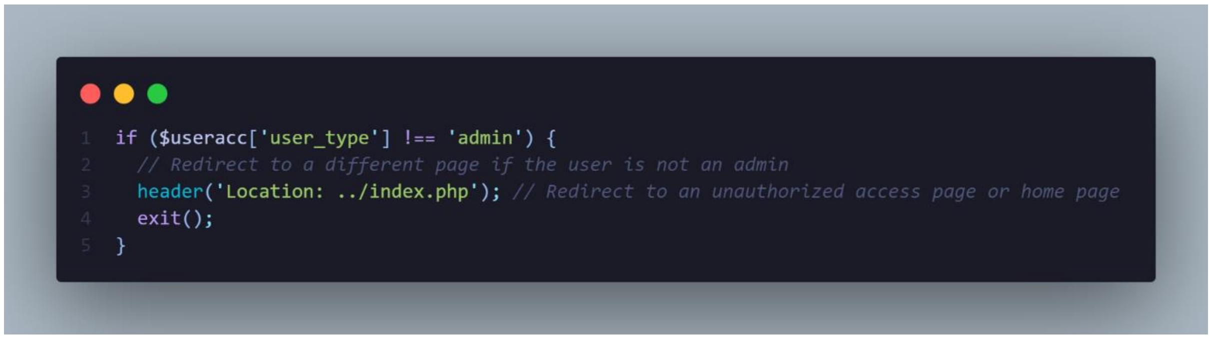


```php
function show_buy_button()
{
  if (isset($_SESSION['user_logged_in']) && $_SESSION['user_logged_in'] == true) {
    // User is logged in, show the Buy Now button
    $buy_button = <<<DELIMETER
          <button type="button" id="buybtn" class="btn">Buy Now</button>
      DELIMETER;
  } else {
    // User is not logged in, show a message or redirect
    $buy_button = <<<DELIMETER
          <p>Please <a href="login.php">log in</a> to proceed with your purchase.</p>
      DELIMETER;
  }
  return $buy_button;
}
```

Figure 147: Displaying the code responsible for the Login Check for Purchase.

Admin Role Check:

The script validates the user's session role. If the user is not an administrator, the page is redirected to a different one, the home page. This ensures that only users with admin privileges can access the website's admin section.



```php
if ($useracc['user_type'] !== 'admin') {
  // Redirect to a different page if the user is not an admin
  header('Location: ../index.php'); // Redirect to an unauthorized access page or home page
  exit();
}
```

Figure 148: Displaying the code responsible for the admin role check.

Error Messages

In this section, we will look at the various error messages that have been implemented in the web application, covering both the backend and the frontend. These messages are critical in providing feedback to users during processes such as registration, login, and data submission. They aid in the detection of problems with input validation, SQL query execution, and user authentication, thereby improving the user experience and preserving the application's integrity and security.

Username and Email Validation Error Messages:

These messages are presented during user registration if the username is empty or the email is not in an acceptable format, ensuring that all required user data is provided correctly.

Password Validation Error Messages:

Displayed at registration if the password does not fit the given criteria: minimum length, letters, and matching the confirmation password.

SQL Preparation Error:

When creating a SQL statement, this notice appears, suggesting a potential problem with the query syntax or database connection.

Duplicate Email Registration Error:

If the email address provided during user registration is already in use, this is triggered, preserving the uniqueness of user accounts.

General SQL Error:

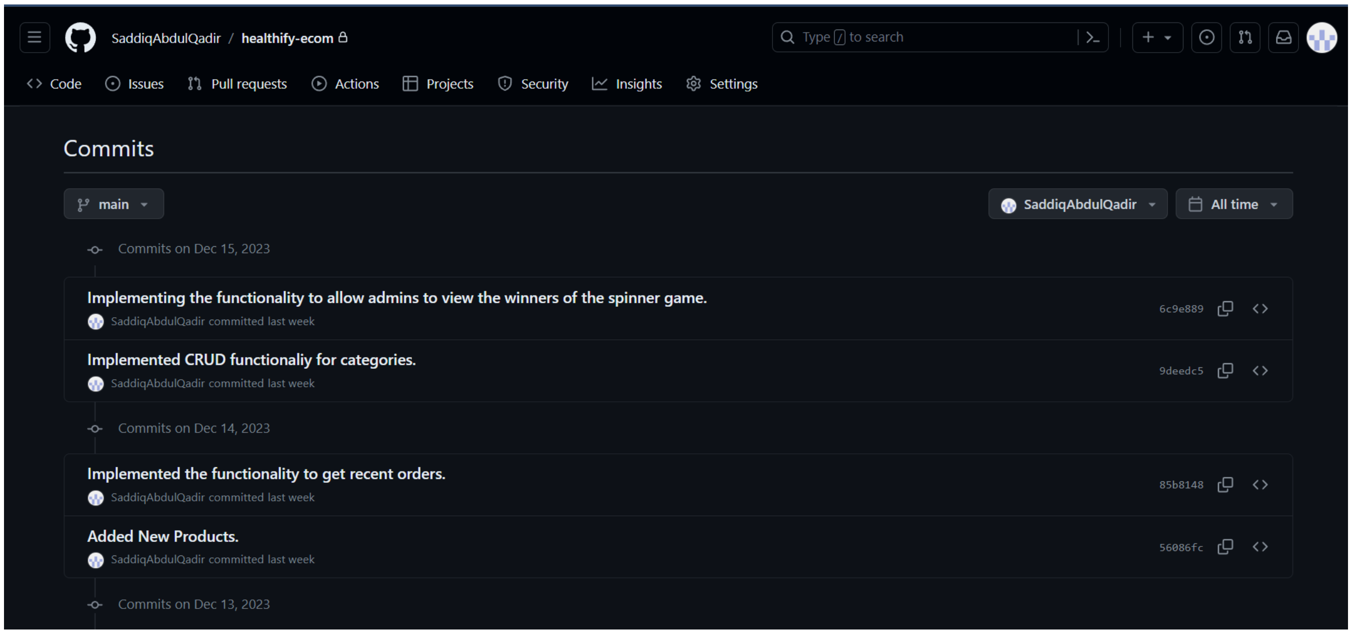
A generic error message for other SQL-related difficulties, containing both the error message and the code for troubleshooting.

Version Control

Tracking and managing changes to source code is useful in software development since it allows for change reversion and comparison to previous versions of the code.

Git was used to log changes for this project, with a git account linked in VScode. Code commits were made for significant modifications to the application or after an extended period of coding. The names of these commits were important because they could be reverted to before major modifications if necessary.

Figure 149 illustrates the log for the project's GITHUB Version Control:



Commits

Commits on Dec 15, 2023

Implementing the functionality to allow admins to view the winners of the spinner game.
SaddiqAbdulQadir committed last week 6c9e889

Implemented CRUD functionaliy for categories.
SaddiqAbdulQadir committed last week 9deedc5

Commits on Dec 14, 2023

Implemented the functionality to get recent orders.
SaddiqAbdulQadir committed last week 85b8148

Added New Products.
SaddiqAbdulQadir committed last week 56086fc

Commits on Dec 13, 2023

Figure 149: Displaying the GitHub log.

Features Not Implemented

While there is always opportunity for improvement and future development, it is great to say that the system has successfully implemented all of the features that were initially planned for implementation.

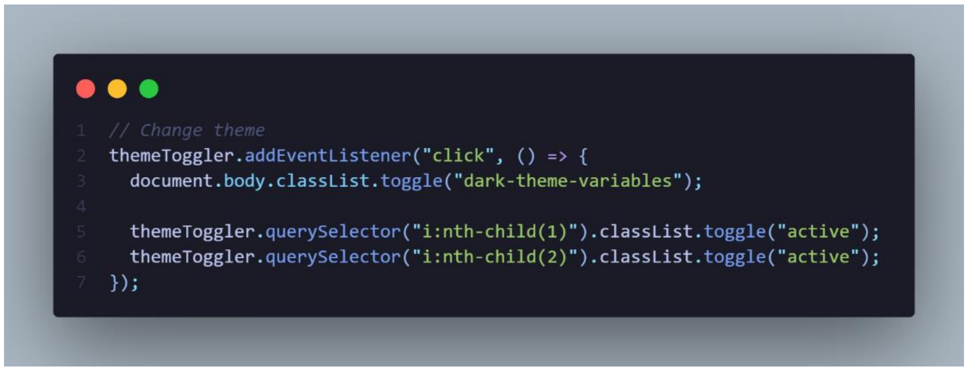
Additional Features Implemented

Due to the completion of all essential functionalities, two additional features were implemented: a dark mode for the admin dashboard and a function that allows admins to view how many admins are online.

Dark theme

When you click the dark theme button on the admin dashboard, the visual appearance changes from light to dark. This function provides admins with a more pleasant and less taxing experience, especially in low-light circumstances, allowing them to utilize the dashboard for longer periods of time.

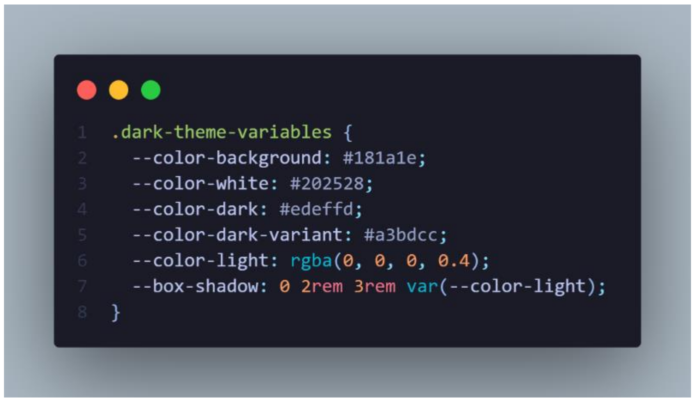
The JavaScript code below monitors the theme toggle button for a click event. When you click the button, it toggles a class (dark-theme), which causes the relevant CSS styles for the selected theme to be applied.



```javascript
// Change theme
themeToggler.addEventListener("click", () => {
  document.body.classList.toggle("dark-theme-variables");

  themeToggler.querySelector("i:nth-child(1)").classList.toggle("active");
  themeToggler.querySelector("i:nth-child(2)").classList.toggle("active");
});
```

Figure 150: Displaying the JS code for the toggle button.

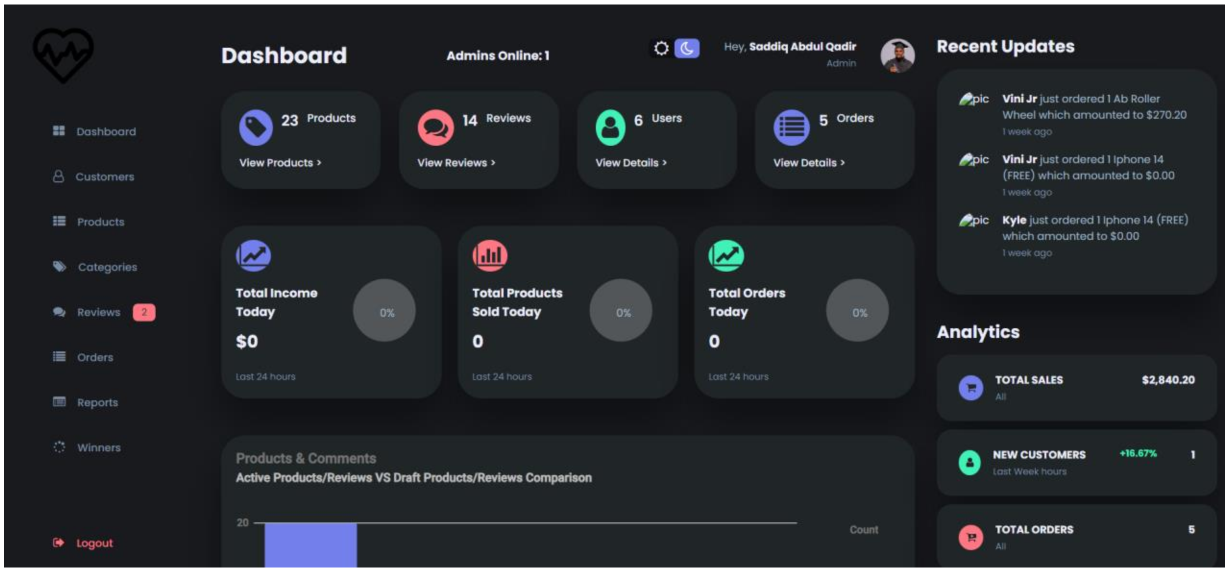
The CSS/SCSS variables displayed below possess styling properties for both the light and dark themes, such as background colors, text colors, and box shadows.



```
.dark-theme-variables {
  --color-background: #181a1e;
  --color-white: #202528;
  --color-dark: #edeffd;
  --color-dark-variant: #a3bdcc;
  --color-light: rgba(0, 0, 0, 0.4);
  --box-shadow: 0 2rem 3rem var(--color-light);
}
```

Figure 151: Displaying the dark theme variables.

Displayed below is an illustration of the dark theme mode on the admin dashboard.



pic Vini Jr just ordered 1 Ab Roller Wheel which amounted to $270.20
1 week ago

pic Vini Jr just ordered 1 Iphone 14 (FREE) which amounted to $0.00
1 week ago

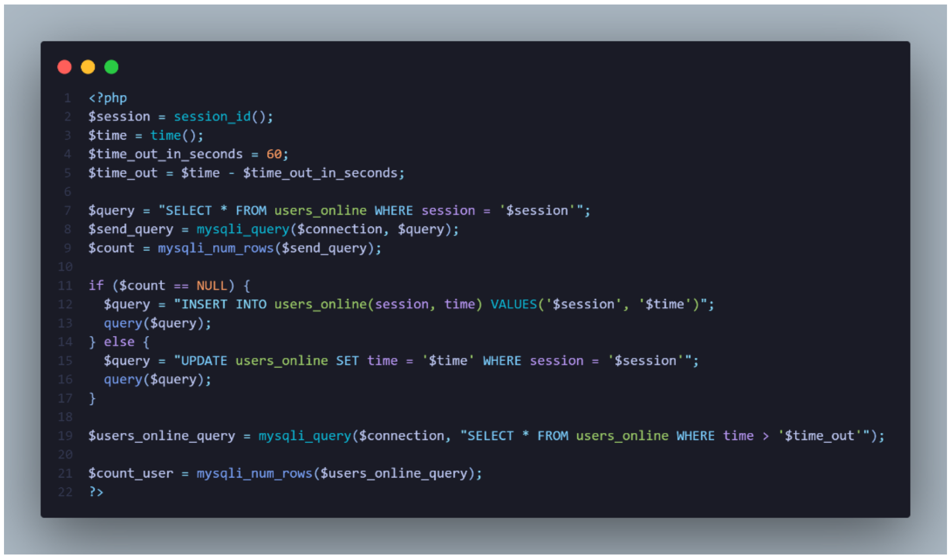
pic Kyle just ordered 1 Iphone 14 (FREE) which amounted to $0.00
1 week ago

Figure 152: Displaying the dark theme mode on the admin page.

Users Online

The users' online function allows administrators to view how many other administrators are online provides real-time visibility into the platform's active administrative users. It operates by monitoring the presence or status of administrators when they log in and out. This feature encourages collaboration and assists administrators in determining the availability of their colleagues, easing contact and coordination when necessary.

This PHP code below keeps track of user online status by storing session IDs and timestamps in a database table. If a user is already listed as online, it updates their timestamp; otherwise, it creates a new record. It estimates how many admins are now online by searching the database for users whose timestamps fall within a specific time-out period, offering real-time insight on active users.



```php
<?php
$session = session_id();
$time = time();
$time_out_in_seconds = 60;
$time_out = $time - $time_out_in_seconds;

$query = "SELECT * FROM users_online WHERE session = '$session'";
$send_query = mysqli_query($connection, $query);
$count = mysqli_num_rows($send_query);

if ($count == NULL) {
  $query = "INSERT INTO users_online(session, time) VALUES('$session', '$time')";
  query($query);
} else {
  $query = "UPDATE users_online SET time = '$time' WHERE session = '$session'";
  query($query);
}

$users_online_query = mysqli_query($connection, "SELECT * FROM users_online WHERE time > '$time_out'");

$count_user = mysqli_num_rows($users_online_query);
?>
```

Figure 153: Displaying the code below that keeps track of user online status.

The image below displays the user's online functionality (how many users are online).

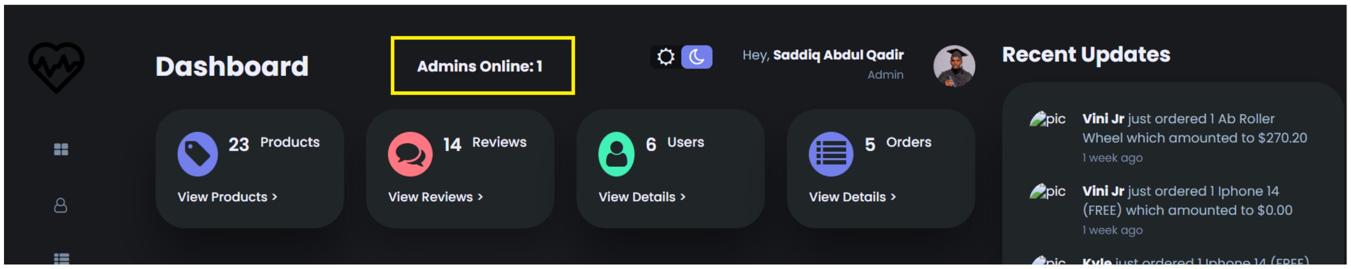


Figure 154: Displaying the result of the user's online functionality.

Testing

Introduction

Testing is an essential stage of software development that provides the foundation of a reliable and secure system. The testing process is rigorous and verifies that the software works as intended, meeting requirements and continues to perform well in the face of unanticipated challenges.

Fundamentally, testing is an organized investigation that examines the functionality of the software with a focus on potential vulnerabilities as well as expected performance. It includes both the detection of unexpected behaviors and the confirmation of desired functionality. In particular, testing is most important when it reveals defects and serves as a driving force for improvement.

This chapter delves into the project's testing phase and examines the specifics of both functional and non-functional testing. The procedure is guided by a well-organized test strategy that makes sure every facet of the system's behavior is examined.

Each test's results are properly documented, which is an important component of our testing process. Any discovered flaws are documented, giving a road map for future bug fixes. This dedication to extensive documentation not only helps the development process, but it also speeds up the transition from testing to deployment.

Test Plan

The testing method for the application includes a comprehensive strategy for evaluating both functional and non-functional features. Functional criteria are inspected using a combination of unit and integration tests to ensure that each component functions as expected. Unit tests, which are methodically developed, evaluate the most basic system components, managing a variety of inputs and outputs, including expected, boundary, and negative outcomes. Meanwhile, integration tests investigate the harmonious interplay between webpages and the database, assuring flawless functionality across all components.

Each test in this strategy is methodically planned, delivering critical information such as test data, expected and actual results, and a clear outcome (success or failure). All tests are backed up by real evidence, and any failed tests are scrupulously documented in the Defects Log. Following testing, the emphasis changes to addressing and resolving any problems noted in the Defects Log, ensuring that the application satisfies the highest reliability and performance standards.

Last but not least, the objective of this testing plan is to identify and fix any sections of the application that may require refinement or modification, ensuring a robust and dependable software system that precisely corresponds with our defined goals.

Functionality Testing

Tables 23 and 24 illustrate the testing of function requirements 1 and 2 as well as the test data and results. Appendix D will detail the results of the remaining function requirement tests.

Table 23: Displaying the test for the registration function.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-R-1	Register user with no information inserted into the fields.	A message indication to fill out fields.	As expected	Success
FR-R-2	Register using wrong email format. Email: Jadangmail.com	Message indicating please include an '@' in the email address.	As expected	Success
FR-R-3	Register using passwords that don't match. Password: Password1# Confirm Password: Password1123	Message indication passwords doesn't match.	As expected	Success
FR-R-4	Register with a password that does not meet the password criteria. Password: Password	Message indicating password must contain minimum eight characters. At least one letter and one number.	As expected	Success
FR-R-5	Register using an email address that is already used by another user in the database.	Message indicating that email is already taken	As expected	Success
FR-R-6	Register using valid data. Email: Jadan@gmail.com Password: Password123# Password: Password123#	Message indication successful sign up with the option to login or go to the home page.	As expected	Success

Table 24: Displaying the test for the login function.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-L-1	Login with password and no email	Message indicating invalid login.	As expected	Success
FR-L-2	Login with email and no password.	Message indicating invalid login and for the email to remain in the field.	As expected	Success
FR-L-3	Login with email and incorrect password.	Message indicating invalid login and for the email to remain in the field whiles the password field is cleared.	As expected	Success
FR-L-4	Login with no data.	Message indicating invalid login.	As expected	Success
FR-L-5	Login using valid data. Email: Saddiqa.qadir@gmail.com Password: Password1#	Redirect user to the home page indicating login was successful and display drop down with user email and option to manage his/her account.	As expected	Success

Non-Functionality Testing

For each non-functional need that could be examined, a set of tests was devised, and the findings were documented, exactly as they were for the functional requirements. Inefficiencies in tests were reported in the defects log. Table 25 illustrates the results of the nonfunctional requirement testing.

Table 25: Displaying illustrates the results of the nonfunctional requirement testing.

ID	Short Description	Expected Outcome	Actual Outcome	Success/Fail
NFR1	Consistency across the application in terms of design, interactions, and workflows	All pages and interactions exhibit a consistent design and user experience	Design and interactions are consistent throughout the application	Pass
NFR2	Codebase, architecture, and technologies allow for easy maintenance and updates	Codebase is well-structured and documented, enabling easy updates and maintenance	Codebase and architecture support easy maintenance and updates	Pass
NFR3	Application functions correctly across various	Application works without issues on Chrome, Firefox,	Application functions correctly on all tested browsers	Pass

	browsers (Chrome, Firefox, Safari, etc.)	Safari, and other major browsers		
NFR4	Application is intuitive and user-friendly with clear navigation	Users can easily navigate and use features	Users find the application intuitive and user-friendly	Pass
NFR5	Database is secure from unauthorized access, injection attacks, and data leaks	Database access is restricted, and no data breaches or injection attacks occur	Database security measures effectively prevent unauthorized access and attacks	NA – Application on localhost
NFR6	Application implements security measures like HTTPS, data encryption, and secure coding	Application uses HTTPS, encrypts sensitive data, and follows secure coding practices	Application is secure against common cyber threats	Pass
NFR7	Application can handle growth in users, data volume, and transactions	Application scales gracefully without performance degradation	Application scales effectively to accommodate increased load	Pass
NFR8	Application is optimized for fast load times and smooth interactions	Application loads quickly, and interactions are responsive	Application demonstrates excellent performance with fast load times	Pass
NFR9	Application is available at all times, minimizing downtime	Application is accessible 24/7 with minimal downtime	Application remains highly available with no significant downtime	NA – Application on localhost

Defects Log

The Defects Log is a documentation tool used to track and report any faults, defects, or barriers discovered during project artifact testing. It acts as a repository for areas that need to be improved while also highlighting any components that failed to meet the requirements during previous inspections.

Defects are classified according to their severity, with the severity levels as follows:

- Severity 0: No significant issue; not a problem.
- Severity 1: Cosmetic issue; requires attention only if time allows.
- Severity 2: Minor problem; low priority for resolution.
- Severity 3: Major problem; requires immediate and high-priority resolution.

Table 26 displays the details of the Defects Log, including the severity scores and reasons for each recorded issue.

Table 26: Displaying the defects log.

Id	Type	Requirement	Severity	Comment
1	Advanced Functional Requirement	FR-19: Gamification Feature	1	If a user has already played the spinner game for the month, they should be told that they have already played for the month and should return the following month. This functionality, however, has a defect, and no message or notification is provided to the user.

Analysis of Test Results

All of the tests were successful, according to the evaluation of the test findings. Due to time constraints, the one defect in the spinning game that was found was not fixed; nonetheless it was minor and did not represent a risk to the program.

Evaluation

Introduction

Software evaluation is an important step that goes beyond simply determining whether or not the application works, delving into its overall performance in real-world settings. The implementation phase focuses on developing the application's components, whereas the evaluation phase evaluates its efficiency and functionality. This phase aims to identify the application's strengths, areas for improvement, and alignment with intended standards by using structured methodologies and gathering feedback from a variety of parties. The project supervisor and the university coordinator will lead the evaluation in this context, ensuring a thorough review.

Method of Evaluation

The project's time limits and the particular academic procedures in place are the main reasons a heuristic assessment method has been selected to evaluate this application. The testing that we can perform on subjects who have completed an ethics clearance process is limited by these protocols. It is noteworthy, yet, that this strategy is not without its drawbacks. One significant disadvantage is that it doesn't prioritize measuring consumer satisfaction due to its inherent lack of subjectivity.

The Heuristics

This evaluation will determine whether the heuristic was implemented completely, partially, or not at all for every item on the checklist. This led to the following rating being used to represent the result:

- 0 indicates the heuristic was not implemented.
- 1 indicates the heuristic was partially implemented.
- 2 indicates the heuristic was fully implemented.

Table 27: Displaying the heuristics evaluation.

1. Visibility of System Status			
ID	**Checklist**	**Rating**	**Comments**
1.1	Title/header for screen contents	2	Effectively labelled titles/headers.
1.2	Consistency in instructions/messages	2	Instructions/messages consistently placed.
1.3	Consistent icon design	2	Icons adhere to a consistent design scheme.
1.4	Use of navigational aids	2	Context labels and navigation aids used.
1.5	Visual feedback for options	2	Visual feedback for selected options.
2. Match Between System and the Real World			
ID	**Checklist**	**Rating**	**Comments**
2.1	Concrete and familiar icons	2	Icons are concrete and familiar.
2.2	Corresponding colours	2	Colours correspond to common expectations.
2.3	Automatic commas for large numbers	0	Complex programming
2.4	User-friendly command language	2	User-friendly language; avoids jargon.
2.5	Related fields placement	2	Related fields placed together.
3. User Control and Freedom			
ID	**Checklist**	**Rating**	**Comments**
3.1	System waiting for user signal	2	System waits for user signal upon task completion.
3.2	Prompt for confirmation of commands	2	Users prompted for confirmation of drastic commands.
3.3	Selection in long menu lists	2	
3.4	Navigation among multipage screens	2	Users can navigate among all pages.
4. Consistency and Standards			
ID	**Checklist**	**Rating**	**Comments**
4.1	Avoidance of all uppercase letters	2	No heavy use of all uppercase letters.
4.2	Alignment of integers and decimals	2	Right-justified integers and decimal-aligned numbers.
4.3	Appropriate labelling of icons	2	Icons are labelled appropriately.
4.4	Consistent page titles	2	All pages have consistent titles.
4.5	Avoidance of saturated blues	2	Saturated blues avoided for text/symbols.
4.6	Limited colour scheme	2	Colour scheme adheres to a limited palette.
5. Help Users Recognize, Diagnose, and Recover from Errors			
ID	**Checklist**	**Rating**	**Comments**
5.1	Constructive error prompts	2	Error prompts are constructive.
5.2	Brief and unambiguous error prompts	2	Error prompts are brief and unambiguous.
5.3	Grammatically correct error messages	2	Error messages are grammatically correct.
5.4	Avoidance of exclamation points	2	No use of exclamation points in error messages.
5.5	Suggestion of error causes	1	Error messages suggest causes of the problem.
5.6	Action to correct errors suggested	2	Error messages indicate corrective actions.
6. Error Prevention			
ID	**Checklist**	**Rating**	**Comments**

6.1	Error prevention when possible	2	System prevents errors whenever possible.
6.2	Warning for potentially serious errors	2	Users warned about potentially serious errors.
	7. Recognition Rather Than Recall		
ID	**Checklist**	**Rating**	**Comments**
7.1	Placement of prompts, cues, messages	2	Prompts, cues, and messages well-placed.
7.2	"Breathing space" around text areas	2	Text areas have sufficient space around them.
7.3	Use of white space for symmetry	2	White space used for symmetry and guidance.
7.4	Consistency in color coding	2	Color coding consistent throughout.
7.5	Contrast in image-background colors	2	Good contrast between image and background.
7.6	Label proximity to fields	2	Field labels close to fields but separated.
	8. Flexibility and Minimalist Design		
ID	**Checklist**	**Rating**	**Comments**
8.1	Clear titles for data entry screens	2	Data entry screens have clear, concise titles.
8.2	Brief, familiar, and descriptive labels	2	Field labels brief, familiar, and descriptive.
8.3	Distinct icons from background	2	Icons stand out from their backgrounds.
8.4	Brief menu titles	2	Menu titles brief yet communicative.
	9. Aesthetic and Minimalist Design		
ID	**Checklist**	**Rating**	**Comments**
9.1	Display of essential information only	2	Only essential information displayed.
9.2	Visually and conceptually distinct icons	2	Icons visually and conceptually distinct.
9.3	Grammatically correct error messages	2	Error messages are grammatically correct.
9.4	Suggestion of error causes in messages	2	Error messages suggest causes of the problem.
	10. Help and Documentation		
ID	**Checklist**	**Rating**	**Comments**
10.1	Distinct online instructions	2	Online instructions visually distinct.
10.2	Easy-to-find navigation	2	Information easy to find in navigation.
10.3	Well-designed visual layout	2	Visual layout well-designed.
10.4	Relevance of information	2	Information relevant to the user.
10.5	Goal-oriented information	2	Information answers "What can I do with this program?"

Summary of Findings

The Heuristic Report gathered at the conclusion of the heuristic evaluation is presented below. There were forty-seven items on the list, one of which was not implemented and another which was partially implemented. Table 28 contains the Heuristic Report.

Table 28: Displaying the heuristic report.

ID	Checklist	Severity	Comments
2.3	Automatic commas for large numbers	1	Complex programming
5.5	Suggestion of error causes	1	Error messages suggest causes of the problem.

Analysis of Results

The Heuristic Report findings were produced and 45 of the 47 checklist items were executed in full. As a result of a lack of time and experience, the two deflects were not corrected, although the fact that they were minor and did not risk the application's usability.

outside resources provided reassurance, but the pressure to complete the project within the limited timeframe was overwhelming.

The development of the artifact was a difficult yet an educational experience. The addition of advanced features and adherence to design principles improved the functionality and user experience of the application. However, technical issues, as well as the possibility of missing critical components due to inexperience and time constraints, highlighted areas for improvement.

The change in coding approach and the addition of advanced features demonstrated growth in my programming abilities. However, technical difficulties and reliance on external resources and lecturer assistance highlighted the need for a more thorough understanding of the tools and technologies used. The application's inspiration came from well-known e-commerce applications, which may have contributed to the pressure felt.

Despite the difficulties encountered and the possibility of missing essential elements, the development of the artifact was a valuable learning experience. Given my lack of programming experience, the application's strengths and the knowledge gained during the process are notable accomplishments.

In future projects, I intend to deepen my understanding of programming tools and technologies in order to prevent technical issues and reduce reliance on third-party resources. I will also set aside enough time for testing and quality assurance to ensure that all critical components are in place. This experience will lay the groundwork for ongoing learning and improvement in action research projects.

References

A.Altameem, E., 2015. Impact of Agile Methodology on Software Development. *Canadian Center of Science and Education,* 8(2), pp. 1-5.

Ahmad, N. A. et al., 2018. Review of chatbots design techniques. *International Journal of Computer Applications,* 181(8), pp. 7-10.

Alhijawi, B. & Kilani, Y., 2020. A collaborative filtering recommender system using genetic algorithm. *Information Processing & Management,* 57(6).

Amazon Alexa, 2023. *https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/.* [Online]
Available at: https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/
[Accessed 03 08 2023].

Apple, 2023. *Ikea Place.* [Online]
Available at: https://apps.apple.com/us/app/ikea-place/id1279244498%5C
[Accessed 20 08 2023].

Augello, A., Gentile, M. & Dignum, F., 2018. An overview of open-source chatbots social skills. In: *Internet Science: INSCI 2017 International Workshops, IFIN, DATA ECONOMY, DSI, and CONVERSATIONS, Thessaloniki, Greece, November 22, 2017, Revised Selected Papers 4.* s.l.:s.n., pp. 236-248.

Bakker, A. B. & Demerouti, E., 2007. The Job Demands-Resources model: state of the art. *Journal of Managerial Psychology,* 22(3).

Bansal, H. & Khan, R., 2018. A Review Paper on Human Computer Interaction. *International Journals of Advanced Research in Computer Science and Software Engineering,* Issue 4.

Bass, L., Clements, P. & Kazman, R., 2012. *Software Architecture in Practice.* 3rd ed. s.l.:Addison-Wesley.

Beck, K. L. et al., 2013. *Manifesto for Agile Software Development.* s.l., s.n.

Borges, M., Hoppen, N. & Luce, F. B., 2009. Information technology impact on market orientation in e-business. *Journal of Business Research,* 62(9), pp. 883-890.

Bridge, D. G., Goker, M. H., McGinty, L. & Smyth, B., 2005. Case-based recommender systems. *The Knowledge Engineering Review,* Volume 20, pp. 315-320.

Cegarra-Navarro, J. G., Jiménez, D. J. & Martínez-Conesa, E. Á., 2007. Implementing e-business through organizational learning: An empirical investigation in SMEs. *International Journal of Information Management,* 27(3), pp. 173-186.

Chacon, S. & Straub, B., 2014. *Pro Git.* s.l.:Apress Berkeley, CA.

Chen, J., 2020. *Research on The Application of Blockchain Technology in Supply Chain Financial Business.* s.l.:s.n.

Cohn, M., 2009. *Succeeding with Agile: Software Development Using Scrum.* s.l.:Addison-Wesley Professional.

Colace, F. et al., 2018. Chatbot for e-learning: A case of study. *International Journal of Mechanical Engineering and Robotics Research,* 7(5), pp. 528-533.

Colby, K. M., Weber, S. & Hilf, F. D., 1971. *Artificial intelligence,* 2(1), pp. 1-25.

Conforto, E. C. et al., 2016. The agility construct on project management theory. *International Journal of Project Management,* Volume 34, pp. 660-674.

Cortana Home Assistant – Microsoft., 2023. *Your personal productivity assistant in Microsoft 365.* [Online]
Available at: https://www.microsoft.com/en-us/cortana
[Accessed 01 08 2023].

Covey, S. R., 2004. *The 7 Habits of Highly Effective People.* revised ed. s.l.:Simon and Schuster, 2004.

Dabbish, L. A., Stuart, C., Tsay, J. & Herbsleb, J. D., 2012. Social coding in GitHub: transparency and collaboration in an open software repository. *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work.*

DataReportal, 2021. *Digital 2021: Global Overview Report.* [Online]
Available at: https://datareportal.com/reports/digital-2021-global-overview-report
[Accessed 05 08 2023].

Deng, J., Guo, J. & Wang, Y., 2019. A Novel K-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering. *Knowledge-Based Systems,* Volume 175, pp. 96-106.

Deterding, S., Dixon, D., Khaled, R. & Nacke, L., 2011. From game design elements to gamefulness: defining "gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments,* pp. 9-15.

Dora, S. K. & Dubey, P., 2013. Software development Life Cycle (SDLC) Analytical Comparison and Survey on Traditional and Agile Methodology. *National Monthly Refereed Journal of Research in Science & Technology,* 2(8), pp. 22-23.

Dybå, T. & Dingsøyr, T., 2008. Empirical studies of agile software development: A systematic review. *Information and Software Technology,* 50(9-10), pp. 833-859.

Edelman, D. C. & Abraham, M., 2023. *Customer Experience in the Age of AI.*

eMarketer, 2023. [Online]
Available at: https://www.insiderintelligence.com/coverage/emarketer/
[Accessed 20 07 2023].

Feizollahi, S., Shirmohammadi, A., Kahreh, Z. S. & Kaherh, M. S., 2014. Investigation the Effect of Internet Technology on Performance of Services Organizations with e-commerce Orientations. *Procedia - Social and Behavioral Sciences,* Volume 108, pp. 605-609.

Geraldi, J. & Lechter, T., 2012. Gantt charts revisited: A critical analysis of its roots and implications to the management of projects today. *International Journal of Managing Projects in Business,* 5(4), pp. 578-594.

Google Assistant, 2023. *Google Assistant, your own personal Google..* [Online]
Available at: https://assistant.google.com/
[Accessed 15 08 2023].

Gujarathi, A. et al., 2018. Competent K-means for Smart and Effective E-commerce. *Artificial Intelligence and Evolutionary Computations in Engineering Systems. Advances in Intelligent Systems and Computing,* Volume 668.

Gupta, A., 2014. E-COMMERCE : ROLE OF E-COMMERCE IN TODAY'S BUSINESS. *International Journal of Computing and Corporate Research ,* 4(1).

Hans, R. T., 2021. Deliverable-Oriented Work Breakdown Structure: A Software Project Scope Verification Tool. *Current Topics on Mathematics and Computer Science,* Volume 1, pp. 59-62.

Highsmith, J. A., 2002. *Agile Software Development Ecosystems.* s.l.:Addison-Wesley Professional.

Hsu, C.-L. & Chen, M.-C., 2018. How gamification marketing activities motivate desirable consumer behaviors: Focusing on the role of brand love. *Comput. Hum. Behav.,* Volume 88, pp. 121-133.

Hussien, F. T. A., A. M. S. R. & A. M. S. R., 2021. Recommendation Systems For E-commerce Systems An Overview. *Journal of Physics: Conference Series.*

Hwangbo, H., Kim, Y. S. & Cha, K. J., 2018. Electronic Commerce Research and Applications. *Recommendation system development for fashion retail e-commerce,* Volume 28, pp. 94-101.

IBM Watson, 2023. *IBM Watson.* [Online]
Available at: https://www.ibm.com/watson
[Accessed 10 08 2023].

ITU, 2015. Measuring the Information Society Report 2015. *International Telecommunication Union.*

Iwanaga, J., Nishimura, N., Sukegawa, N. & Takano, Y., 2019. Improving collaborative filtering recommendations by estimating user preferences from clickstream data. *Electronic Commerce Research and Applications,* Volume 37.

Jabakji, A. & Dağ, H., 2016. Improving item-based recommendation accuracy with user's preferences on Apache Mahout. *2016 IEEE International Conference on Big Data (Big Data),* pp. 1742-1749.

Jai, T.-M. (., Burns, L. D. & King, N. J., 2013. The effect of behavioral tracking practices on consumers' shopping evaluations and repurchase intention toward trusted online retailers. *Computers in Human Behavior,* 29(3), pp. 901-909.

Jiang, W., Wang, G., Bhuiyan, M. Z. A. & Wu, J., 2016. Understanding Graph-based Trust Evaluation in Online Social Networks: Methodologies and Challenges. *ACM Computing Surveys, Vol. V, No. N, Article A.*

Jung, S., 2019. Semantic vector learning for natural language understanding. *Computer Speech Language,* Volume 56, pp. 130-145.

Jung, S., 2019. Semantic vector learning for natural language understanding. *Computer Speech Language,* pp. 130-145.

Kansana, K., Khan, J. & Bhat, S. A., 2016. A review paper on e-commerce. *Asian Journal of Technology & Management Research,* 6(1).

Khanna, A. et al., 2015. A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence. *International Journal of u- and e- Service, Science and Technology,* Volume 8, pp. 277-284.

Klopfenstein, L. C., Delpriori, S., Malatini, S. & Bogliolo, A., 2017. The rise of bots: A survey of conversational interfaces, patterns, and paradigms. *Proceedings of the 2017 conference on designing interactive systems,* pp. 555-565.

Koivisto, J. & Hamari, J., 2014. Demographic differences in perceived benefits from gamification. *Computers in Human Behavior,* Volume 35, pp. 179-188.

Kucherbaev, P., Bozzon, A. & Houben, G.-J., 2018. Human-aided bots. *IEEE Internet Computing,* 22(6), pp. 36-43.

Kumar, P. & Reddy, G. R. M., 2018. Friendship Recommendation System Using Topological Structure of Social Networks. *Praveen Kumar ; G. R. M. Reddy.*

Lin, D. & Jingtao, S., 2015. Third International Conference on Cyberspace Technology (CCT 2015). In: *A recommender system based on contextual information of click and purchase data to items for e-commerce.* s.l.:s.n., pp. 1-6.

Li, W. et al., 2018. A collaborative filtering recommendation method based on discrete quantum-inspired shuffled frog leaping algorithms in social networks. *Future Generation Computer Systems,* Volume 88, pp. 262-270.

Loeliger, J. & McCullough, M., 2009. *Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development.* s.l., s.n.

Lorenzi, F. & Ricci, F., 2005. Case-Based Recommender Systems: A Unifying View. *Intelligent Techniques for Web Personalization,* Volume 3169.

MacGregor, R. a. V. L., 2005. *Advanced Topics in Electronic Commerce, Volume 1.* s.l.:IGI Global.

Marcotte, E., 2010. *Responsive Web Design.* s.l.:s.n.

Marietto, M. d. G. B. et al., 2013. ARTIFICIAL INTELLIGENCEMARKUPLANGUAGE: A BRIEF TUTORIAL. *International Journal of Computer Science & Engineering Survey (IJCSES),* 4(3).

Marietto, M. d. G. B. et al., 2013. Artificial Intelligence MArkup Language: A Brief Tutorial. *International Journal of Computer science and engineering Survey (IJCSES).*

Martens, C. D. P., 2022. Challenges in the implementation of internet of things projects and actions to overcome them. *Technovation,* Volume 118.

Mir, F. A. & Pinnington, A. H., 2014. Exploring the value of project management: Linking Project Management Performance and Project Success. *International Journal of Project Management,* 32(2), pp. 202-217.

Molnár, G. & Szüts, Z., 2018. *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY).* s.l.:IEEE.

Munassar, N. M. A. & Govardhan, A., 2010. A Comparison Between Five Models Of Software Engineering. *International Journal of Computer Science Issues,* 7(5), pp. 94-97.

Nilashi, M. et al., 2013. Collaborative filtering recommender systems. *Research Journal of Applied Sciences, Engineering and Technology,* Volume 5, pp. 4168-4182.

Nimavat, K. & Champaneria, T., 2017. Chatbots: An overview types, architecture, tools and future possibilities. *Int. J. Sci. Res. Dev,* 5(7), pp. 1019-1024.

Pasaribu, B., Y. L. & L. S. R., 2019. Development of Risk-Based Standardized Work Breakdown (WBS). *Earth and Environmental Science.*

Pressman, R. S. & Maxim, B. R., 2014. *Software Engineering: A Practitioner's Approach.* s.l.:McGraw-Hill Education.

Project Manager PM, 2023. *Project Planning: How to Make a Project Plan.* [Online]
Available at: https://www.projectmanager.com/guides/project-planning

Rahayu, R. & Rahayu, R., 2015. Determinant Factors of E-commerce Adoption by SMEs in Developing Country. *Procedia - Social and Behavioral Sciences,* Volume 195, pp. 142-150.

Ramesh, B. & Reeba, R., 2017. Secure recommendation system for E-commerce website. *International Conference on Circuit ,Power and Computing Technologies (ICCPCT),* pp. 1-5.

Ramesh, K., Ravishankaran, S., Joshi, A. & Chandrasekaran, K., 2017. A survey of design techniques for conversational agents. *International conference on information, communication and computing technology,* pp. 336-350.

Ramesh, K., Ravishankaran, S., Joshi, A. & Chandrasekaran, K., 2017. *International conference on information, communication and computing technology.* s.l.:Springer.

Ramesh, K., Ravishankaran, S., Joshi, A. & Chandrasekaran, K., 2017. *International conference on information, communication and computing technology.* s.l.:Springer.

Ricci, F. et al., 2006. *Case-based travel recommendations.* s.l., s.n.

RiveScript, 2023. *What is RiveScript?.* [Online]
Available at: https://www.rivescript.com/about
[Accessed 15 07 2023].

Salehi, F., Abdollahbeigi, B., Langroudi, A. C. & Salehi, F., 2012. The Impact of Website Information Convenience on E-commerce Success of Companies. *Procedia - Social and Behavioral Sciences,* Volume 57, pp. 381-387.

Sardi, L., Idri, A. & Fernández-Alemán, J. L., 2017. A systematic review of gamification in e-Health. *Journal of Biomedical Informatics,* Volume 71, pp. 31-48.

Schwaber, K. & Sutherland, J., 2012. *The Scrum Guide.* s.l., s.n.

Schwalbe, K., 2013. *Information Technology Project Management, Revised.* 7, revised ed. s.l.:Cengage Learning, 2013.

Scopus, 2022. *Document search.* [Online]
Available at: https://www.scopus.com/search/form.uri?display=basicReturn
[Accessed 03 08 2023].

Scully-Allison, C. & Isaacs, K. E., 2021. Design and Evaluation of Scalable Representations of Communication in Gantt Charts for Large-scale Execution Traces. *Human-Computer Interaction.*

Shawar, B. A. & Atwell, E., 2007. Chatbots: are they really useful?. *Journal for Language Technology and Computational Linguistics,* 22(1), pp. 29--49.

Silva, B. D., Silva, S. d. & Bhuptai, R. S., 2010. Behavioral Aspect of Teenagers towards Internet Banking: An Empirical Study. *Indian Journal of Marketing,* Volume 40, pp. 44-53.

Singh, S., Darbari, H., Bhattacharjee, K. & Verma, S., 2016. Open source NLG systems: A survey with a vision to design a true NLG system. *Int J Control Theory Appl,* 9(10).

Siri, 2023. *Siri.* [Online]
Available at: https://www.apple.com/siri/
[Accessed 29 07 2023].

Snyder, H., 2019. Journal of Business Research. *Literature review as a research methodology: An overview and guidelines,* Volume 104, pp. 333-339.

Snyder, H., 2019. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research,* Volume 104.

Sommerville, I., 2016. *Software Engineering, Global Edition.* 10 ed. s.l.:Pearson Higher Ed, 2016.

Starbucks, 2023. [Online]
Available at: https://www.starbucks.tt/
[Accessed 10 08 2023].

Statista, 2021. *Forecast number of mobile users worldwide from 2020 to 2025.* [Online]
Available at: https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/
[Accessed 10 08 2023].

Statista, 2022. *Percentage of mobile device website traffic worldwide from 1st quarter 2015 to 4th quarter 2022.* [Online]
Available at: https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/
[Accessed 25 07 2023].

Suleiman, D. A., Awan, T. M. & Javed, M., 2021. Enhancing digital marketing performance through usage intention of AI-powered websites. *IAES International Journal of Artificial Intelligence (IJ-AI),* 10(4), pp. 810-817.

Taher, G., 2021. E-commerce: advantages and limitations. *International Journal of Academic Research in Accounting Finance and Management Sciences,* 11(1), pp. 153-165.

Thomas, N. T., 2016. An e-business chatbot using AIML and LSA. *International Conference on Advances in Computing, Communications and Informatics (ICACCI),* pp. 2740-2742.

Turing, A. M., 2009. *Computing machinery and intelligence.* s.l.:Springer.

UNCTAD, 2021. *Global e-commerce jumps to $26.7 trillion, fuelled by COVID-19.* [Online]
Available at: https://news.un.org/en/story/2021/05/1091182

Verloop.io, 2023. *Best Open Source Chatbot Platforms of 2023? Look No Further!.* [Online]
Available at: https://verloop.io/blog/the-best-open-source-chatbot-platforms/
[Accessed 21 07 2023].

Wallace, R. S., 2009. *The anatomy of ALICE.* s.l.:Springer.

Wang, K. et al., 2020. E-commerce personalized recommendation analysis by deeply-learned clustering. *Journal of Visual Communication and Image Representation,* Volume 71.

Weizenbaum, J., 1996. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM,* 9(1), pp. 36-45.

Wysocki, R. K., 2019. *Effective Project Management: Traditional, Agile, Extreme, Hybrid.* s.l.:© 2019 John Wiley & Sons, Inc..

Xiao, Y. & Watson, M., 2017. Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research (JPER),* 39(1), p. 93–112.

Xiong, J. et al., 2021. Augmented reality and virtual reality displays: emerging technologies and future perspectives. 10(216).

Yang, Q. et al., 2015. Exploring consumer perceived risk and trust for online payments: An empirical study in China's younger generation. *Computers in Human Behavior,* Volume 50, pp. 9-24.

Yudhoatmojo, S. B. & Ramadana, R., 2016. *Analysis on Gamificaiton Features Usage on Indonesia e-Commerce Sites using Octalysis Framework.* s.l., s.n.

Zahir, A., Yuan, Y. & Moniz, K., 2019. A Simple Implicit Trust Inference Model for Memory-Based Collaborative Filtering Recommendation Systems. *Electronics.*

Zumstein, D. & Hundertmark, S., 2017. CHATBOTS--AN INTERACTIVE TECHNOLOGY FOR PERSONALIZED COMMUNICATION, TRANSACTIONS AND SERVICES. *IADIS International Journal on WWW/Internet,* 15(1).

Appendices

Appendix A: Requirements Analysis – Functional and Non-Functional Requirements

Below showing the remaining functional requirements, with its description and rationale

Table 29: Displaying the breakdown of the second requirement FR2 (Login).

Requirement	Login
Requirement Number	FR2
Description	By entering their special login credentials, users can access tailored content and capabilities within the application or platform thanks to the login capability.
Rationale	The integrity of the platform and the safety of user data depend on a quick and safe login procedure. By limiting access to tailored information and features to only those who are permitted, it protects user privacy and improves user experience in general.

Table 30: Displaying the breakdown of the third requirement FR3 (User Management Profile).

Requirement	User Management Profile
Requirement Number	FR3
Description	Users can create, edit, and maintain their personal data, preferences, and settings inside the program via user profile management.
Rationale	Through the provision of profile management capabilities, the platform cultivates a feeling of customization and authority for users. A more personalized and engaging user experience is made possible by the users' ability to modify their settings, update their information, and customize their experiences.

Table 31: Displaying the breakdown of the fourth requirement FR4 (Product Catalogue).

Requirement	Product Catalogue
Requirement Number	FR4
Description	The product catalogue comprises an exhaustive list of all the products that the platform offers, together with

	specific details such product descriptions, photos, and prices.
Rationale	A thorough and well-structured product catalogue provides users with the framework to investigate and decide what to buy. It improves the usability of the platform by making it easier for customers to browse and find the products they want, which raises the possibility of successful conversions and increases customer happiness.

Table 32: Displaying the breakdown of the fifth requirement FR5 (Search Functionality).

Requirement	**Search Functionality**
Requirement Number	FR5
Description	Using keyword-based searches, the search feature enables users to locate particular goods, services, or information on the platform quickly.
Rationale	Effective search features greatly improve usability and accessibility for users, making it easier for them to find the products or content they want more quickly. It improves customer happiness and engagement by streamlining the user experience and making it simpler for consumers to access pertinent products and information.

Table 33: Displaying the breakdown of the sixth requirement FR6 (Product Filtering).

Requirement	**Product Filtering**
Requirement Number	FR6
Description	Users can refine and focus their product searches using product filtering by selecting particular parameters like price range, category, or other pertinent features.
Rationale	Product filtering lets users tailor their product searches to their needs and tastes, which improves their browsing experience. It makes the process of product exploration more effective and focused, which raises user happiness and boosts conversion rates.

Table 34: Displaying the breakdown of the seventh requirement FR7 (Product Detail Page).

Requirement	Product Detail Page
Requirement Number	FR7
Description	A product's characteristics, photographs, user reviews, and thorough descriptions are all included on the product detail page.
Rationale	To give users comprehensive information about the products they are interested in, a well-organized product detail page is essential. Making educated decisions about purchases, encouraging openness, and developing user trust all increase the likelihood of profitable deals and client retention.

Table 35: Displaying the breakdown of the eighth requirement FR8 (Add to Cart).

Requirement	Add to Cart
Requirement Number	FR8
Description	Users can gather chosen products for later purchase without starting the checkout process right away by using the "Add to Cart" feature.
Rationale	By enabling users to gather their favorite products and buy them whenever it's convenient for them, the "Add to Cart" feature streamlines their purchasing experience. Improved customer satisfaction and higher sales result from encouraging consumers to explore more products and providing a smooth and convenient purchasing experience.

Table 36: Displaying the breakdown of the ninth requirement FR9 (Shopping Cart Management).

Requirement	Shopping Cart Management
Requirement Number	FR9
Description	Before completing the checkout process, consumers can review and modify the items in their shopping basket with shopping cart management. It has the ability to change quantities, take things out, and apply discounts.
Rationale	Good shopping cart management gives customers authority over the items they intend to buy, giving them the freedom to check, adjust, and complete their orders as necessary. It facilitates repeat business and customer loyalty by streamlining the purchasing

	process, lowering user annoyance, and improving the entire shopping experience.

Table 37: Displaying the breakdown of the tenth requirement FR10 (Checkout Process).

Requirement	**Checkout Process**
Requirement Number	FR10
Description	Before completing the transaction, the user must provide the required information during the checkout procedure, such as the shipping information, the payment method, and an order review.
Rationale	A seamless and safe transaction experience depends on an effective and user-friendly checkout procedure. It increases the conversion rate and fosters favorable user opinions of the platform by minimizing user friction, streamlining the payment process, and lowering the chance of cart abandonment.

Table 38: Displaying the breakdown of the eleventh requirement FR11 (Payment Integration).

Requirement	**Payment Integration**
Requirement Number	FR11
Description	Payment integration enables users to transact within the platform with ease by facilitating the safe processing of many payment methods.
Rationale	Ensuring a secure and convenient payment experience for users requires a strong payment integration system. It boosts user happiness and promotes recurring usage and purchases by establishing trust in the platform, reducing obstacles linked to payments, and guaranteeing a smooth transaction procedure.

Table 39: Displaying the breakdown of the twelfth requirement FR12 (Order Confirmation).

Requirement	**Order Confirmation**
Requirement Number	FR12
Description	Users who complete their orders can get order numbers, itemized lists, and projected delivery dates through the order confirmation.
Rationale	In order to reassure users that their transactions have been completed correctly, order confirmation is an essential step. Users' trust in the platform is increased,

	and future interactions and transactions are encouraged, as it gives them a sense of security, transparency, and responsibility.

Table 40: Displaying the breakdown of the thirteenth requirement FR13 (Order History).

Requirement	Order History
Requirement Number	FR13
Description	Users can view and examine their previous purchase records, including order details, transaction dates, and payment information, by accessing and reviewing their order history.
Rationale	Users are able to monitor their previous interactions and transactions on the platform thanks to the availability of a complete order history function. It increases user pleasure and encourages sustained involvement with the platform by fostering accountability, openness, and ease of reference for future purchases.

Table 41: Displaying the breakdown of the fourteenth requirement FR14 (User Review and Rating).

Requirement	User Review and Rating
Requirement Number	FR14
Description	Users can share their experiences and offer feedback on the goods and services available on the site by using user reviews and ratings.
Rationale	User opinions and ratings have a big impact on what other users think of the platform and what they decide to buy. They encourage openness, establish trustworthiness, and develop a feeling of community, all of which boost user engagement, loyalty, and trust.

Table 42: Displaying the breakdown of the fifteenth requirement FR15 (Account Management).

Requirement	Account Management
Requirement Number	FR15
Description	Users can monitor and manage their privacy settings, security features, and account settings on the site through account management.
Rationale	Users that have efficient account management are able to adjust their privacy and security settings,

	update their information, and customize their experiences. In addition to promoting a safe and customized user experience within the platform, it increases user happiness and a sense of ownership.

Table 43: Displaying the breakdown of the sixteenth requirement FR16 (Admin Dashboard).

Requirement	**Admin Dashboard**
Requirement Number	FR16
Description	Administrators have a centralized platform to monitor and control many parts of the platform, such as product management, sales statistics, and user administration, with the help of an admin dashboard.
Rationale	An essential tool for administrators to effectively monitor and control platform activities is the admin dashboard. It makes managing the platform easier, makes data-driven decisions easier, and improves operational effectiveness, all of which contribute to the platform's overall success and long-term viability.

Table 44: Displaying the breakdown of the seventeenth requirement FR17 (Chatbot Integration).

Requirement	**Chatbot Integration**
Requirement Number	FR17
Description	Using an automated messaging system that can communicate with consumers, offer support, and respond to inquiries instantly is known as chatbot integration.
Rationale	By offering prompt assistance and direction, chatbot integration improves user engagement and the user experience as a whole. In the end, it increases customer happiness and retention by streamlining user interactions, cutting down on response times, and fostering a more effective and user-friendly communication process.

Table 45: Displaying the breakdown of the eighteenth requirement FR18 (Recommendation Engine).

Requirement	**Recommendation Engine**
Requirement Number	FR18
Description	Recommendation engines leverage user data and behavior to offer tailored product recommendations

	and suggestions to users according to their browsing history and interests.
Rationale	A strong recommendation engine may simplify the process of finding the right product and entice consumers to look into other options by providing tailored and pertinent product recommendations. It makes upselling and cross-selling easier, which raises client happiness and boosts conversion rates.

Table 46: Displaying the breakdown of the nineteenth requirement FR19 (Gamification).

Requirement	**Gamification**
Requirement Number	FR19
Description	Gamification features are the addition of aspects from games, such challenges, badges, and points, to a platform in order to encourage user participation and engagement.
Rationale	Gamification features encourage users to actively participate and explore different elements of the platform by fostering a dynamic and engaging user experience. They enhance user retention, foster a feeling of satisfaction and achievement, and encourage user loyalty, all of which eventually contribute to the platform's long-term viability and success.

Table 47: Displaying the breakdown of the twentieth requirement FR20 (Advanced Analytics and Data Visualization).

Requirement	**Advanced Analytics and Data Visualization**
Requirement Number	FR20
Description	In order to glean insights and patterns from intricate datasets, advanced analytics and data visualization employ a variety of sophisticated data analysis methods and visual aids.
Rationale	Extraordinary analytics and data visualization are necessary to wring useful insights from complex and big datasets. They make it easier to make decisions based on data, make it possible to spot patterns and connections, and offer insightful data that can be used to improve overall performance and optimize corporate plans. These technologies help create clear and understandable business solutions by presenting

	data in an aesthetically pleasing and easily understood way. They also encourage informed decision-making.

Table 48: Displaying the breakdown of the twenty first requirement FR21 (Responsive Design).

Requirement	**Responsive Design**
Requirement Number	FR21
Description	The development methodology that guarantees a website or application's smooth adaptation and ideal presentation across a range of devices and screen sizes, such as PCs, tablets, and smartphones, is known as responsive design.
Rationale	Ensuring a consistent and user-friendly experience for users across all devices to access the platform is dependent on responsive design. By increasing accessibility, reducing difficulties with the user interface, and guaranteeing that the material is easily readable and navigable, it increases user retention and happiness. Responsive design encourages diversity and accessibility by supporting a range of screen sizes and resolutions. This helps the platform reach a wider audience and improve its overall competitiveness in the market.

Appendix B: Data Dictionary

Below are showing the remaining data dictionaries.

Table 49: Displaying the data dictionary for the products entity.

PRODUCTS							
Description: Product Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
product_id	bigint	10	PK	No	None	Autoincremented	Uniquely identifies every product.
cat_id*	bigint	10	FK	No	None	Must exist in categories table	Reference to the category the product belongs to.
product_name	varchar	255	No	No	None	None	Name of the product.
product_desc	varchar	2000	No	Yes	None	None	Full description of the product.
product_price	decimal	10,2	No	No	None	None	Price of the product.
product_quantity	integer	10	No	No	None	None	Available quantity of the product.
product_status	varchar	50	No	No	None	None	Status of the product (e.g., available, discontinued).

product_tags	varchar	255	No	Yes	None	None	Tags associated with the product.
product_date	date		No	Yes	None	None	Date the product was added or modified.
product_comment_count	integer	10	No	Yes	None	None	Number of comments on the product.
productimg_1	varchar	255	No	Yes	None	None	Path to the first product image.
productimg_2	varchar	255	No	Yes	None	None	Path to the second product image.
productimg_3	varchar	255	No	Yes	None	None	Path to the third product image.
productimg_4	varchar	255	No	Yes	None	None	Path to the fourth product image.
stripe_product_id	varchar	200	No	Yes	None	None	Stripe's identifier for the product.
stripe_product_url	varchar	250	No	Yes	None	None	URL to the product on Stripe.

Table 50: Displaying the data dictionary for the categories entity.

CATEGORIES							
Description: Category Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
cat_id	bigint	10	PK	No	None	Autoincremented	Uniquely identifies every category.
cat_title	varchar	255		No	None		Name of the category.

cat_img	varchar	255		Yes	None		Path to the category image.
is_active	tinyint	1		No	1		Indicates if the category is active.

Table 51: Displaying the data dictionary for the reviews entity.

REVIEWS							
Description: Review Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
review_id	bigint	10	PK	No	None	Autoincremented	Uniquely identifies every review.
product_id	bigint	10	FK	No	None	Must exist in products table	Reference to the reviewed product.
user_id	bigint	10	FK	No	None	Must exist in users table	Reference to the user who wrote the review.
review_author	varchar	255		No	None		Name of the review author.
review_email	varchar	255		No	None	Must be email format	Email of the review author.
review_content	text			No	None		Content of the review.

review_status	varchar	255		No	None		Status of the review (e.g., approved, pending).
review_date	timestamp			No	None		Date and time the review was posted.

Table 52: Displaying the data dictionary for the orders entity.

ORDERS							
Description: Order Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
order_id	integer	10	PK	No	None	Autoincremented	Uniquely identifies every order.
user_id*	integer	10	FK	No	None	Must exist in users table	Reference to the user who placed the order.
stripe_session_id	varchar	200		Yes	None		Identifier for the Stripe payment session.
order_amount	decimal	10, 2		No	None		Total amount of the order.
order_transaction	varchar	255		Yes	None		Transaction identifier for the order.
order_status	varchar	255		No	None		Current status of the order.

order_currency	varchar	255		No	None		Currency in which the order was placed.
create_at	timestamp			No	None		Timestamp when the order was created.

Table 53: Displaying the data dictionary for the reports entity.

REPORTS							
Description: Report Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
report_id	integer	10	PK	No	None	Autoincremented	Uniquely identifies every report.
product_id*	integer	10	FK	No	None	Must exist in products table	Reference to the reported product.
stripe_product_id	varchar	200		Yes	None		Stripe identifier for the reported product.
stripe_price_id	varchar	200		Yes	None		Stripe identifier for the price of the product.
user_id*	integer	10	FK	No	None	Must exist in users table	Reference to the user related to the report.
order_id*	integer	10	FK	No	None	Must exist in orders table	Reference to the order related to the report.
product_price	decimal	10, 2		No	None		Price of the product at the

							time of the report.
product_title	varchar	255		No	None		Title of the reported product.
product_quantity	integer			No	None		Quantity of the product reported.
product_total	decimal	10, 2		No	None		Total value of the product reported.
create_at	timestamp			No	None		Timestamp when the report was created.

Table 54: Displaying the data dictionary for the spinner winners entity.

SPINNER WINNERS							
Description: Spinner winners Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
winner_id	integer	10	PK	No	None	Autoincremented	Uniquely identifies each winner record.
product_id*	integer	10	FK	No	None	Must exist in products table	Reference to the product that was won.
user_id*	integer	10	FK	No	None	Must exist in users table	Reference to the user who won the product.
win_date	timestamp			No	None		Date and time when the win was recorded.

Table 55: Displaying the data dictionary for the user spins entity.

USER SPINS							
Description: User Spin Details							
Field Name	Datatype	Length	Index	Null	Default	Validation Rule	Description
id	integer	10	PK	No	None	Autoincremented	Uniquely identifies each spin record.
user_id*	integer	10	FK	No	None	Must exist in users table	Reference to the user who made the spin.
spin_date	timestamp			No	None		Date and time when the spin was made.

Appendix C: Project Progress Reports

UNIVERSITY OF BEDFORDSHIRE

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

FINAL YEAR UG PROJECT: CIS017-3 SEM1

WEEKLY PROGRESS REPORT FORM

Student's Name: Saddiq Abdul Qadir	**Supervisor's Name:** Mr. Ravi Ragoonath
Month: September 2023	Report No. 1
Summary of progress (including any problems)	**<u>PROGRESS I HAVE MADE TO DATE:</u>** **WORK DONE ON REFLECTIVE REPORT:** Completed Section 1: Introduction • Completed the following subsections. • The Aim • The Objectives • The Problem • The Purpose of the Reflective Report Completed Section 4: Project Report Contents • Completed the following subsections. • The Role of the Project Report • List of the chapters that will be included in the final report. • Detailed discussion of the content of each chapter.

	WORK DONE ON ARTEFACT: Completed the following with the software artefact: • Installation of software development environment • Revised PHP, HTML, CSS, and Bootstrap • Implemented Version Control USING Git and GitHub • Created and designed the header and footer. • Creating and designed html forms for the login and registration process • Create Database using phpMyAdmin. • Connected the database and created the user’s table. • Implemented the login and registration functionality. **<u>WORK I’VE LEARNT:</u>** ❖ A detailed overview of non-functional needs, as well as fundamental and advanced functional requirements. ❖ Gantt Chart construction using Microsoft Excel. ❖ CSS, HTML, PHP, and JavaScript coding knowledge. - How to create a login and registration form with functionality. - Learned various CSS topics such as flexbox and grid. - How to create a database and connect it to a project. ❖ Development tool implementation including: - Visual Studio Code - https://code.visualstudio.com - Laragon - https://laragon.org/download/index.html

	- GitHub account creation for commits and managing version control - https://github.com/ **CHALLENGES ENCOUNTERED:** During the development of my login and registration system, I stumbled upon a serious problem with how I handled user passwords. Initially, I was storing passwords using a basic encryption method, which I later recognized was not a secure practice. To overcome this problem, I performed some research and discovered that PHP offers a built-in function for this problem called hash. I choose to use PHP's built-in hash() function, which provides a secure method of hashing passwords. I found the solution to my problem at the following link: https://www.php.net/manual/en/function.password-hash.php
Plan for next week	**THE FOLLOWING ARE THE PLANS FOR THE NEXT 2 WEEKS.** Reflective Report: **Work on Section 3: Reflection** Introduction 3.1 The Project Plan 3.2 The Artefact 3.3 Project Progress Report 3.4 Experiences Gained Software Artefact: This week focus is an overview of Design pages and getting data from database.

	• Review existing e-commerce websites. • Review and practice CSS topics • Design the products, details, cart and home pages. • Review database diagram and update as necessary. • Create products and cart pages. • Display products dynamically from the database • Include the header and footer in each page.
Supervisor’s comments	***A very detailed report provided with excellent articulation of the work done. I am very pleased with the level of progress made on the project as well as the work done on the artefact.*** ***Excellent project management displayed with the student clearly identifying plans for the next week.***

Student's Name: Saddiq Abdul Qadir	**Supervisor's Name:** Mr. Ravi Ragoonath
Month: October	Report No. 2
Summary of progress (including any problems)	**<u>WORK COMPLETED TO DATE:</u>** **Reflective Report:** **Completed the following work on the Reflective Report** **Completed Section 1: Introduction** • Completed the following subsections. • The Aim • The Objectives • The Problem • The Purpose of the Reflective Report **Completed Section 3: Reflection** • Completed the following subsections. • Introduction • The Project Plan • The Artefact • Project Progress Report • Experiences Gained **Completed Section 4: Project Report Contents** • Completed the following subsections. • The Role of the Project Report

	• List of the chapters that will be included in the final report. • Detailed discussion of the content of each chapter. **Software Artefact:** Completed the following work with the software artefact: • Installation of software development environment • Revised PHP, HTML, CSS, and Bootstrap • Implemented Version Control using Git and GitHub • Created and designed the header and footer. • Creating and designed html forms for the login and registration process • Create Database using phpMyAdmin. • Connected the database and created the user's table. • Implemented the login and registration functionality. • Review existing e-commerce websites. • Review and practice CSS topics • Design the products, details, cart and home pages. • Review database diagram and update as necessary. • Create products and cart pages. • Display products dynamically from the database • Include the header and footer on each page. **WORK I'VE LEARNT:**

- Gained proficiency in using PHP to dynamically fetch and display product information from the database.
- Learned how to write server-side scripts to interact with the database and generate dynamic page content.
- Learned the importance of maintaining consistency across different web pages by including the same header and footer on each page.
- Developed problem-solving skills as I encountered and overcame various challenges during the development process.
- Developed a deeper understanding of CSS properties and how to apply them for sophisticated styling.
- I've learned to evaluate my work critically, identifying areas of strength and areas for improvement.
- Recognized the significance of ongoing learning and practice, particularly in the ever-changing field of web development.

CHALLENGES:

A challenge I stumbled upon was retrieving data from the database using PHP and dynamically displaying products on the products page. Initially, I had difficulty retrieving and displaying the product data accurately, and the product images and descriptions weren't displaying as they should.

I addressed this problem by thoroughly reviewing my PHP code and SQL queries. I realized there was an error in my SQL query that was preventing the correct data from being retrieved. The products began to appear correctly on the page after the query was corrected and the PHP code was properly written to display the data. I also made certain that the data was properly sanitized to avoid any potential security vulnerabilities. This experience taught me the value of paying attention to detail and thoroughly testing and validating data when working with dynamic content and databases.

	I had trouble articulating the precise learning outcomes and personal growth that came from the project when I was writing the reflective report on my experience building an e-commerce website. I knew exactly what I had done and the skills I had employed, but it was challenging for me to turn these experiences into succinct and understandable reflections on what I had learned. To overcome this obstacle, I decided to take a step back and divide the project into smaller parts, focusing on each task separately. I reflected on the difficulties I encountered during each task, how I overcame them, and what I learned in the process. This aided me in expressing my learning outcomes and personal development in a structured manner. In addition, I sought feedback from peers and mentors, which provided me with new perspectives and aided in the improvement of the quality of my reflective writing.
Plan for next week	**PLANS FOR THE NEXT TWO WEEKS:** **Reflective Report:** **Work on Section 2: Summary of Progress to date** **2.1 Project Management** **2.2 Project Progress Reports** **2.3 Work done since Contextual Report** **2.3.1 Requirements Analysis** **2.3.2 Design** **2.3.3 Implementation** **ARTEFACT** **Next week's focus is the Cart and Checkout**

	• Improve the design of the Cart Page • Implement Add-to-Cart Functionality • Design the Checkout Page • Design the Checkout Page • Test the Cart and Checkout Flow • Focus on User Feedback • Learn about Sessions and Cookies • Learn good Security practices in PHP. • Learn about Redirection and User Feedback • Learn about Integrating Payment Gateways
Supervisor's comments	***Outstanding level of work done to date in both the written report and the artefact development. Excellent documentation and reflection are shown in the progress reports as well. Student is well on track to produce a high-quality, exceptional piece of work if the progress continues at this pace.***

Student's Name: Saddiq Abdul Qadir		**Supervisor's Name:** Mr. Ravi Ragoonath
Month: October		Report No. 3
Summary of progress (including any problems)	**<u>WORK DONE TO DATE:</u>** **REFLECTIVE REPORT:** **Completed the following work on the Reflective Report** **Completed Section 1: Introduction** • Completed the following subsections. • The Aim • The Objectives • The Problem • The Purpose of the Reflective Report **Completed Section 2: Summary of Progress to date** • Completed the following subsections. • Project Management • Project Progress Reports • Work done since Contextual Report • Requirements Analysis • Design • Implementation **Completed Section 3: Reflection**	

- Completed the following subsections.
    - Introduction
    - The Project Plan
    - The Artefact
    - Project Progress Report
    - Experiences Gained

Completed Section 4: Project Report Contents

- Completed the following subsections.
    - The Role of the Project Report
    - List of the chapters that will be included in the final report.
    - Detailed discussion of the content of each chapter.

Software Artefact:

Completed the following work with the software artefact:

- Installation of software development environment
- Revised PHP, HTML, CSS, and Bootstrap
- Implemented Version Control using Git and GitHub
- Created and designed the header and footer.
- Creating and designing HTML forms for the login and registration process
- Create Database using phpMyAdmin.
- Connected the database and created the user’s table.
- Implemented the login and registration functionality.
- Review existing e-commerce websites.

- Review and practice CSS topics
- Design the products, details, cart and home pages.
- Review the database diagram and update as necessary.
- Create products and cart pages.
- Display products dynamically from the database
- Include the header and footer in each page.
- Improve the design the Cart Page
- Implement Add-to-Cart Functionality
- Design the Checkout Page
- Design the Checkout Page
- Test the Cart and Checkout Flow
- Focus on User Feedback
- Learn about Sessions and Cookies
- Learn good Security practices in PHP.
- Learn about Redirection and User Feedback
- Learn about Integrating Payment Gateways

WHAT WAS LEARNT:

- Creation of the login and registration wireframe using: https://www.figma.com/
- ER diagram formation using: https://visualstudio.microsoft.com/downloads/
- Enhanced Design Skills: Improved the aesthetics and usability of the Cart Page, ensuring a user-friendly experience.

	• Learned how to run comprehensive tests on the Cart and Checkout flow to ensure that all functionalities function properly. • Learned how to use sessions and cookies to maintain state across multiple pages, which is critical for cart functionality. • I learned about integrating payment gateways and handling transactions securely. • In PHP, I learned about good security practices like data validation, sanitization, and secure session management. **CHALLENGES:** A challenge was ensuring the security of user data, particularly during the checkout process, which handles sensitive information such as addresses and potential payment details. I addressed this issue by researching good PHP security practices. I discovered the significance of data validation and sanitization in preventing SQL injection and XSS attacks. I additionally established secure session management and made certain that any communication with payment gateways was secure. This not only helped to protect user data but also to build trust with website visitors.
Plan for next week	**THE FOLLOWING ARE MY PLANS FOR THE NEXT 2 WEEKS:** FINAL REPORT: **Commence work on the final report.** • Complete Chapter 1 Introduction • Complete Chapter 2 Project Management • Complete Chapter 3 Background Research • Complete Chapter 4 Requirements Analysis • Complete Chapter 5 Design

	ARTEFACT: **Focus on improving the checkout process, implement chatbot and reviewing code.** • Order Confirmation • Create a Test Plan • Integrate Chatbot and Configure Responses • Implement Code Improvements • Create a Test Plan and conduct testing.
Supervisor's comments	***Excellent progress made by the student with a clear indication of the tasks done, what was learnt, challenges and how he overcame these as well as plans for the next session.*** ***Excellent work on the artefact and the written report. Student well on track to complete the project ahead of schedule.***

Appendix D: Testing of Functional Requirements

The remaining testing of the functional requirements are displayed below:

Table 56: Displaying test for the product catalogue functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-PC-1	Click products (Navigate To) on the navbar.	All products are displayed.	As expected	Success

Table 57: Displaying test for the search functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-SF-1	Type a search query (makeup) in the search bar on the products page.	All products related to the phrase (makeup) to display	As expected	Success

Table 58: Displaying test for the product filtering functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-PF-1	Filter products by a specific brand	All products related to the specific category to display	As expected	Success

Table 59: Displaying test for the product details functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-PD-1	Navigate to any product and click details.	Product Details for the specific product to display with the option to add to cart.	As expected	Success

Table 60: Displaying test for the add-to-cart functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-ATC-1	Navigate to any product detail and select the add to cart button.	Item to be added to cart.	As expected	Success

Table 61: Displaying test for the shopping cart management functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-SCM-1	Remove a product from cart.	Item to be removed to cart.	As expected	Success
FR-SCM-2	Add one to the product quantity	For the product quantity to increase by one.	As expected	Success
FR-SCM-3	Minus one from the product quantity	For the product quantity to decrease by one.	As expected	Success

Table 62: Displaying test for the checkout process, the payment integration and the order confirmation functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-CPO-1	Add a Product to Cart then proceed to checkout. Users enter all their billing and shipping details then proceed to click make payment.	Payment and checkout made successfully and receive a confirmation message.	As expected	Success

Table 63: Displaying test for the account management functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-OH-1	A logged-in user navigates to account management and scrolls to the order history section.	For all the orders previously made to be displayed.	As expected	Success
FR-OH-2	A logged-in user navigates to account management and scrolls to the account information section.	For the user information to be displayed (email, name etc.)	As expected	Success

Table 64: Displaying test for the user reviews and ratings functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-URR-1	A user can navigate to a product detail page and add a review.	For the review to be added successfully	As expected	Success

FR-URR-2	A user can navigate to a product detail page and view reviews.	For the reviews for that specific product to be displayed.	As expected	Success

Table 65: Displaying test for the chatbot integration functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-CI-1	A user asks the chatbot a question. Data: What is the contact information?	For the contact information to be given.	As expected	Success

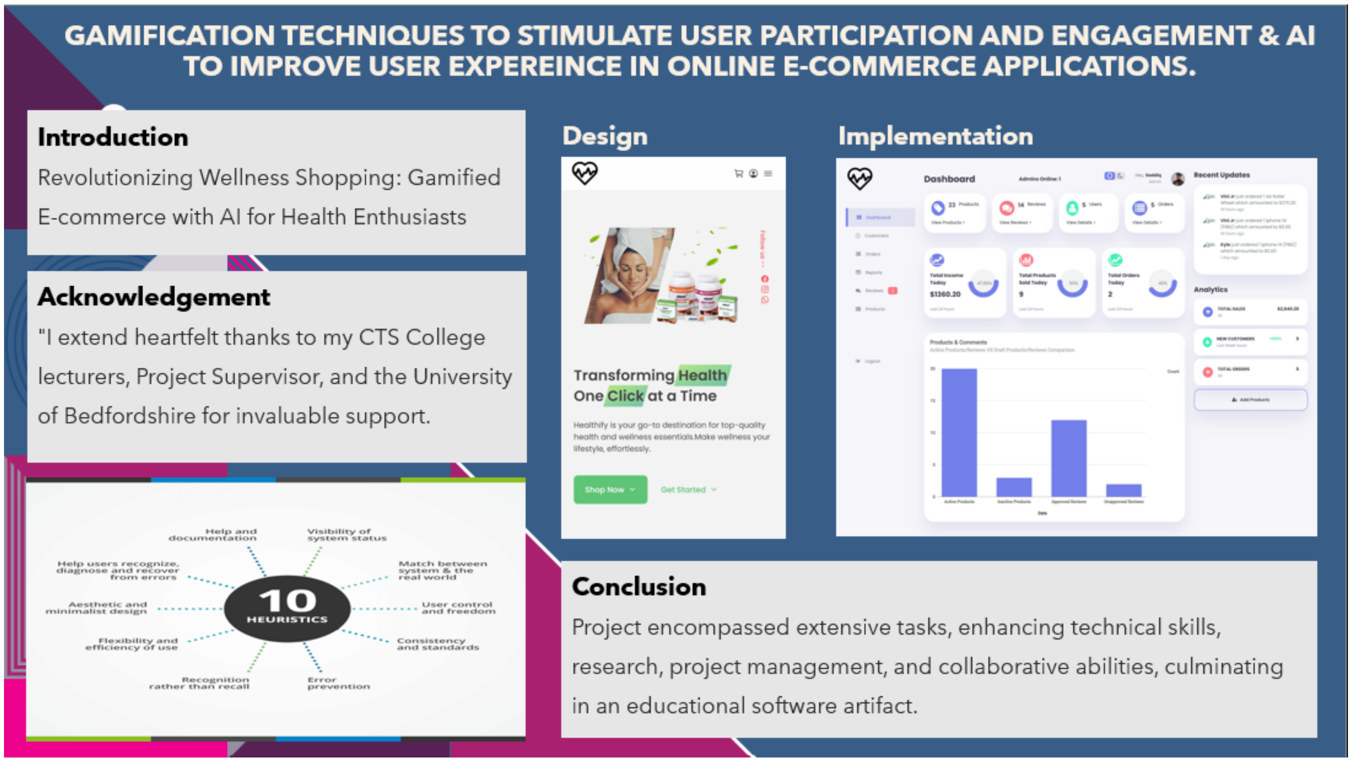
Table 66: Displaying test for the recommendation engine functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-RE-1	A user goes to the product detail page for a product and scrolls down to the recommended products section.	For all the recommended products to display.	As expected.	Success

Table 67: Displaying test for the responsive design functionality.

Test#	Test with Data	Expected Outcome	Actual Outcome	Result
FR-RD-1	A user opens the website on their phone, tablet, and laptop.	For the website to adjust/respond to the different screen sizes.	As expected.	Success

Appendix E: Poster



GAMIFICATION TECHNIQUES TO STIMULATE USER PARTICIPATION AND ENGAGEMENT & AI TO IMPROVE USER EXPEREINCE IN ONLINE E-COMMERCE APPLICATIONS.

Introduction

Revolutionizing Wellness Shopping: Gamified E-commerce with AI for Health Enthusiasts

Acknowledgement

"I extend heartfelt thanks to my CTS College lecturers, Project Supervisor, and the University of Bedfordshire for invaluable support.

Design

Transforming Health One Click at a Time

Healthify is your go-to destination for top-quality health and wellness essentials.Make wellness your lifestyle, effortlessly.

Implementation

Conclusion

Project encompassed extensive tasks, enhancing technical skills, research, project management, and collaborative abilities, culminating in an educational software artifact.

Appendix F: Sketches

The remainder of the sketches are displayed below.

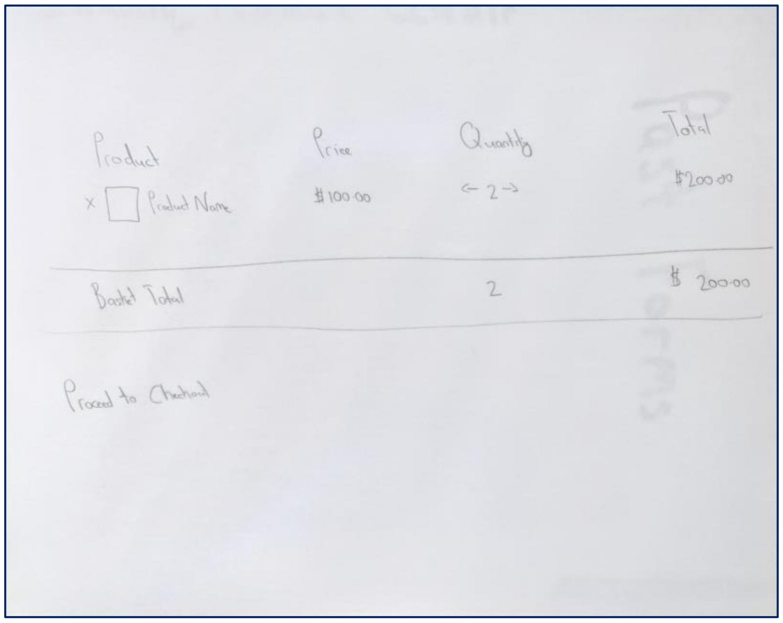


Figure 155: Displaying sketch of the cart page.

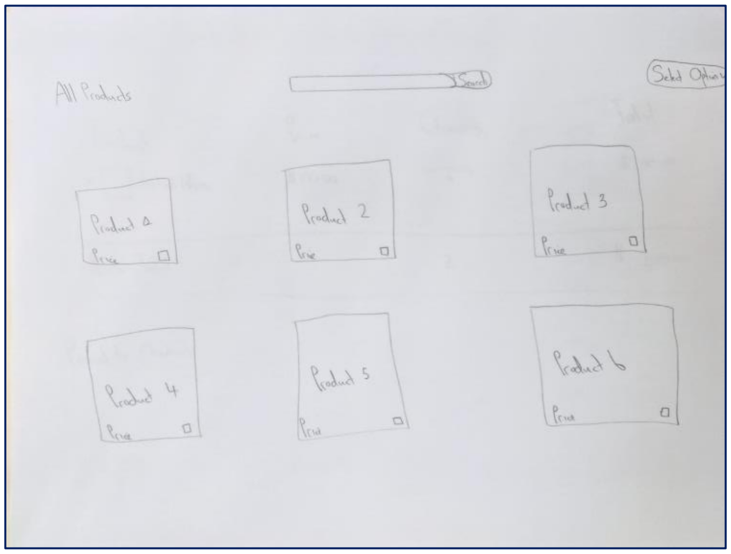
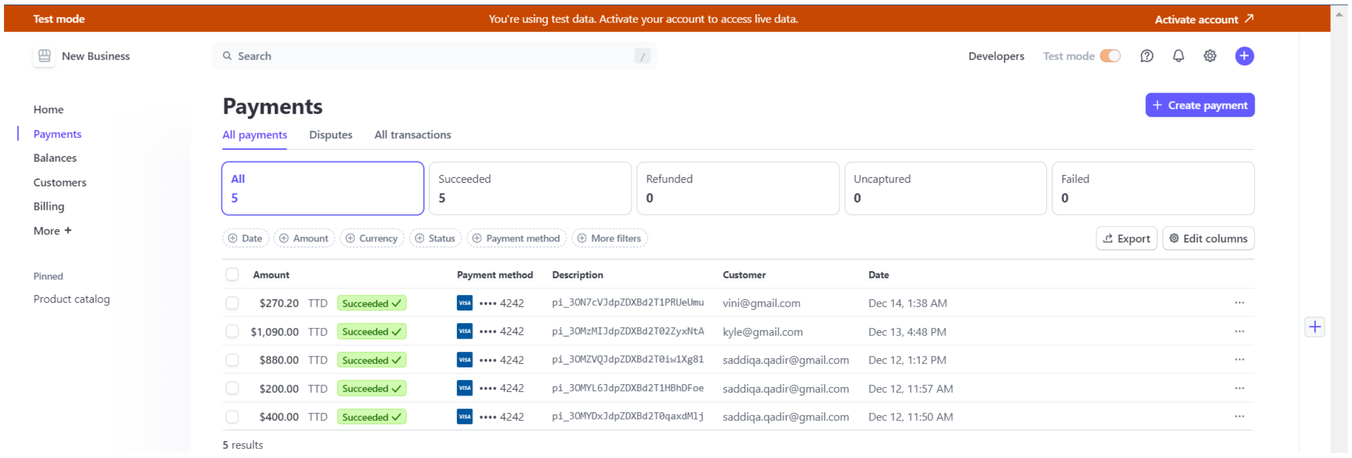


Figure 156: Displaying sketch of the products page.

Appendix G: Stripe API

Pictures from the stripe API are shown below.



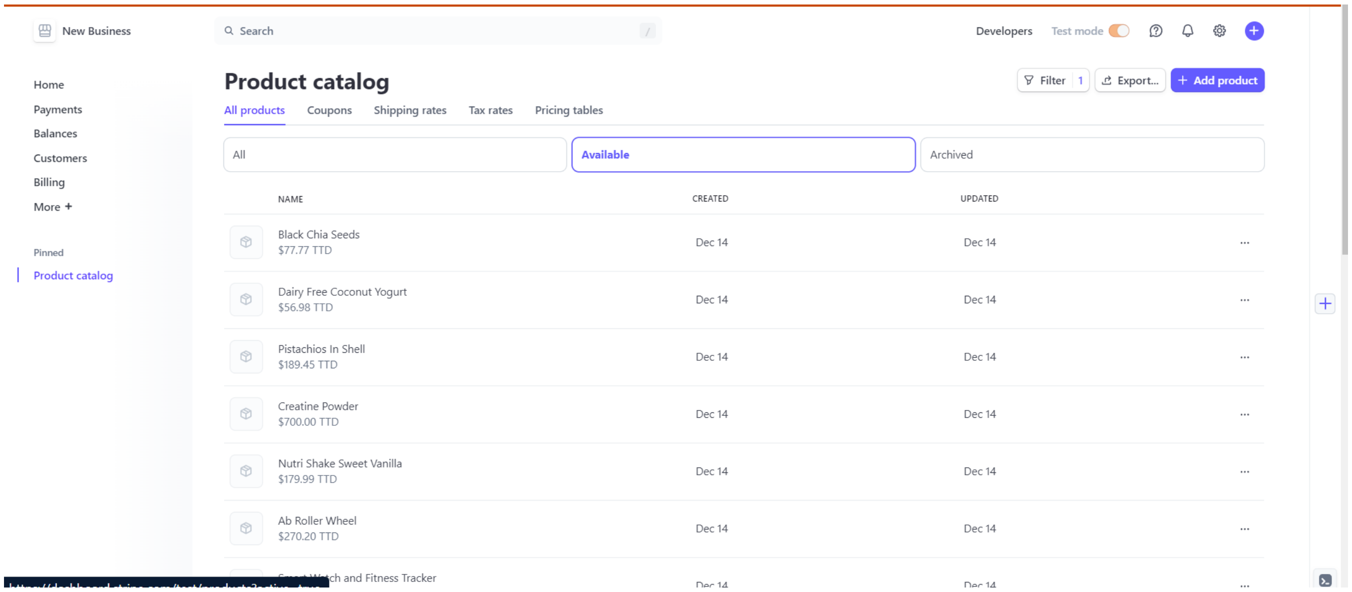
Payments

All	Succeeded	Refunded	Uncaptured	Failed
5	5	0	0	0

Amount			Payment method	Description	Customer	Date
$270.20	TTD	Succeeded ✓	•••• 4242	pi_3ON7cVJdpZDXBd2T1PRUeUmu	vini@gmail.com	Dec 14, 1:38 AM
$1,090.00	TTD	Succeeded ✓	•••• 4242	pi_3OMzMIJdpZDXBd2T02ZyxNtA	kyle@gmail.com	Dec 13, 4:48 PM
$880.00	TTD	Succeeded ✓	•••• 4242	pi_3OMZVQJdpZDXBd2T0iw1Xg81	saddiqa.qadir@gmail.com	Dec 12, 1:12 PM
$200.00	TTD	Succeeded ✓	•••• 4242	pi_3OMYL6JdpZDXBd2T1HBhDFoe	saddiqa.qadir@gmail.com	Dec 12, 11:57 AM
$400.00	TTD	Succeeded ✓	•••• 4242	pi_3OMYDxJdpZDXBd2T0qaxdMlj	saddiqa.qadir@gmail.com	Dec 12, 11:50 AM

5 results

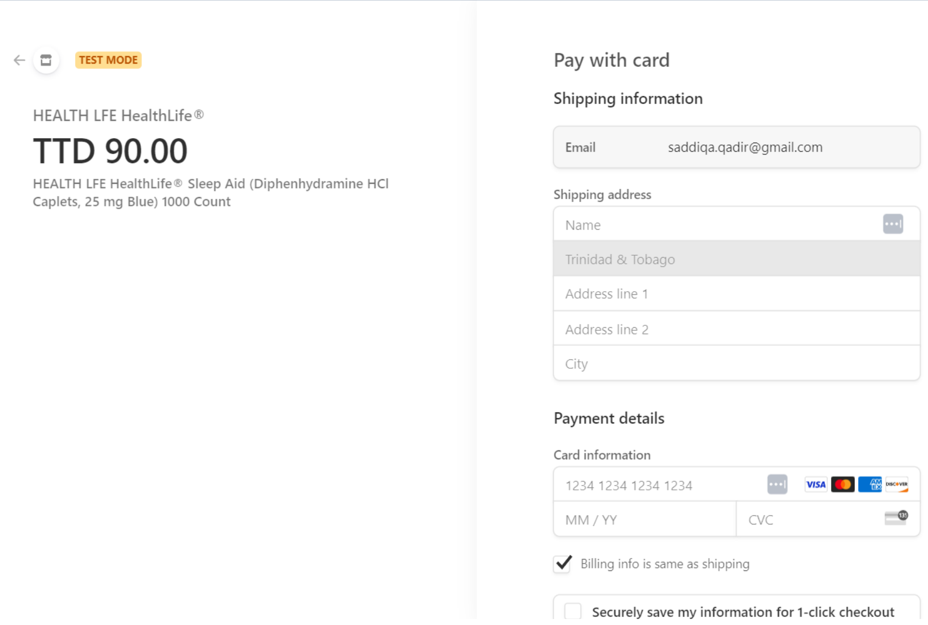
Figure 157: Displaying Stripe Payments



Product catalog

NAME	CREATED	UPDATED
Black Chia Seeds $77.77 TTD	Dec 14	Dec 14
Dairy Free Coconut Yogurt $56.98 TTD	Dec 14	Dec 14
Pistachios In Shell $189.45 TTD	Dec 14	Dec 14
Creatine Powder $700.00 TTD	Dec 14	Dec 14
Nutri Shake Sweet Vanilla $179.99 TTD	Dec 14	Dec 14
Ab Roller Wheel $270.20 TTD	Dec 14	Dec 14
ch and Fitness Tracker	Dec 14	Dec 14

Figure 158: Displaying product catalogue in stripe.



TEST MODE

HEALTH LFE HealthLife®

TTD 90.00

HEALTH LFE HealthLife® Sleep Aid (Diphenhydramine HCl Caplets, 25 mg Blue) 1000 Count

Pay with card

Shipping information

Email saddiqa.qadir@gmail.com

Shipping address

Name

Trinidad & Tobago

Address line 1

Address line 2

City

Payment details

Card information

1234 1234 1234 1234

MM / YY

CVC

☑ Billing info is same as shipping

☐ Securely save my information for 1-click checkout

Figure 159: Stripe Checkout Page.