FOOTEASE LOCAL SHOE SHOPPING DIRECTORY

NORAIESHURA BINTI MUHAMAD KHAIR

UNIVERSITI TEKNIKAL MALAYSI MELAKA

FOOTEASE LOCAL SHOE SHOPPING DIRECTORY

NORAIESHURA BINTI MUHAMAD KHAIR

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Database)] with Honors.

FAKULTI TEKNOLOGI DAN MAKLUMAT KOMUNIKASI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[2024]

DECLARATION

I hereby declare that this project report entitled

FOOTEASE LOCAL SHOE SHOPPING DIRECTORY

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : _____ Date : 6/9/2024

(NORAIESHURA BINTI MUHAMAD KHAIR)

I hereby declare that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Database Management) with Honours.

SUPERVISOR : _____ Date : 6 Sep.2024

([TS. ABDUL RAZAK BIN HUSSAIN])

DEDICATION

To my beloved parents

My mom, Mrs. Hayati Binti Md Isa

My dad, Mr. Muhamad Khair Bin Hussein

To my beloved supervisor, Mr. Ts. Abdul Razak Bin Hussain

To my beloved evaluator, Mrs. Ts. Hidayah Binti Rahmalan

And fellow friends that give me support and sharing knowledge in order to finish my project

ACKNOWLEDGEMENTS

Assalamualaikum. Firstly, thank you and Praise to Allah Almighty Who giving me strength, good health and acknowledged me to complete my Final Year Project (FYP) I. I would like to thank En. Abdul Razak for giving assistant to complete this project successfully

I would also like to thank my beloved parents and my friends who have been giving me support and motivation throughout my project. They were always giving me a huge moral support and passionate encouragement to me and make it possible for me to complete this project.

Lastly, thanks again to all those involved for helping me throughout in completing my final year project. May Allah bless all of your kindness.

ABSTRACT

”FootEase” can be described as a brand-new programme that has the primary role of encouraging locally-based shoe companies as well as recreating the online shoes buying experience. This revolutionary technology was designed to address fundamental problems with the prevailing model of ecommerce, including an inherent search capability; scant information about stores; and no ability to monitor the performance of stores that the platform targets and serves. According to ‘’FootEase’’, there are several ingenious solutions to get round the above obstacles. To begin with, it has relatively advanced search options based on geolocation; thereby, it is easy for the clients to identify shoe selling stores of their interest within their region. This improvement, which connects customers to nearby companies, enhances not only the message’s relevance and value for users, but also supports the concept of community. Moreover, “FootEase” offers complete store details and this eliminates the issues of lack of such details. People can also find out more about what each store brings, specifically, the shoes they carry and ongoing promotions by going to the user profiles. The platform assists the individuals in being informed about the best products to purchase and assists the local businesses in decision-making. It also lays down the groundwork for a perfect performance appraisal system for the other stores so that one may track their relative productivity over time. ‘FootEase’ also provides business owners with tips on how to enhance their business visibility as well as efficiency through the assessment of sales data and other factors. The process of constructing “FootEase” is presented through the following systematic research model: user feedback analysis, the process of developing and refining the module, and the use of technological advancement. Thanks to this rigorous process, the platform has been developed to meet several needs of different stakeholders including administrators, lovers of shoes, and owners of local stores. The fruit of it a very considerable platform helping not only to provide the people with the new type of shoe purchasing experience online, but also supporting societal engagement and local business’ sustainability. Through “FootEase” customers can easily navigate the ever-evolving market of shoe stores without a doubt that they are being provided with crucial services of small shops.

TABLE OF CONTENTS

10

11

LIST OF TABLES

PAGE

12

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
DBMS	**-**	**Database Management System**

LIST OF ATTACHMENTS

PAGE

CHAPTER 1: INTRODUCTION

1.1 Introduction

Online commerce is a dynamic field that offers businesses, particularly small shops, a range of benefits and problems in the current digital era. "FootEase" became a leading project, inspired by a deep admiration for the unique goods that local shoe businesses have to offer. This programme, which understands the need to breathe new life into these businesses' digital presence, has the potential to completely transform the online shoe purchasing experience. Fundamentally, "FootEase" is motivated by the goal of overcoming the gap that exists between customers and small companies, creating a closer bond between communities while honoring the variety of services offered by local shoe retailers.

The idea behind "FootEase" is to change the way people purchase shoes online while simultaneously promoting a sense of community involvement among shoe enthusiasts. Local retailers sometimes find it difficult to compete with larger, more well-established online platforms in today's fast-paced digital industry. By utilizing modern technologies such as geolocation information and real-time updates, this project aims to close the gap between the two by developing a dynamic, user-friendly platform.

Recognising the important contributions that small businesses make to the structure of communities is fundamental to the "FootEase" concept. Through the creation of a digital platform that enhances their visibility and highlights their distinctive products, this initiative seeks to empower small businesses, promote diversity, and strengthen local economies. Furthermore, by putting user experience and security first, "FootEase" provides a safe space where users can interact and discover a variety of local shoe businesses.

"FootEase" is really more than simply a technology advancement; it is an effort to promote community, empower the local economy, and honor local enterprise. This project aims to transform the online shoe purchasing experience by prioritizing diversity and taking a strategic approach, which will keep local shops at the forefront of the digital marketplace.

1.2 Problem Statement

1. The absence of search functionality impedes their ability to explore stores that match their criteria efficiently and users experience challenges in managing their accounts, preferences, and saved information such as favorites,
2. Users often encounter difficulties in accessing comprehensive information about local shoe stores, including current promotions and available shoe brands. Incomplete details hinder their decision-making process and limit their engagement with the platform.
3. Lack of a system to track and assess how well each store performs and the sales they make each year. Without a structured approach, it's challenging to understand and manage the yearly effectiveness of individual stores.

1.3 Objective

1. To implement a search functionality for users to discover shoes based on category, brand, color and allow users to manage their accounts and save information (favorites).
2. To individual pages for each store displaying information about the current promotions, and available shoe brands.
3. To record the performance of stores and the quantity of sales per year based on the store.

1.4 Scope

1.4.1 Target User

a) Admin
b) Shoe Enthusiasts
c) Store Owner

1.4.2 Module to Develop

A) Admin Modules

- User Management Module

  i) Manage user accounts, including account moderation.

  ii) Handle user-related issues.

- Store Management Module

  i) Handle store-related concerns and inquiries.

  ii) Approve promotions submitted by local stores.

- Report Module

  i) View report based on shoe, sales and orders.

B) Shoe Enthusiasts Modules

- User Registration and Authentication Module

  i) Register for an account and log in securely.

  ii) Manage their user profile and preferences.

- Search and Discovery Module

  ii) Utilize filters for shoe attributes such as brand, size, and color.

- Store and Shoe Exploration Module

  i) View individual pages for each store, displaying information about current promotions, and available shoe brands.

- Promotions Section Module

  i) View ongoing promotions and discounts for different shoe stores.

  ii) Access details and terms of each promotion.

- User Accounts and Preferences Module

  i) Manage their user account, preferences, and save information (favorites).

- Shopping Module

  i) Browse and buy shoes directly from the website, providing an additional avenue for shoe enthusiasts to make purchases.

C) Store Modules

- User Registration and Authentication Module

  i) Register for an account and log in securely.

  ii) Manage their user profile and preferences.

- Store Management Module

  i) Manage store details, including address and contact information.

  iii) Submit store details for approval.

  iv) Edit, add, and delete local shoe stores.

- Shoe Management Module

  i) Edit, add, delete and transfer shoes.

  ii) Update inventory and shoe listings.

- Promotions Section Module

  i) Manage store details.

  ii) Edit, add, delete and view promotions based on the store.

- Report Module

  i) View report based on shoe, sales and orders.

1.4.3 Hardware

a) Laptop

b) Wi-Fi

1.4.4 Software

a) Operating System

- Windows

b) Development Frameworks and Tools

- Server: XAMPP
- Back-end: PHP
- Database Management: MySQL, phpMyAdmin

1.4.5 Constraints

a) Internet Connectivity

Users with limited or unreliable internet access may experience difficulties accessing real time data, impacting the overall user experience.

b) Local Business Participation

The success of the system relies on the active participation of local shoe stores. Constraints may arise if local businesses are hesitant to join the platform, submit accurate information, or engage with the interactive features. Building trust and incentivizing participation will be essential.

c) Device Compatibility

While efforts will be made to ensure the website's responsiveness on various devices, the system may face constraints related to compatibility with older devices or certain web browsers.

a) Web-Based System: Able to be accessed using common web browsers, guaranteeing cross-platform compatibility.
b) Database Management System: provides effective use of a powerful database management system to store and retrieve information about shoe retailers, inventories, sales, and user interactions.
c) Integration of Geolocation: uses geolocation technology to give users routes and distances to shoe stores in the area. employs geographical settings to deliver customized search results, improving the user experience.

1.6 Conclusion

"FootEase" promotes the wellness of local stores while working to transform the online shoe purchase experience. With the use of modern technology and community involvement, the initiative hopes to close the distance between customers and local shoe businesses. By focusing the user and enabling them, "FootEase" not only improves the shopping experience but also encourages financial sustainability and highlights the variety of small companies. "FootEase" is a monument to the innovative power of creativity and the ongoing significance of promoting local business in the digital era as we move forward.

CHAPTER 2: LITERATURE REVIEW AND PROJECT

METHODOLOGY

2.1 Introduction

Creating a database system involves gathering requirements, assessing them in relation to the system's functionality and data architecture, and then putting each function into each of its modules. When the Database Life Cycle approach is used, it guarantees that the system is sufficiently prepared before deployment. Further details regarding this phase are elaborated upon in this chapter.

2.2 Database Development Methodology

a) Planning

System analysis helps break down the current system into its parts to help us reach our project goals. We're looking at the design of the "FootEase Local Shopping Directory" to see what's good and what needs improvement. In this chapter, we're drawing diagrams that show how the old system and the new one will work together. These diagrams, like the Context Diagram (CD) and Data Flow Diagram (DFD), give us a clear picture of how the system operates.

b) Analysis

During the analysis phase, we carefully examine the issues outlined in the problem statements and define our objectives clearly. We identify key elements like Users, Stores, Promotions, and Reviews, and understand how they relate to each other. For example, Users interact with Stores by leaving Reviews and engaging with Promotions. We also collect requirements for each group involved, such as what Admins, Shoe Enthusiasts, and Store Owners need from the platform. Additionally, we figure out what specific 24

information we need to store for each entity and module. For instance, Users might have attributes like username and password, while Stores might have

details like location and contact information. Lastly, we look into potential security and privacy issues to make sure the system keeps everyone's information safe.

c) Design

In the design phase, we create a blueprint for the database using diagrams like Entity-Relationship Diagrams (ERDs) or Unified Modeling Language (UML) diagrams. These diagrams help us visualize how different entities, like Users and Stores, are connected to each other. We then normalize the database, which means we organize the data to minimize redundancy and ensure that information is consistent and accurate. This involves breaking down data into smaller, more manageable parts and linking them together efficiently. We define tables for each entity, specifying the fields (like name, address, etc.) and establish primary keys to uniquely identify each record. Foreign keys are used to establish relationships between tables. We also determine the appropriate data types for each field, such as text, numbers, or dates, to ensure data consistency and efficiency. Additionally, we design views and stored procedures to handle complex queries and operations, making it easier to retrieve and manipulate data as needed.

d) Implementation

In the implementation phase, we bring our design to life by creating the actual database structure based on the blueprint we've designed. This involves setting up tables, defining relationships between them, and establishing constraints to ensure data integrity. We select a suitable database management system (DBMS), such as MySQL, PostgreSQL, or MongoDB, to store and manage our data efficiently. Once the database is set up, we implement security measures to protect sensitive information. This includes implementing authentication to verify user identities, authorization to control access to certain data or functionalities, and encryption to safeguard data 25

from unauthorized access or theft. Additionally, we may write scripts or use Object-Relational Mapping (ORM) frameworks to interact with the database, making it easier to perform operations like data retrieval, insertion, and updates. These measures help ensure the system's reliability and security while providing a robust foundation for the FootEase platform.

e) Testing

In the testing phase, we ensure the FootEase platform operates smoothly and meets user expectations. First, we develop test cases based on the requirements and anticipated user interactions. This helps us verify that each aspect of the platform functions as intended. We conduct unit testing to examine individual modules in isolation, checking that they perform their specific tasks correctly. Integration testing follows, where we test how these modules interact with each other to ensure seamless operation of the entire system. Additionally, we test for performance, scalability, and data integrity to ensure the platform can handle a large number of users and transactions without compromising speed or reliability. Security testing is also crucial, where we identify and address vulnerabilities to protect user data and prevent unauthorized access. Through thorough testing, we aim to deliver a robust and reliable FootEase platform that provides a seamless and secure experience for users and stakeholders alike.

f) Project Conclusion

As we wrap up the project, the application is deployed to a live environment where users can access it. To ensure smooth operation, training sessions and comprehensive documentation are provided for administrators, shoe enthusiasts, and store owners. After the launch, we closely monitor the system's performance and gather feedback from users. Based on this feedback and any changing requirements, we make necessary improvements and updates to the application. Finally, we evaluate the project's success against the predefined objectives and compile a final report summarizing the development process and outcomes, officially closing the project.

2.3 Database Development Project Requirement

2.3.1 Software Requirements

1. Development Tools:

- Server: XAMPP
- Development Environment: Microsoft Visual Studio Code Programming

Languages:

- PHP
- HTML
- CSS
- JavaScript

Database Management:

- MySQL
- phpMyAdmin

Tool for development in this system is XAMPP is a versatile, cross-platform web server solution stack package that includes Apache, MariaDB, PHP, and Perl, making it ideal for local development and testing of web applications. It has a control panel for component management, is easy to use, and is appropriate for newbies. Lightweight and open-source, Microsoft Visual Studio Code (VS Code) is a powerful code editor that includes features like Git support, an integrated terminal, and IntelliSense. It has extensions for a wide range of tools and computer languages, making it extremely configurable. When combined, XAMPP and VS Code offer a potent development environment that facilitates effective web development processes through local testing, debugging, and coding.

PHP, HTML, CSS, and JavaScript are necessary for web development. PHP is a server-side programming language that's perfect for building dynamic websites and apps since it can connect with databases and produce dynamic content. Web pages and information are organized using HTML (HyperText Markup Language), which serves as the basis for web development. Web developers may style HTML components with fonts, colors, and spacing thanks to CSS (Cascading Style Sheets), which also governs how web pages look and are laid out. JavaScript is a programming language that runs on the client side and improves user interaction and experience on websites by allowing animations, asynchronous server connection, dynamic content changes, and form validation. When combined, these languages allow for the development of interactive, aesthetically pleasing, and useful websites and applications.

For database management, I use MySQL and phpMyAdmin. A open-source relational database management system (RDBMS) for managing, storing, and retrieving data in web applications is MySQL. It is perfect for managing huge quantities of data effectively since it has strong data security features, allows complex queries, and transactions. A web-based application called phpMyAdmin makes MySQL database management jobs easier. For tasks including building and maintaining databases, tables, and users, running SQL queries, and importing or exporting data, it provides an easy-to-use graphical user interface. I can efficiently manage and work with my databases with the help of phpMyAdmin and MySQL, which guarantees top speed and user-friendliness for my web development projects.
28 **2. Operating System: Microsoft Windows 10 Database,**

The operating system chosen for this system is Microsoft Windows 10, primarily due to its compatibility with the hardware tools. Windows 10, a part of the Windows NT family, is favored for its familiarity to users of previous Windows versions like 7 and 8, ensuring an easy transition. It also boasts quick startup times, enhancing productivity.

3. Other Software: Microsoft Office Word, Canva, Draw.io

The additional software for this system includes Microsoft Office Word, Canva, and Draw.io. Microsoft Word is used for creating, editing, and formatting documents, making it ideal for proposals and reports. Canva is a graphic design platform that facilitates collaboration on visual content creation, making it suitable for designing posters and presentation slides collaboratively. Draw.io is a diagramming tool used for creating various diagrams, such as flowcharts for visualizing processes, mind maps for brainstorming, network diagrams for mapping out infrastructure, and UML diagrams for modeling software systems.

2.3.2 Hardware Requirements

Hardware Requirements: Laptop

Effective group project systems heavily rely on laptops. Their portability, versatility, and affordability render them indispensable tools for facilitating collaborative activities. Laptops are essential for launching development tools, software applications, and databases, enabling seamless project execution.

Network Requirements: Wi-Fi

The FootEase Local Shopping Directory relies on network necessities like Wi-Fi. Wi-Fi stands as the leading wireless networking technology globally, utilized in various settings like homes, businesses, schools, and public areas. It enables internet access and resource utilization without the constraints of cables, enhancing convenience for users of portable devices such as laptops, tablets, and smartphones. Additionally, an internet connection is essential for the database used to efficiently retrieve and maintain data.

2.4 Project Management

This section details the management of the project from its initiation to completion. Subsection 2.4.2 provides further insight into the specific milestones set for the gradual progression of the project, offering additional information on measurable goals and checkpoints defined for each stage of development. It elaborates on the detailed objectives and specific checkpoints established to track the project's evolution systematically, ensuring that progress is monitored and aligned with the project's overall objectives.

2.4.1 Project Milestones

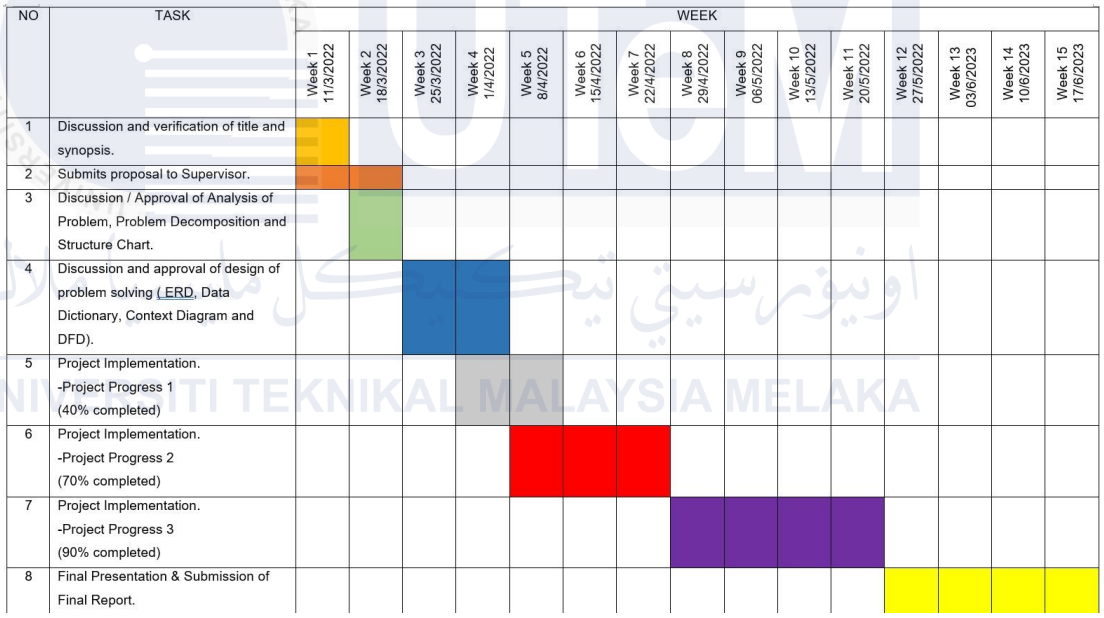


Figure 2.1 Gantt Chart

Gantt Chart has been decided to be used, as seen in Figure 2.3.1.1 above, to efficiently and successfully manage the project. This graphic tool made it simple to monitor and track progress across the many stages. The project began with a thorough PSM 1 briefing that was led by , Ts. Abdul Razak Bin Hussain. This meeting provided an essential basis, guaranteeing that every student had a thorough comprehension of the complexities associated with PSM 1. A unanimous consensus 31

was reached to begin developing the "FootEase local Shopping Directory." This conclusion resulted from Ts. Abdul Razak Bin Hussain.

During the first week, I was actively involved in creating the system proposal. This important paperwork set the direction for the project. I worked hard to determine the needs and specifications for the FootEase Local Shopping Directory throughout this phase and to start drawing system designs to present to Supervisor Ts. Abdul Razak Bin Hussain for insightful feedback. After the proposal was finished and turned in to the supervisor, I eagerly awaited feedback and got approval. The idea was finally accepted, even though it needed a few modifications. The supervisor's insightful comments were really helpful.

For designing software systems, outline the steps like creating flowcharts, pseudocode, Data Flow Diagrams (DFD), Entity-Relationship Diagrams (ERD), and a Data Dictionary. Flowcharts map out processes visually, while pseudocode helps clarify logic before coding. DFDs visualize data movement, ERDs show data structures, and the Data Dictionary defines data elements. Tasks are scheduled considering dependencies. For example, DFD and ERD might precede flowcharts and pseudocode development, while the Data Dictionary is developed alongside them for consistency.

The process begins with gathering requirements to understand what the database needs to achieve. Then, design the database structure, ensuring it's organized and efficient. Once the analysis is done, move on to implementing the database system. This involves setting up the database software on the appropriate hardware, creating the database schema, and migrating data if needed. It also can optimize the system for better performance, implement security measures to protect data, and establish backup and recovery procedures.

For database implementation and web development, the planning the steps to build a web application with a backend database. For the database part, it starts by designing the database structure, setting it up on a server. Then, optimize it for speed,

add security measures, and set up backups. As for web development, design the look
32

and feel of the app, build the behind-the-scenes functionality, and make sure everything works well together. Testing will ensure everything functions as expected before deploying it to users. Once live, maintain and update the app as needed. Tasks are scheduled considering their order and how they depend on each other, ensuring a smooth project flow.

In the process of preparing the final report draft, start by planning the report's objectives, gathering all necessary information, and outlining its structure. Then, during the writing phase, craft the report, including an introduction, main findings, analysis, and conclusions, supported by data presentation and discussions. After writing, review and revise the draft, proofreading for errors, incorporating feedback, and making necessary revisions. Once the content is polished, finalize the report by formatting it correctly, conducting a final proofread, gaining approval, and determining distribution methods. Throughout these stages, a Gantt chart helps in scheduling tasks, managing dependencies, and ensuring the report is completed efficiently and on time.

For the final presentation of a system, plan various stages to ensure a successful showcase of the completed software. Initially, focus on preparing the content, including creating slides, demos, and documentation, followed by rehearsals to ensure everyone's ready. Next, polish the system by refining it and conducting user testing for feedback. Then, handle logistics like securing a venue and setting up equipment. On the presentation day, run final checks, set up, deliver the presentation, and hold a Q&A session. Afterward, collect feedback and decide on any follow-up actions needed. This organized approach helps ensure everything runs smoothly and the system is presented effectively.

Scheduling the submission of the final report and logbook involves planning the last steps of a project, where documentation is key. For the final report, gather project details, write and format the report, review and revise it, and get any necessary

approvals before submitting it by the deadline. Similarly, for the logbook, compile project entries, organize them, review for accuracy, format them appropriately, seek approvals if needed, and submit alongside the final report. Each 33

task is allocated time on the Gantt chart, considering dependencies and ensuring both documents are completed and submitted on time.

2.5 Conclusion

In conclusion, the process of creating a database system entails a meticulous journey from requirement gathering to deployment, ensuring that every aspect of functionality and data architecture is carefully considered. By employing the Database Life Cycle approach, we ensure thorough preparation of the system before deployment, minimizing risks and maximizing efficiency. This chapter has provided a comprehensive overview of the importance of this phase in the database development process, emphasizing the need for meticulous planning and execution. Moving forward, a thorough literature review and a robust project methodology will serve as foundational pillars to guide us through the intricacies of database development, ultimately leading to the successful implementation of our project.

CHAPTER 3: ANALYSIS

3.1 Introduction

System analysis helps break down the current system into its parts to help us reach our project goals. We're looking at the design of the "FootEase " to see what's good and what needs improvement. In this chapter, we're drawing diagrams that show how the old system and the new one will work together. These diagrams, like the Context Diagram (CD) and Data Flow Diagram (DFD), give us a clear picture of how the system operates.

3.2 Current System Analysis

1. **Zappos**

One of the biggest online retailers, Zappos is well-known for having a huge selection of clothes, accessories, and shoes. The website provides a wide array of products for men, women, and children, according to a variety of interests and preferences. The products include formal and casual footwear, stylish attire, bags, accessories, and cosmetic products. Zappos' broad range of brands, which includes both well-known and up-and-coming labels, is one of its standout qualities. This guarantees that customers have access to a multitude of selections to fit their budget and style preferences. The user interface of the website is highly intuitive, making it easy for customers to browse through various categories, filter items based on attributes like size, color, brand, and price, and make well-informed judgements about what to buy.

One of Zappos' most notable features is its customer-focused practices, which include free shipping and returns and provide users more freedom and ease when making purchases online. In keeping with its dedication to client

happiness, Zappos also offers outstanding customer service that is available 24/7 by phone, email, and live chat, guaranteeing that support is always close at hand. Additionally, the website's review and rating 35

system, which enables users to share their opinions and product experiences and help others make decisions about what to buy, promotes a feeling of community and trust.

Customers may easily buy on using Zappos' mobile app that is available for iOS and Android smartphones, in addition to its user-friendly website. In addition, Zappos displays its commitment to sustainability and social responsibility by putting policies in place to lessen its environmental effect and help underserved areas. Zappos has established itself as a top online shoe and clothes store, drawing in a huge global client base because of its wide range of items, outstanding customer support, and dedication to social responsibility.

2. Vionic Shoes

Vionic is a well-known footwear brand that combines fashion with mechanical stability technology to improve the health of your feet. Their website features a wide selection of shoes for men and women, from dress shoes, sandals, boots and slippers to athletic shoes and casual sneakers. What makes Vionic unique is that they incorporate innovative biomechanical features into each design, such as arch support, cushioning and stability elements, all of which are designed to encourage proper foot alignment.

Customers may buy with ease on the Vionic website because to its user-friendly features, which make it simple to explore the many footwear categories. High-quality photos and thorough product descriptions offer a thorough understanding of every item, and filters let customers focus their search results based on things like size, color, and style. By helping consumers find the ideal fit based on their own measurements and preferences, the Fit

Finder tool further improves the shopping experience by guaranteeing the best possible comfort and support.

Apart from offering a wide variety of products, Vionic's website functions as an important center of resources for foot health and wellness. 36

Customers may make knowledgeable decisions regarding their foot health with the help of informative articles, blog posts, and videos that provide information on common foot disorders, how to properly take care of their feet, and the advantages of supportive footwear. Additionally, client ratings and reviews offer insightful commentary and approval, promoting openness and confidence between the company and its customers.

Beyond only conducting business, Vionic actively promotes relationships with the community by posting on social media and participating in online discussion boards. In order to develop a welcoming and inclusive community of Vionic enthusiasts, customers are invited to interact with like-minded individuals, ask questions, and share their experiences.

All things considered, Vionic is a shining example of innovation and wellbeing in the footwear sector, providing not just fashionable and supportive shoe alternatives but also a wealth of information and a lively community centered around foot health and wellness. With its dedication to biomechanical perfection, customer-focused strategy, and educational programmes, Vionic keeps enabling people to move with comfort and confidence at every step.

3. Foot Locker

Foot Locker is a well-known source for sports clothing, accessories, and footwear. It has a wide selection of items suitable for men, women, and kids. Foot Locker offers a wide range of products to accommodate different sports and hobbies, ranging from lifestyle wear and training gear to basketball sneakers and running shoes. Foot Locker's wide range of brands, which

includes well-known names like Nike, Adidas, Jordan, Puma, Reebok, and more, is what makes it stand out. Customers can remain on the leading edge of athletic fashion thanks to this wide variety, which guarantees they have access to the newest trends and innovations from their favorite companies.

Apart from its wide array of products, Foot Locker is widely recognised for its exclusive releases and partnerships, which create a lot of excitement in the sneaker scene. The attraction of Foot Locker as a center of sneaker culture is enhanced by these limited-edition releases, which frequently include distinctive designs and collaborations with sports, singers, and artists. In addition, the website acts as a forum for promoting sneaker culture through blog entries, films, and interviews. It interacts with sneakerheads all around the world and offers insights into the newest arrivals and trends.

Foot Locker puts the needs of its customers first by offering a flawless online shopping experience. Customers may simply explore different categories, search for specific goods, and simplify their results depending on size, color, brand, and price range thanks to the website's user-friendly navigation and filtering capabilities. To help increase customer anticipation and enthusiasm, Foot Locker also has a release calendar that lets users know when highly anticipated sneaker launches are happening.

Foot Locker provides the FLX rewards programme, which allows members to earn points for each purchase made, as a way to reward loyal customers. Loyal clients may exchange these points for discounts, special deals, and admission to events, creating a sense of gratitude and community. Additionally, Foot Locker's dedication to providing excellent customer service is seen by the range of support channels it offers, including as phone, email, and live chat, which guarantee that help is always available for any questions or issues.

All things considered, Foot Locker is a one-stop shop for sports clothing and footwear, offering a wide range of products, first access to new releases, and interesting content to satisfy the various demands and interests of sneakerheads, athletes, and everyday consumers. Foot Locker is a leading authority in the sports retail sector because of its constant innovation, dedication to customer happiness, and strong roots in sneaker culture.

3.3 System To-Be Analysis

3.3.1 Context Diagram

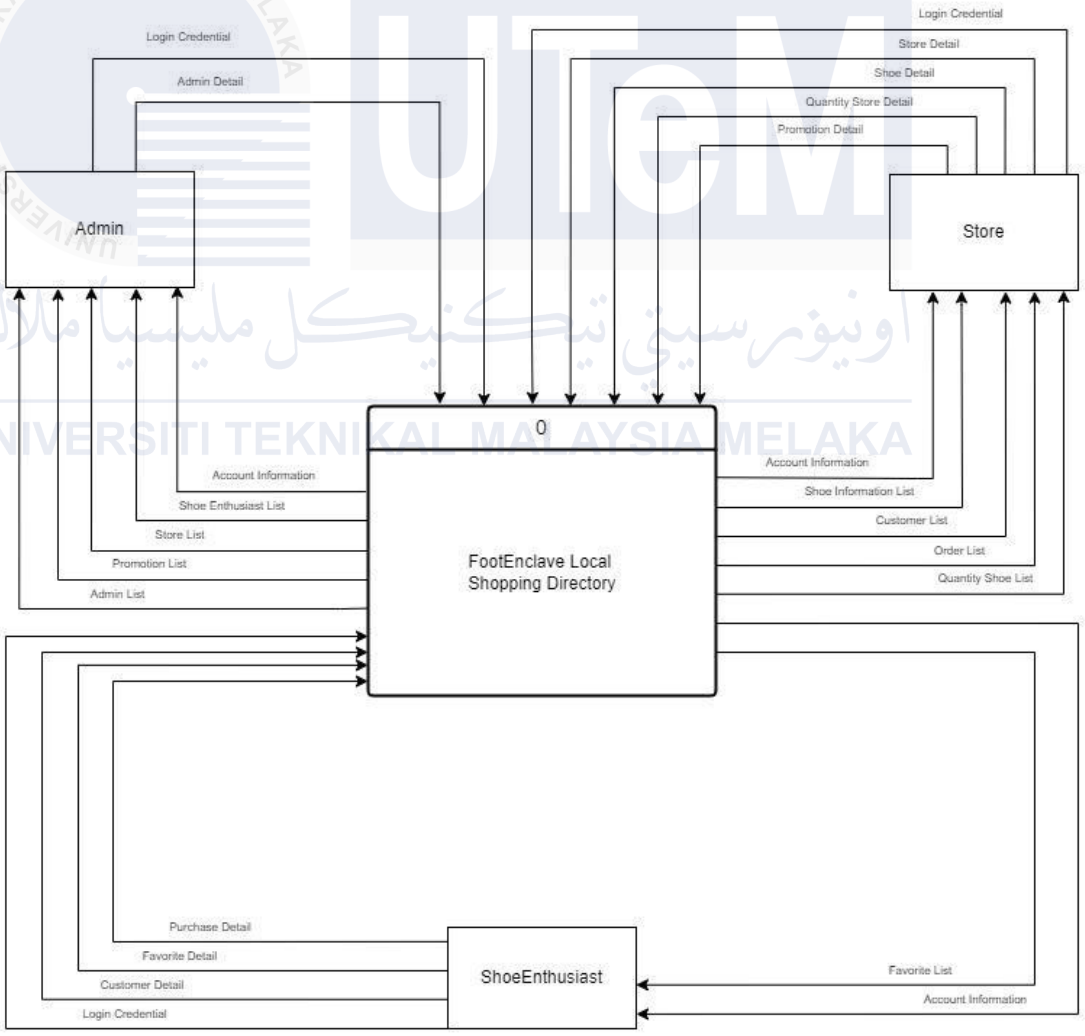


Figure 3.1 Context Diagram

39 **3.3.2 Data Flow Diagram**

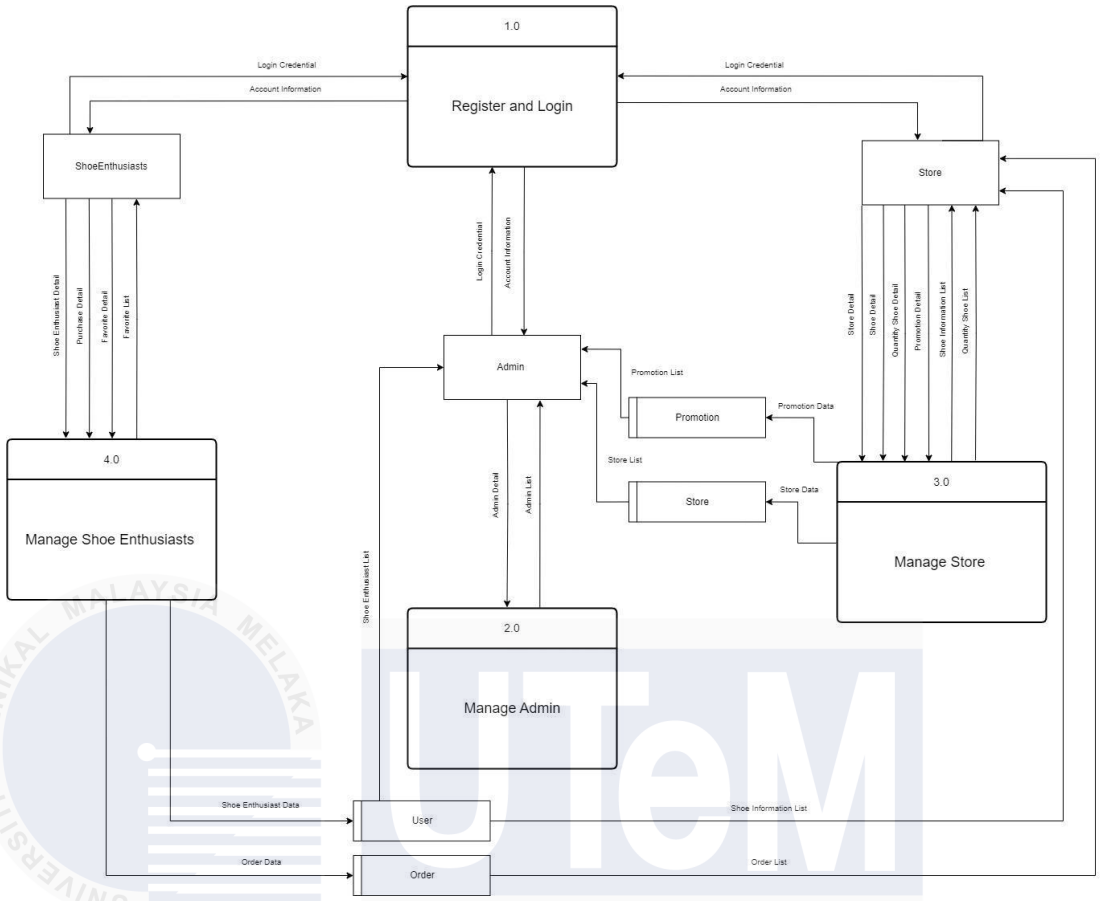


Figure 3.3 Data Flow Diagram

3.4 Conclusion

In conclusion, system analysis plays a crucial role in dissecting the existing system into its components to facilitate the achievement of our project goals. By examining the design of the "FootEase Local Shopping Directory," we identify areas of strength and areas needing improvement. Throughout this process, we create diagrams such as the Context Diagram (CD) and Data Flow Diagram (DFD) to illustrate the integration of the old and new systems. These visual representations offer clarity on how the system functions and how enhancements can be implemented. Through this analysis, we gain valuable insights that guide us in refining the system to better meet user needs and project objectives.

CHAPTER 4: DESIGN

4.1 Introduction

The proposed system design aims to enhance the relationship between shoe enthusiasts and nearby retailers while providing system administrators with effective management capabilities. Its features are customized for the demands of users like administrators, hobbyists, and store owners can utilize different versions of it. The system seeks to improve communication between all parties involved, streamline procedures, and boost user involvement through the meticulous development and customization of modules for each group. With administrators able to effortlessly manage accounts and information, the whole idea is to make shoe purchasing more thrilling and convenient. In order to maintain continuity and transparency, the system places a high priority on unambiguous communication among administrators, enthusiasts, and business owners.

4.2 Flowchart

4.2.1 Main Flowchart of the System

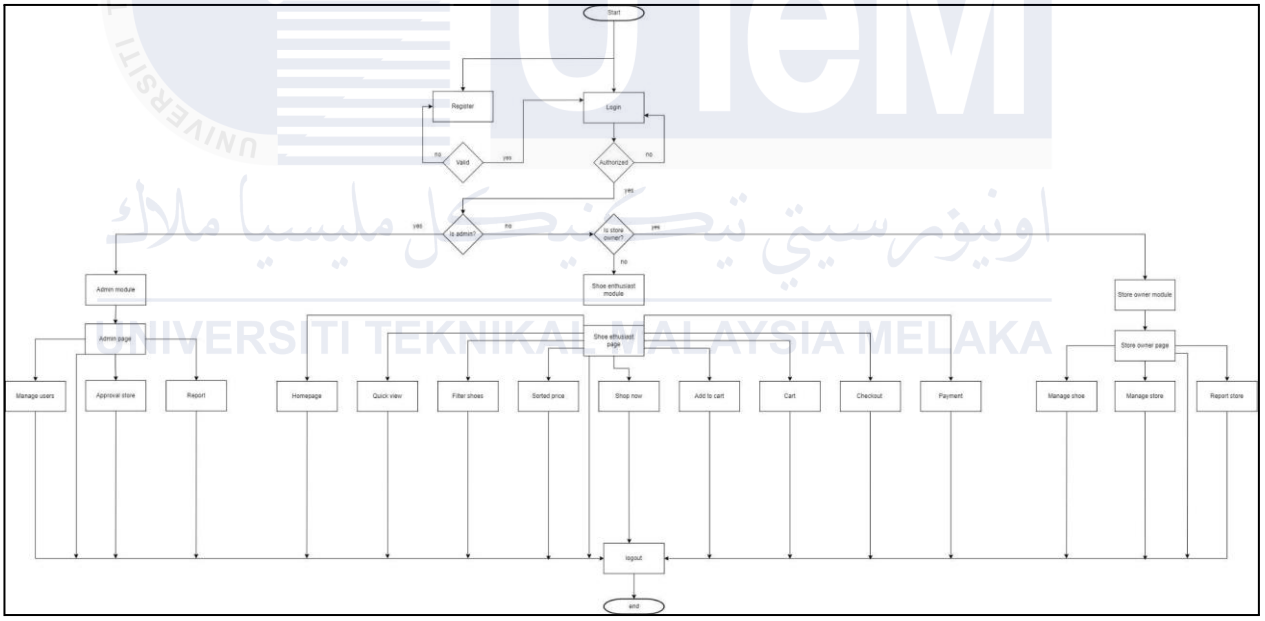


Figure 4.1 Main System

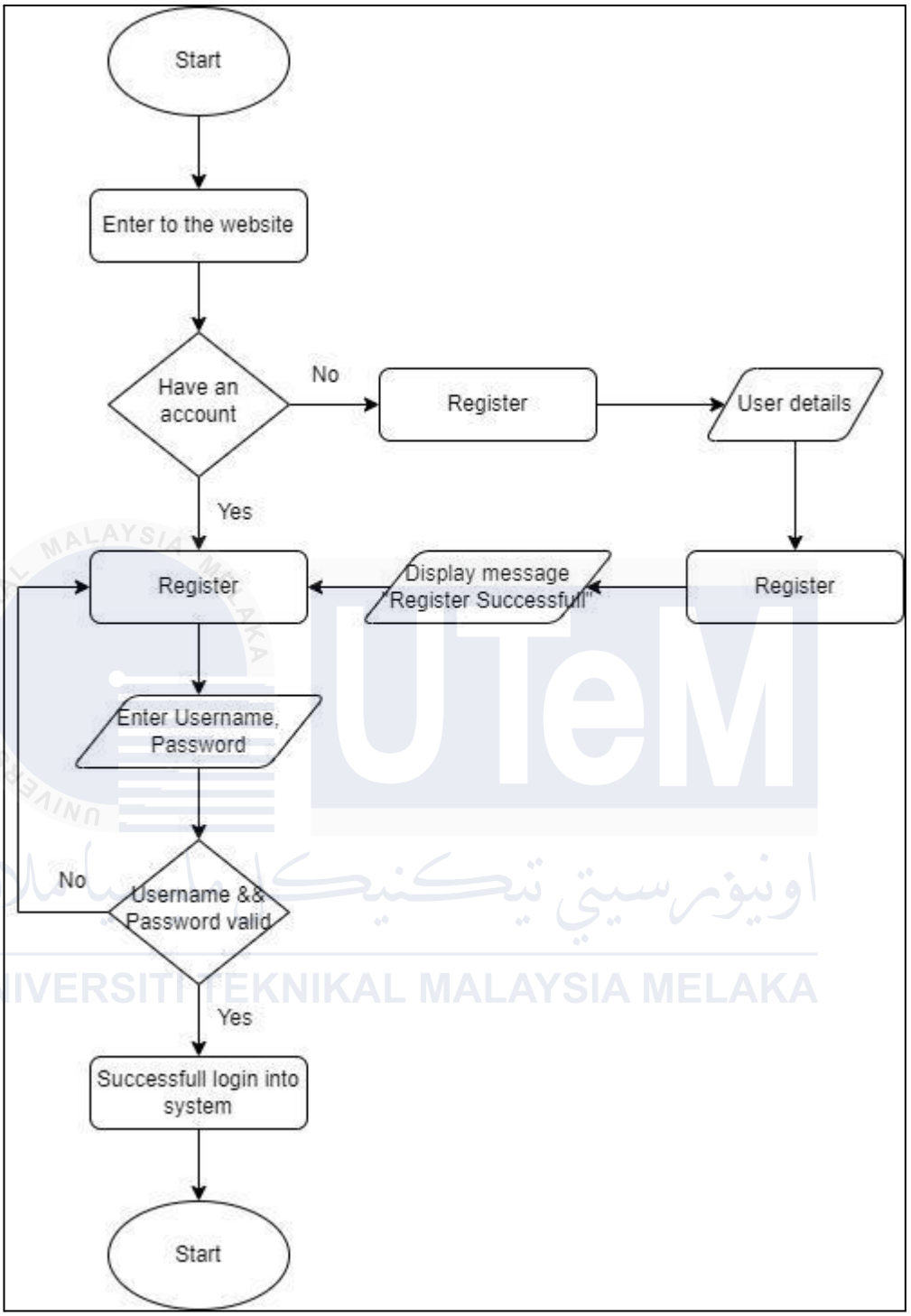


Figure 4.2 Registration and Login

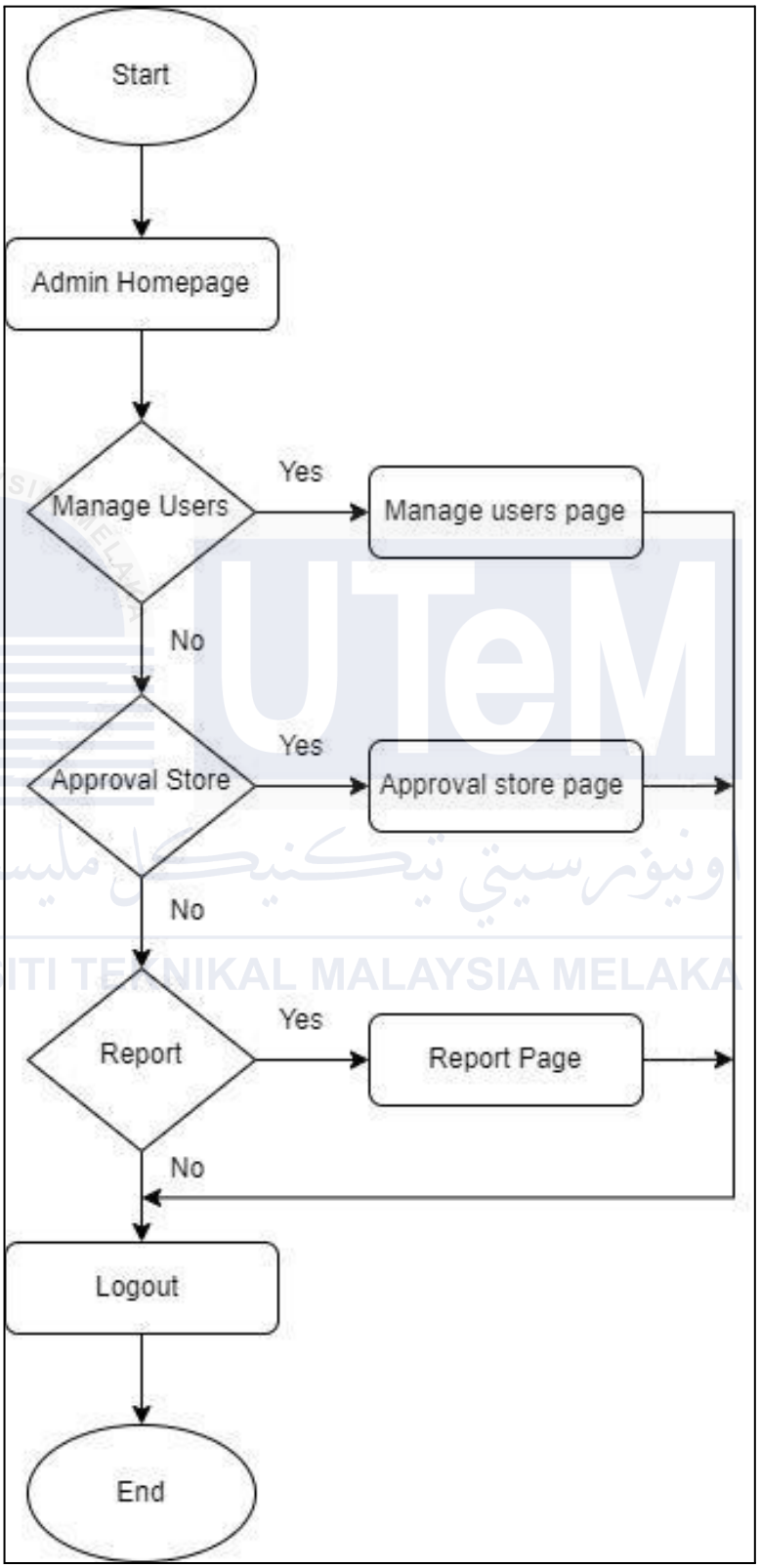


Figure 4.3 Admin Homepage

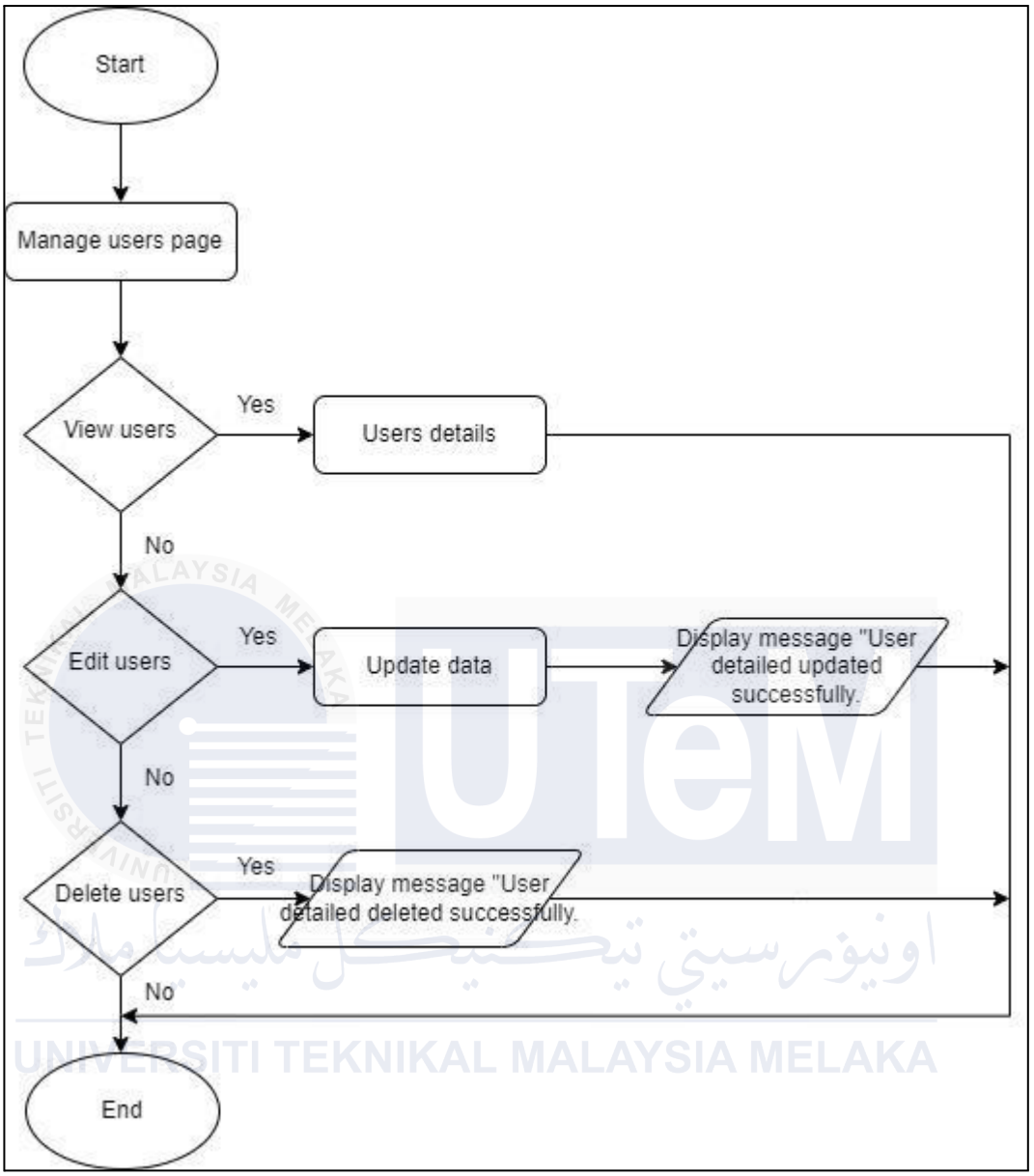
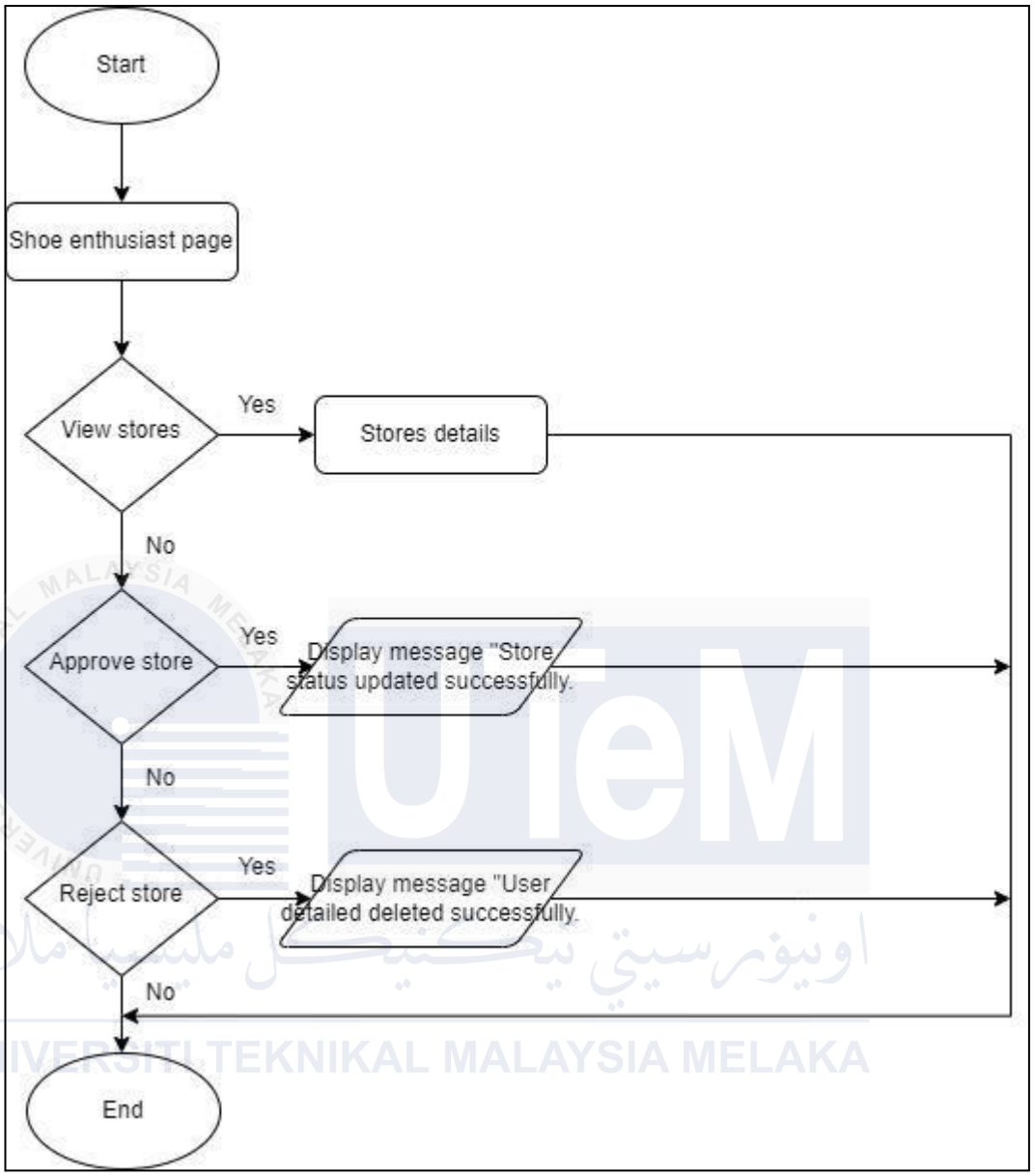


Figure 4.4 Manage User



45

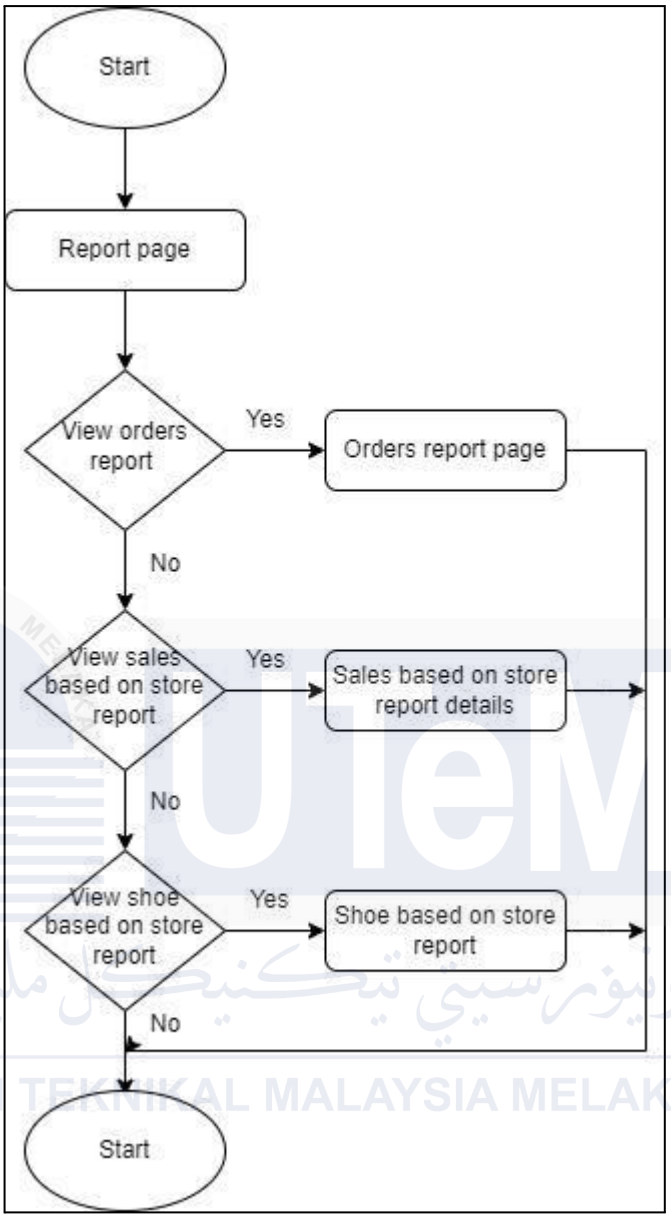


Figure 4.6 Report Admin

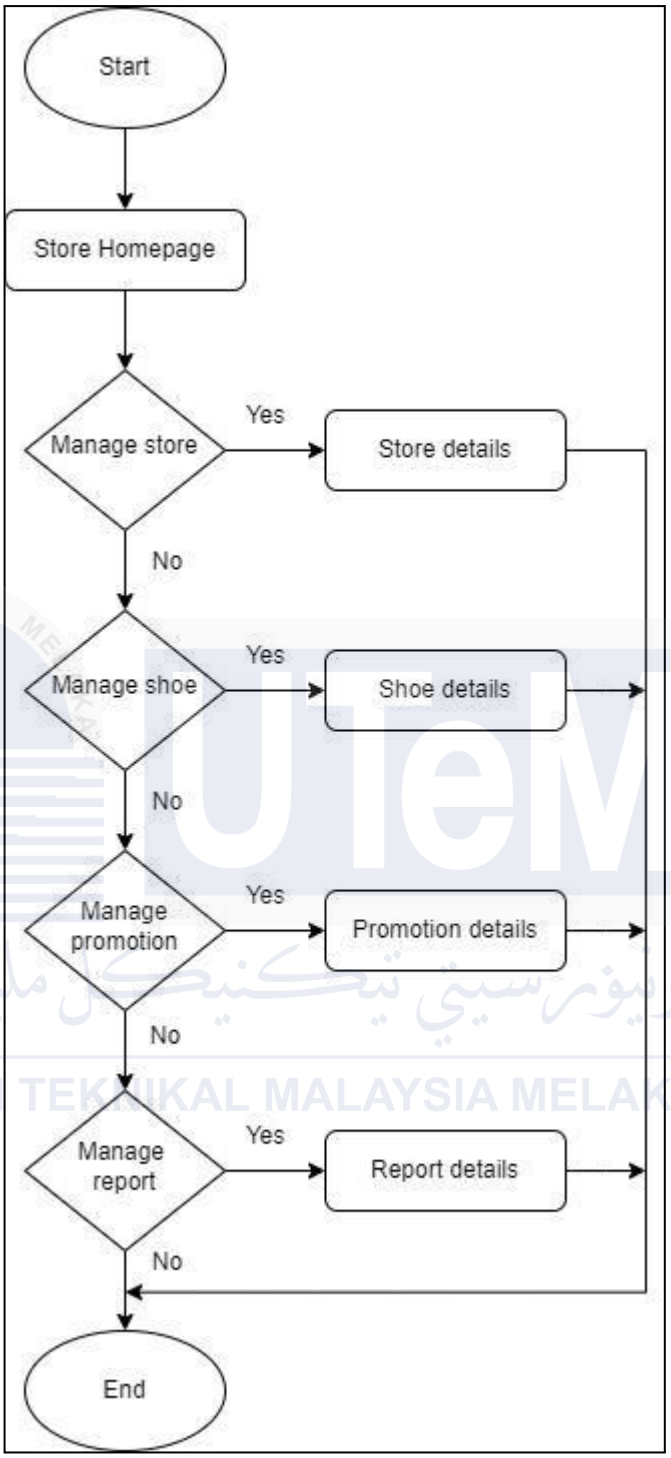
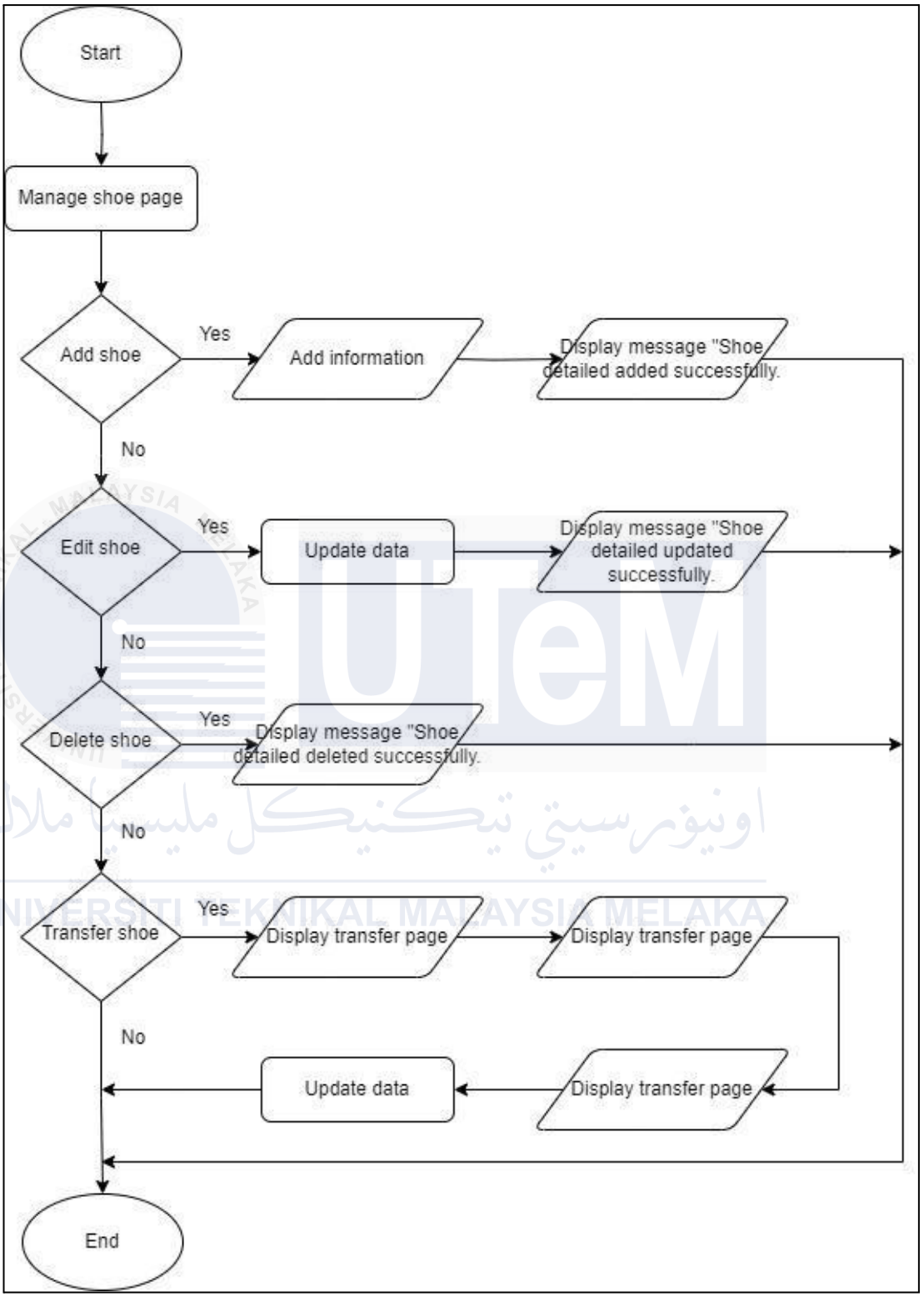
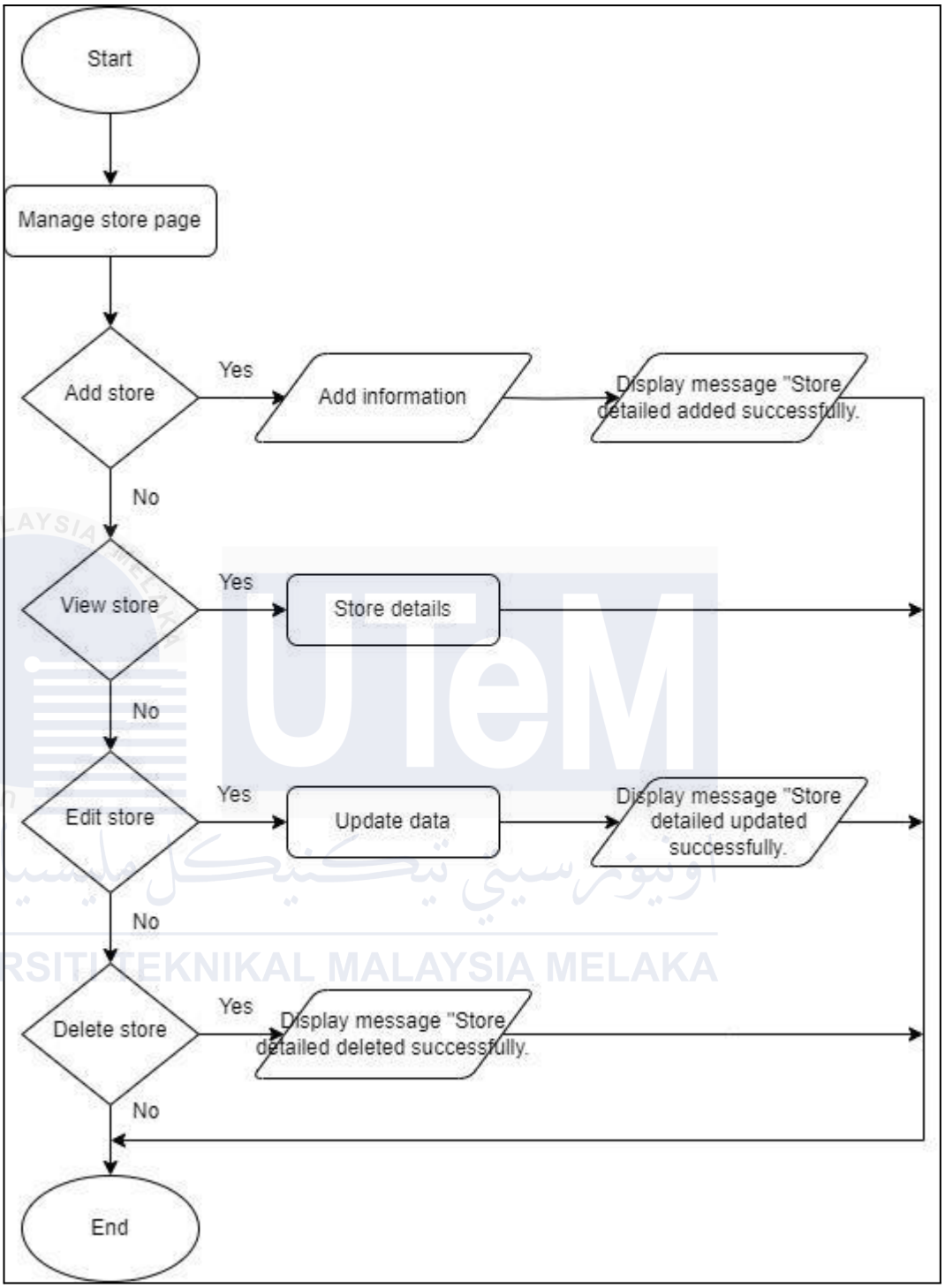


Figure 4.7 Store Owner Homepage



48



49

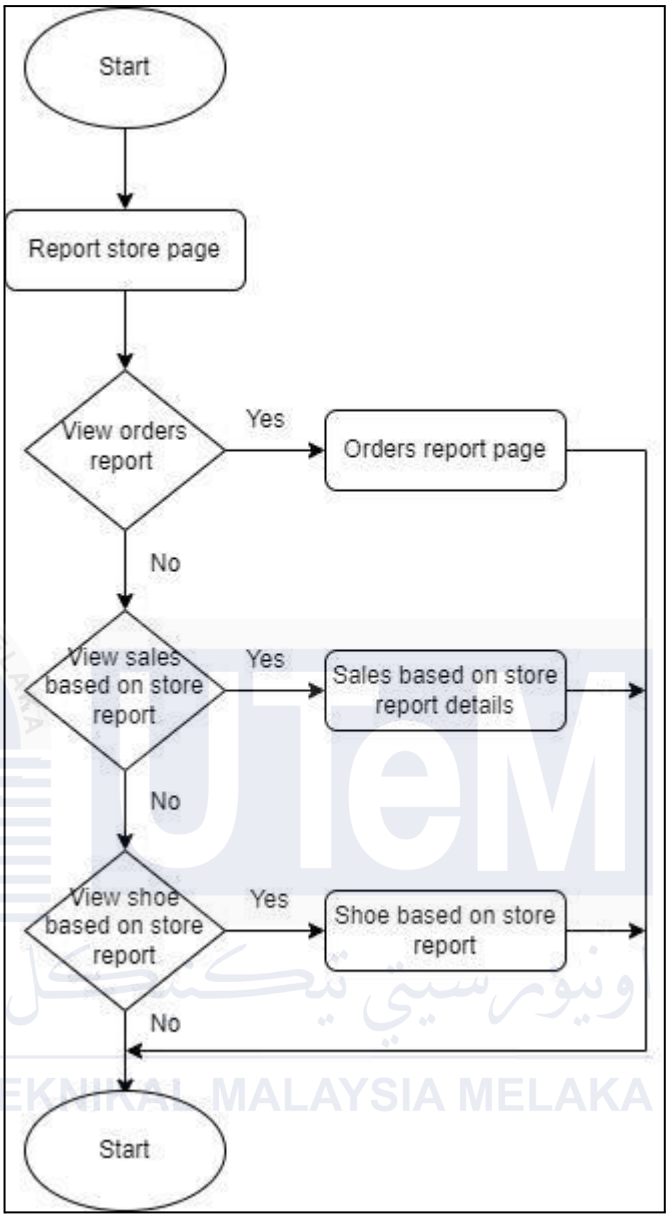


Figure 4.10 Report Store

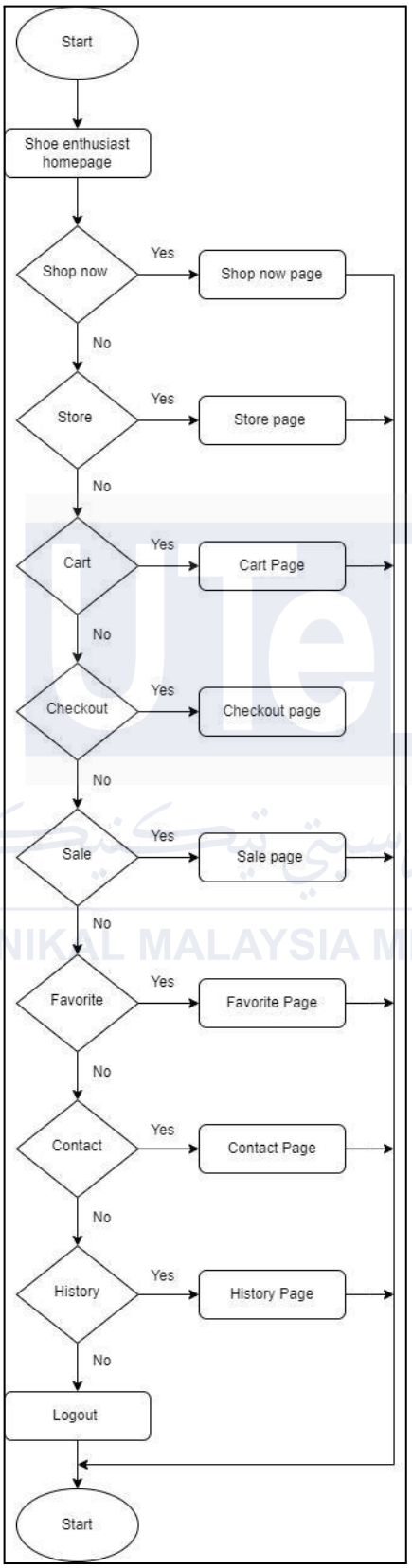


Figure 4.11 Shoe Enthusiast Homepage

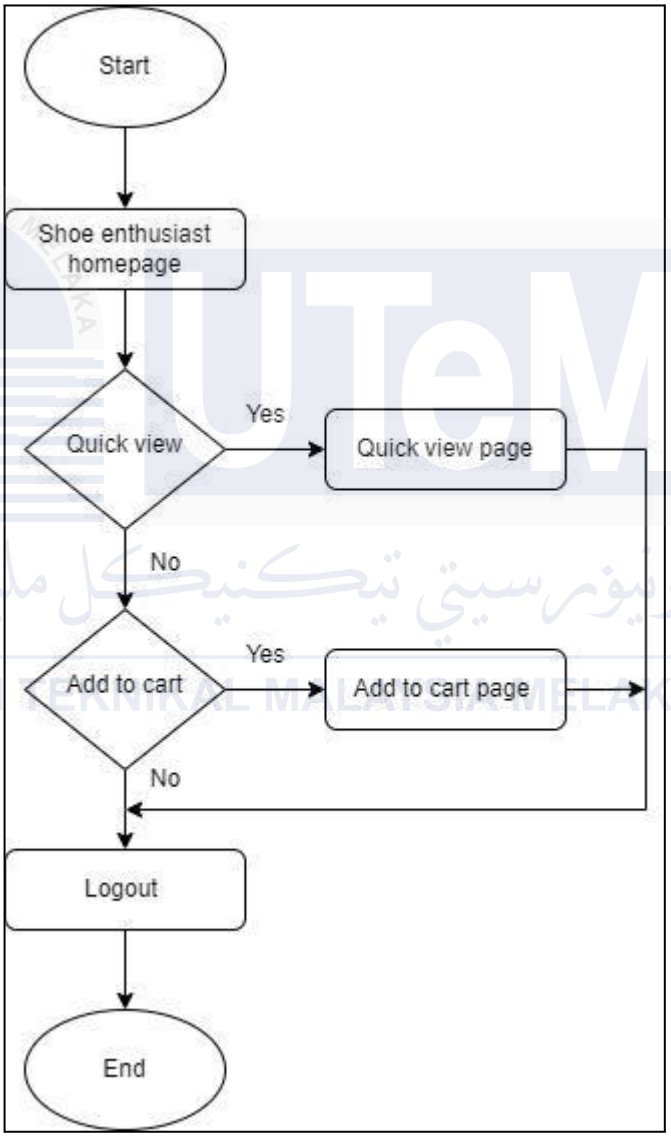


Figure 4.12 Quick view and Add to Cart

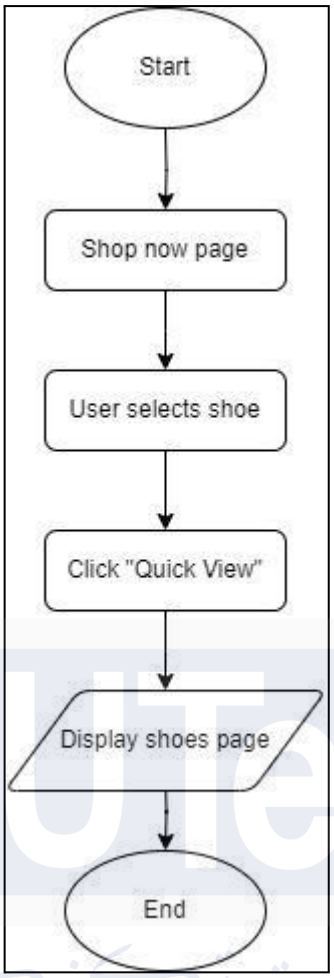


Figure 4.13 Shop Now

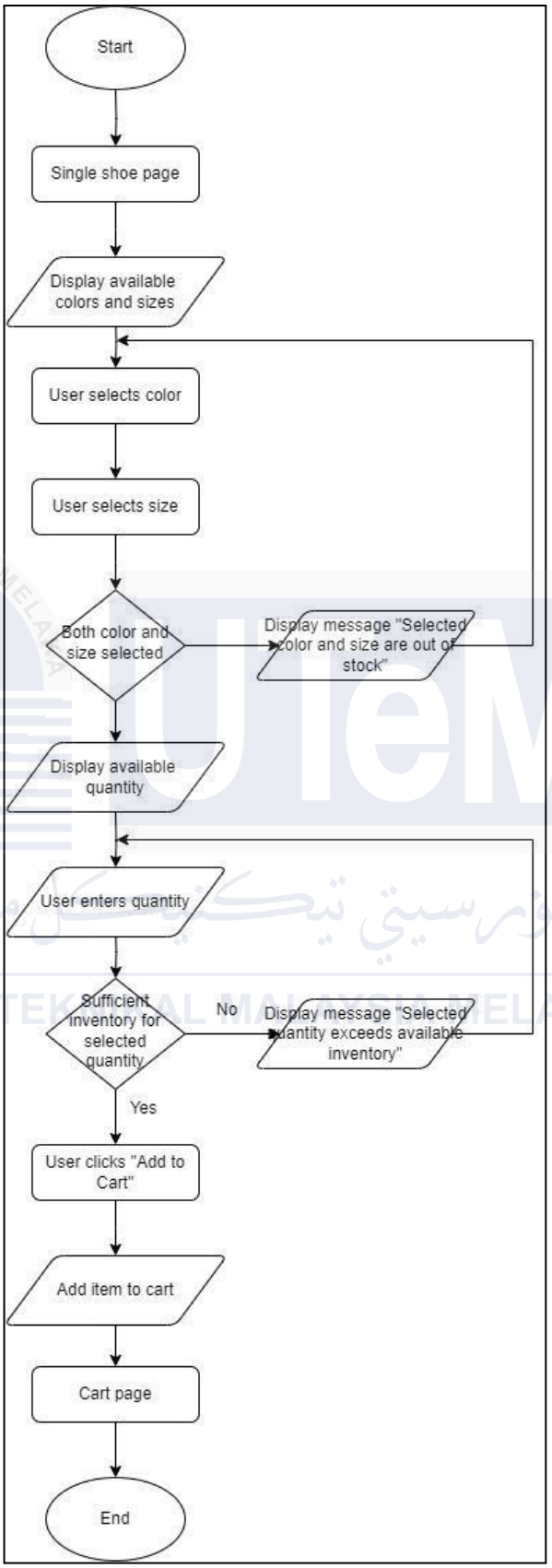


Figure 4.14 Details Shoe

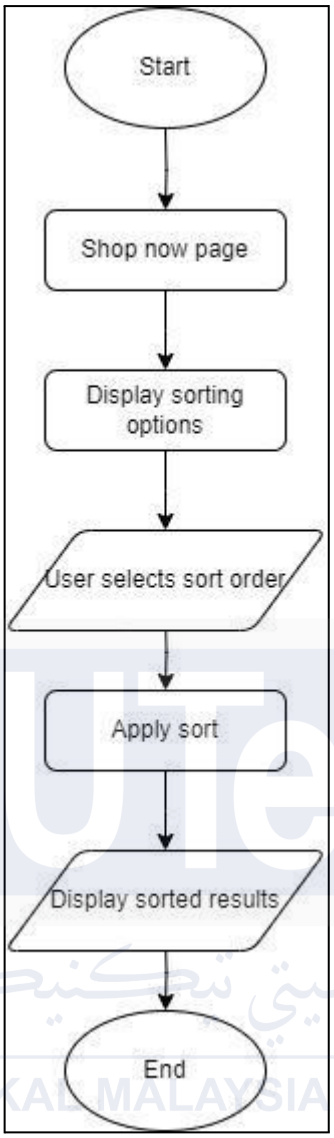


Figure 4.15 Sort Price Shoes

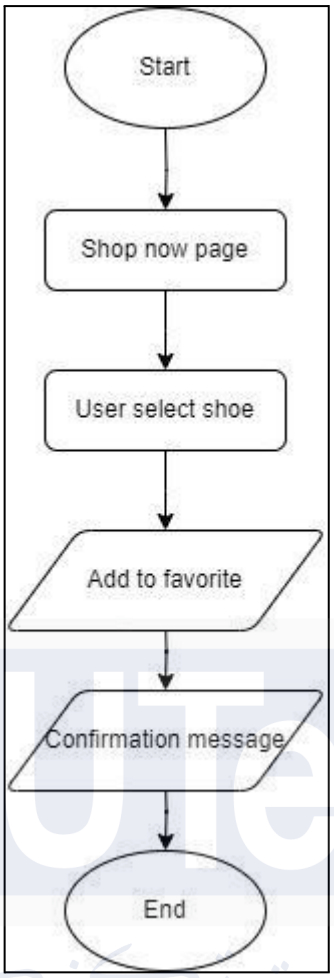


Figure 4.16 Add to Favorite

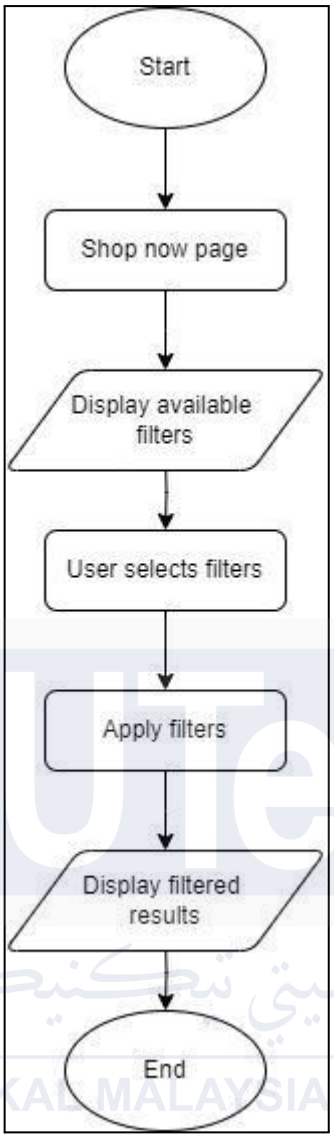
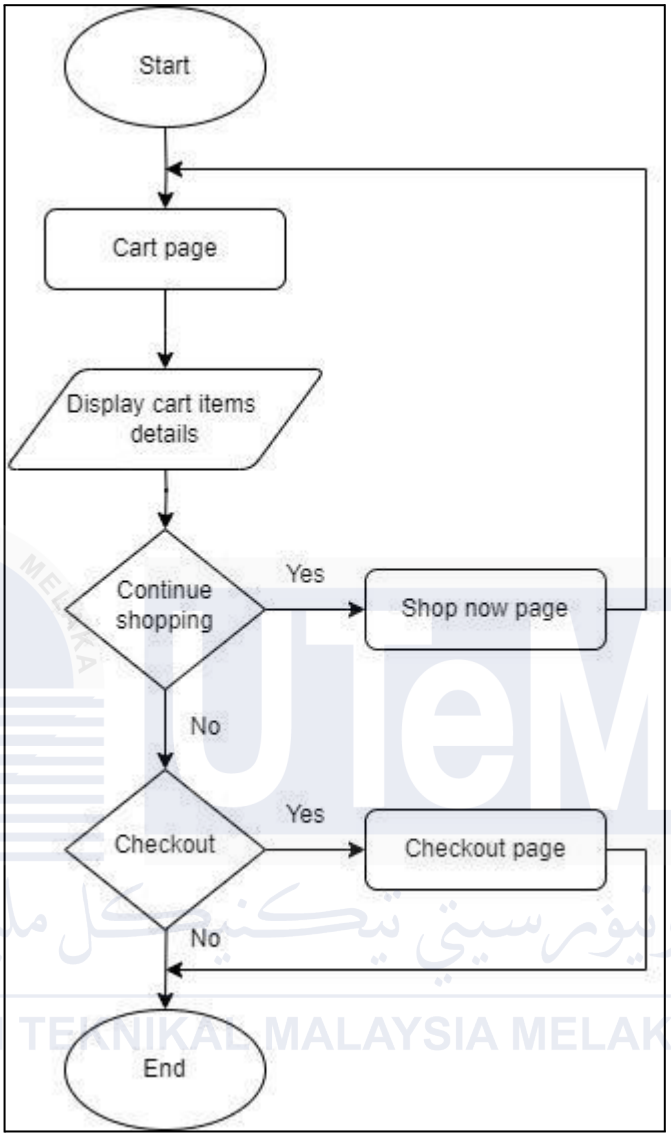
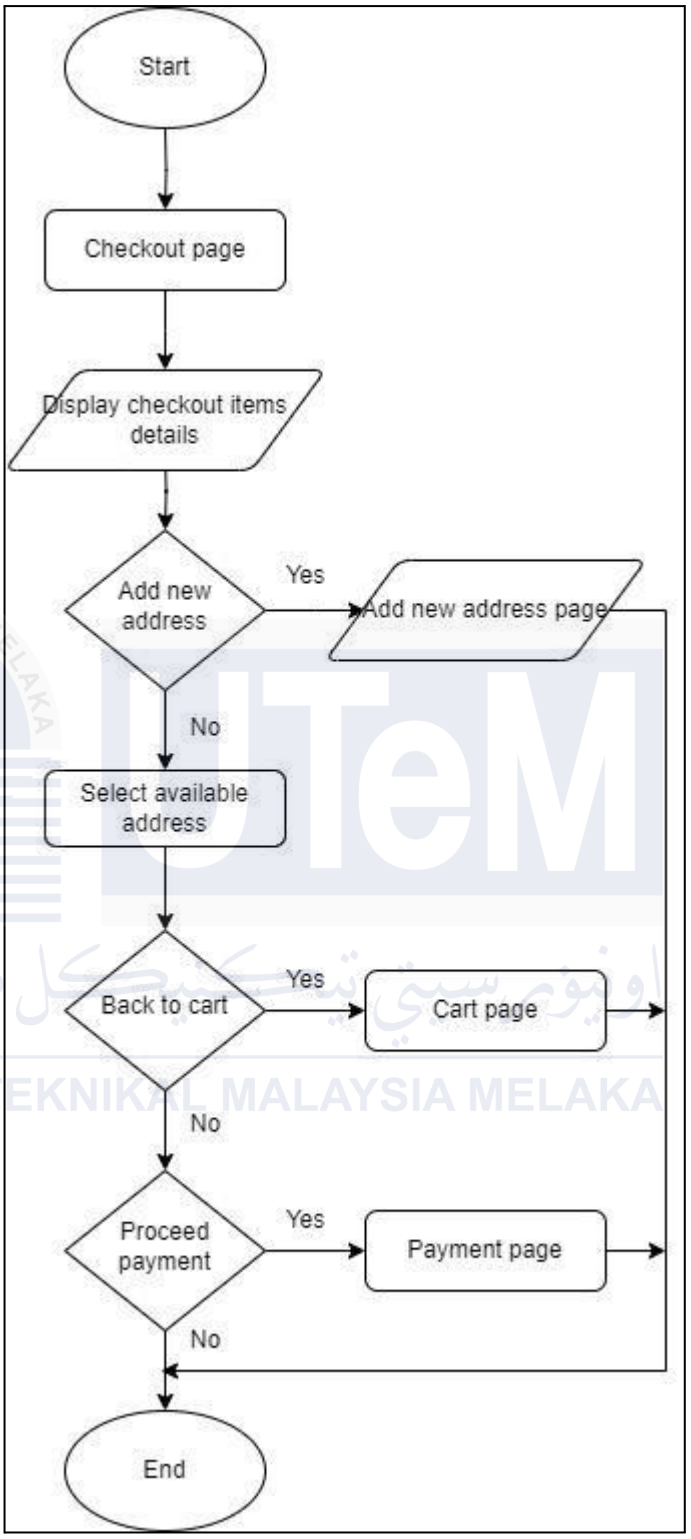


Figure 4.17 Filter shoes





61

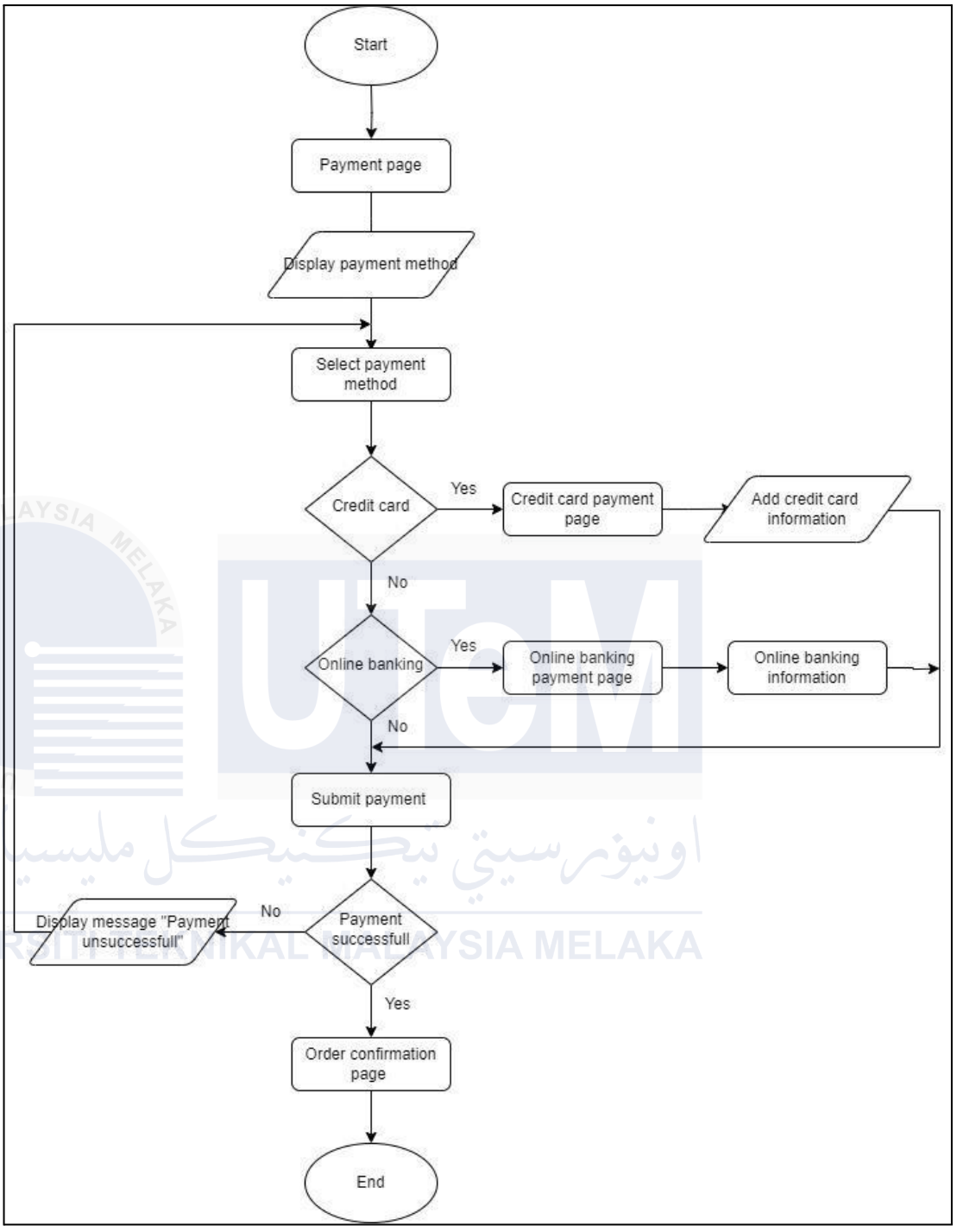


Figure 4.20 Payment Process

4.3 Conceptual Database Design

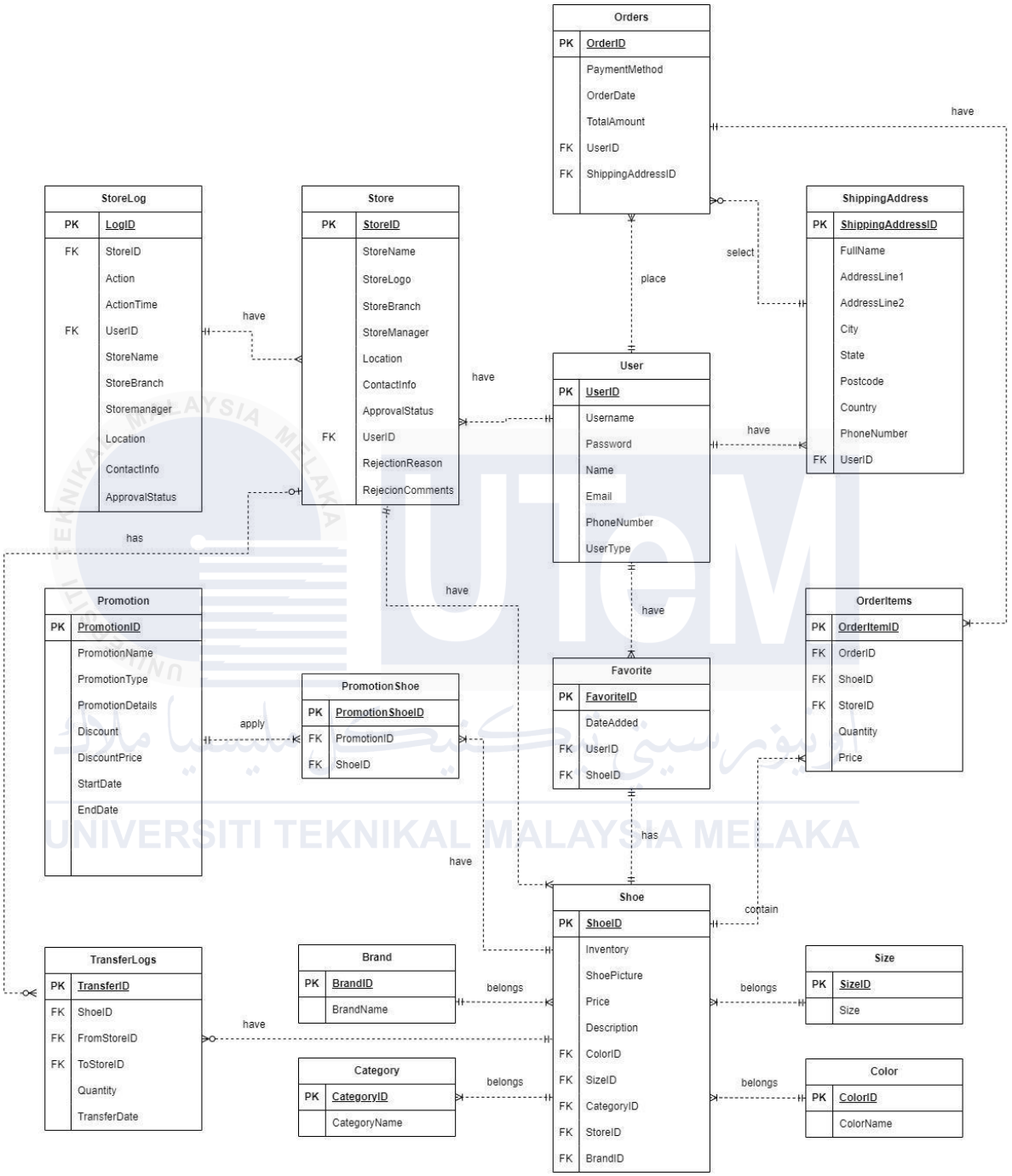


Figure 4.21 ERD

4.4 Logical Database Design

4.4.1 Data Dictionary

Table 4.1 User Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	USERID	USER IDENTITY DOCUMENT	XXXXXXXXXXXXXXX	INT(11)	YES	NO	PK	
2.	USERNAME	USERNAME	XXXXXXXXXXXX	VARCHAR (250)	NO	NO		
3.	PASSWORD	PASSWORD	**********	VARCHAR (255)	NO	NO		
4.	NAME	NAME	XXXXXXXXXXXXXXX	VARCHAR (100)	NO	NO		
5.	EMAIL	EMAIL	XXXXXXX@XXXXX.COM	VARCHAR (250)	NO	NO		

6.	PHONENUMBER	PHONE NUMBER	XXXXXXXXXX	VARCHAR (20)	NO	NO		
7.	USERTYPE	USER TYPE	XXXXXXXXXX	ENUM	NO	NO		

Table 4.2 Store Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	STOREID	STORE IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	YES	NO	PK	
2.	STORENAME	STORE NAME	XXXXXXXX	VARCHAR (250)	NO	NO		
3.	STOREBRANCH	STORE BRANCH	XXXXXXXXXX	VARCHAR (200)	NO	NO		
4.	STOREMANAGER	STORE MANAGER	XXXXXXXXXX	VARCHAR (250)	NO	NO		
5.	LOCATION	LOCATION	XXXXXXXXX	VARCHAR (200)	NO	NO		

6.	CONTACTINFO	CONTACT INFO	XXXXXXXXXXX	VARCHAR (20)	NO	NO		
7.	APPROVALSTATUS	APPROVAL STATUS	XXXXXXXXXX	ENUM	NO	NO		
8.	USERID	USER IDENTITY DOCUMENT	XXXXXXXXXXX	INT(11)	YES	NO	FK	USER
9.	REJECTION REASON	REJECTION REASON	XXXXX	TEXT	NO	YES		

			XXXXXX					
10.	REJECTION COMMENTS	REJECTION COMMENTS	XXXXXXXXX	TEXT	NO	YES		

Table 4.3 Storelog Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	LOGID	STORE IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	YES	NO	PK	
2.	STOREID	STORE IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	YES	NO	PK	STORE

3.	ACTION	ACTION	XXX XX XXX XX	VARCHAR (50)	NO	NO		
4.	ACTIONTIME	ACTION TIME	XXX XX XXX XXX XXX	TIMESTAMP	NO	NO		
5.	STORENAME	STORE NAME	XXX XX XXX XX	VARCHAR (250)	NO	NO		
6.	STOREBRANCH	STORE BRANCH	XXX XX XXX XXX XXX	VARCHAR (200)	NO	NO		

66

7.	STOREMANAGER	STORE MANAGER	XXX XX XXX XXX XXX	VARCHAR (250)	NO	NO		
8.	LOCATION	LOCATION	XXX XXX XXX	VARCHAR (200)	NO	NO		
9.	CONTACTINFO	CONTACT INFO	XXX XX XXX XXX	VARCHAR (20)	NO	NO		
10.	APPROVALSTATUS	APPROVAL STATUS	XXX XX XXX XX	ENUM	NO	NO		

11.	USERID	USER IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	FK	USER

Table 4.4 Shoe Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	SHOEID	SHOE IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	PK	
2.	INVENTORY	INVENTORY SHOE	XXX XX XXX XX	INT(255)	NO	NO		
3.	SHOEPICTURE	BRAND SHOE	XXX XX XXX XXX XXX	VARCHAR (255)	NO	NO		

4.	PRICE	CATEGORY SHOE	XXX XX XXX XXX XXX	FLOAT	NO	NO		
5.	DESCRIPTION	COLOR SHOE	XXX XXX	TEXT	NO	YES		
6.	SIZEID	SIZE IDENTITY DOCUMENT	XXX XX XXX XXX	INT(11)	NO	NO	FK	SIZE

7.	STOREID	STORE IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	NO	NO	FK	STORE
8.	BRANDID	BRAND IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	NO	NO	FK	BRAND
9.	CATEGORYID	CATEGORY IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	NO	NO	FK	CATEGORY
10.	COLORID	COLOR IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	NO	NO	FK	COLOR

Table 4.5 Transferlogs Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	TRANSFERID	TRANSFER IDENTITY DOCUMENT	XXXXXXXXXXXX	INT(11)	YES	NO	PK	
2.	SHOEID	SHOE IDENTITY DOCUMENT	XXXXXXXXXXXX	INT(11)	YES	NO	FK	SHOE

3.	FROMBRANCHID	FROM BRANCH ID	XXXXX XXXXXX XXX	INT(11)	YES	NO	FK	STORE
4.	TOBRANCHID	TO BRANCH ID	XXXXX XXXXXX XXX	INT(11)	YES	NO	FK	STORE
5.	QUANTITY	QUANTITY	XXXXX XXXXXX XXX	INT(11)	YES	NO		
6.	TRANSFERDATE	TRANSFERDATE	XXXXX XXXXXX XXX	TIMESTAMP	NO	NO		

Table 4.6 Brand Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	BRANDID	BRAND IDENTITY DOCUMENT	XXX XX XXX XXX	INT(11)	YES	NO	PK	
2.	BRANDNAME	BRAND NAME	XXX XX XXX XX	VARCHAR (255)	NO	NO		

Table 4.7 Category Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	CATEGORYID	CATEGORY IDENTITY DOCUMENT	XXXXXXXXXXXXXXX	INT(11)	YES	NO	PK	
2.	CATEGORYNAME	CATEGORY NAME	XXXXXXXXXX	VARCHAR (255)	NO	NO		

Table 4.8 Color Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	COLORID	COLOR IDENTITY DOCUMENT	XXXXXXXXXXXXXXX	INT(11)	YES	NO	PK	
2.	COLORNAME	COLOR NAME	XXXXXXXXXX	VARCHAR (255)	NO	NO		

Table 4.9 Size Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE

1.	SIZEID	SIZE IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	PK	
2.	SIZENAME	SIZE NAME	XXX XX XXX XX	VARCHAR (255)	NO	NO		

Table 4.10 Promotion Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	PROMOTION ID	PROMOTION IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	PK	
2.	PROMOTION NAME	PROMOTION NAME	XXX XX XXX XX	VARCHAR (255)	NO	NO		
3.	PROMOTION TYPE	PROMOTION TYPE	XXX XX XXX XXX XX	ENUM	NO	NO		
4.	PROMOTION DETAILS	PROMOTION DETAILS	XXX XX XXX XXX XX	VARCHAR (255)	NO	NO		
5.	DISCOUNT	DISCOUNT	XXX XX XXX	DECIMAL (5,2)	NO	NO		

			XX					
6.	DISCOUNT PRICE	DISCOUNT PRICE	XXXXXXXXXX	FLOAT	NO	NO		
7.	STARTDATE	START DATE	YYYY/MM/DD	DATE	NO	NO		
8.	ENDDATE	END DATE	YYYY/MM/DD	DATE	NO	NO		
9	STOREID	STORE IDENTITY DOCUMENT	XXXXXXXXXX	INT(11)	YES	NO	FK	STORE
10.	SHOEID	SHOE IDENTITY DOCUMENT	XXXXXXXX	INT(11)	YES	NO	FK	SHOE

Table 4.11 Promotion Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	PROMOTION SHOEID	PROMOTION SHOE IDENTITY DOCUMENT	XXXXX XXXXX X XXX	INT(11)	YES	NO	PK	
2.	PROMOTIONID	PROMOTION IDENTITY DOCUMENT	XXXXX XXXXX	INT(11)	NO	NO	FK	PROMOTION
3.	STOREID	STORE IDENTITY DOCUMENT	XXXXX XXXXX	INT(11)	NO	NO	FK	STORE

4.	SHOEID	SHOE IDENTITY DOCUMENT	XXXXX XXXXX X XXX	INT(11)	NO	NO	FK	SHOE

72

Table 4.12 Shipping Address Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	SHIPPINGADDRESSID	SHIPPING ADDRESS IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT (11)	YES	NO	PK	
2.	USERID	USER IDENTITY DOCUMENT	XXX XX XXX XX	INT (11)	NO	NO	NO	
3.	FULLNAME	FULL NAME	XXX XX XXX XX	VARCHAR (100)	NO	NO	NO	
4.	ADDRESSLINE1	ADDRESS LINE1	XXX XX XXX XXX XXX	VARCHAR (255)	NO	NO	NO	
5.	ADDRESSLINE2	ADDRESS LINE2	XXX XXX XXX	VARCHAR (255)	NO	NO	NO	

6.	CITY	CITY	XXX XX XXX XXX	VARCHAR (100)	NO	NO	NO	
7.	STATE	STATE	XXX XX XXX XX	VARCHAR (100)	NO	NO	NO	
8.	POSTCODE	POSTCODE	XXX XX XXX	VARCHAR (20)		NO	NO	

			XX					
9.	COUNTRY	COUNTRY	XXX XX XXX XX	VARCHAR (100)		NO	NO	
10.	PHONENUMBER	PHONE NUMBER	XXX XX XXX XX	VARCHAR (20)		NO	NO	

Table 4.13 Orders Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	ORDERID	ORDER IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	PK	
2.	USERID	USER IDENTITY DOCUMENT	XXX XX XXX XX	INT(11)	NO	NO		

3.	SHIPPINGADDRESSID	SHIPPING ADDRESS IDENTITY DOCUMENT	XXX XX XXX XXX XX	INT(11)	NO	NO		SHIPPINGADDRESS
4.	PAYMENTMETHOD	PAYMENT METHOD	XXX XX XXX XXX XXX	ENUM	NO	NO		
5.	ORDERDATE	ORDER DATE	XXX XX XXX	TIMESTAMP	NO	NO		

74

			XXX XXX					
6.	TOTAL AMOUNT	TOTAL AMOUNT	XXX XX XXX XXX XXX	DECIMAL (10,2)	NO	NO		

Table 4.14 Orderitems Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE
1.	ORDERITEMID	ORDER ITEM IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	PK	

2.	ORDERID	ORDER IDENTITY DOCUMENT	XXX XX XXX XX	INT(11)	NO	YES		ORDER
3.	SHOEID	SHOE IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	NO	YES	FK	SHOE
4.	STOREID	STORE IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	NO	YES	FK	STORE
5.	QUANTITY	QUANTITY	XXX XX XXX XXX XXX	INT(11)	NO	YES	FK	
6.	PRICE	PRICE	XXX XX XXX	DECIMAL (10,2)	NO	YES	FK	

			XXX XXX					

Table 4.15 Favorite Table

NO	ATTRIBUTE	DESCRIPTION	FORMAT	DATA TYPE	AUTO INCREMENT	NULL	PK OR FK	FK TABLE

1.	FAVORITEID	FAVORITE IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	YES	NO	PK	
2.	DATEADDED	DATE ADDED	XXX XX XXX XX	DATETIME	NO	NO		
3.	USERID	SHOE ENTHUSIAST IDENTITY DOCUMENT	XXX XX XXX XXX XX	INT(11)	NO	NO	FK	USER
4.	SHOEID	SHOE IDENTITY DOCUMENT	XXX XX XXX XXX XXX	INT(11)	NO	NO	FK	SHOE

4.5 Physical Database Design

4.5.1 Data Definition Language

```
CREATE TABLE `user` (
  `UserID` int(11) NOT NULL,
  `Username` varchar(250) NOT NULL,
  `Password` varchar(255) NOT NULL,
  `Name` varchar(100) NOT NULL,
  `Email` varchar(250) NOT NULL,
  `PhoneNumber` varchar(20) NOT NULL,
  `UserType` varchar(200) NOT NULL)


INSERT INTO `user` (`UserID`, `Username`, `Password`, `Name`, `Email`,
`PhoneNumber`, `UserType`) VALUES
(1, 'Aida',
'$2y$10$77C1j35mSFyp3hA9fnJvsOZmRubuIdTSg1qtcxrZnd.sOC5litYHa',
'Nurul Aida', 'aida21@gmail.com', '01123456788', 'ShoeEnthusiast'),
(2, 'Shura',
'$2y$10$OvNp.Wu59sJpjo6sTRZr2elIYAGmCONOeo7Z2Ujhh45mtgu53uFMW',
'Noraieshura Khair', 'aieshura123@gmail.com', '0109448144', 'Admin'),
(3, 'Nuyin',
'$2y$10$.9HlLJXSAL.fZMHLj8q7deg3LuzFTlhEWGyELMj7WJIpQkvAUcsk2',
'Bahirah', 'bahirah@gmail.com', '0128913088', 'Store '),
(23, 'Cuyaaa',
'$2y$10$qUGI/pG1DQnDe79pJKgTl.qy6Pqj64qSM4eHOpev4v3OOXcS3b472',
'Aiecuyaaa', 'cuyaaa@gmail.com', '0109448100', 'Store'),
(24, 'Faqihah',
'$2y$10$oKJwJzYABcjFYMHaHCwDlevOoOySNvSfcKCSqbZZgWsGH09Dcx
A.6', 'Faqihah', 'nurulfaqihah11@gmail.com', '01132705454', 'ShoeEnthusiast'),
(25, 'Jatul',
'$2y$10$VOunO6yapu1AmKRJ6Z82buIw4H2dSkhVbMj0qYzUaFjD//3Ktlh32',
'Izzatul Syamira', 'jatul123@gmail.com', '0126654322', 'ShoeEnthusiast'),
(26, 'Hayati',
'$2y$10$QB8Kj0mukru8NqKIqqrdTOivbseOiJqyAxQbbNNl8w9EmId5a3KH.',
'Hayati', 'hayati123@gmail.com', '019992456', 'ShoeEnthusiast'),
(27, 'Khairul',
'$2y$10$V6/B0GksLeIhFlwHY/6wEe2o0dxJLfny7ZTnBCgYwuGyil6xxIrRS',
'Khairulanwar', 'khairul123@gmail.com', '019887766', 'ShoeEnthusiast'),
(28, 'Baby',
'$2y$10$yEC50n1vvF2KtIWFHukNHezoMwdsJcI4UrbY8zNOwzz5VAqwmjcdG'
, 'Baby Muiz', 'baby@gmail.com', '0123443662', 'Store'),
(29, 'Hani',
'$2y$10$76tWqmjIZcDH9t.8ZrNGC.yHT78cRj5oAbjFXhoWFFybsTsA5My8e',
'Hani Honey', 'hani123@gmail.com', '011234562', 'Store');
```

```
CREATE TABLE `store` (
`StoreID` int(11) NOT NULL,
`StoreName` varchar(250) NOT NULL,
`StoreBranch` varchar(200) NOT NULL,
`StoreManager` varchar(250) NOT NULL,
`Location` varchar(200) NOT NULL,
`ContactInfo` varchar(20) NOT NULL,
`ApprovalStatus` enum('Pending','Approved','Rejected') NOT NULL,
`UserID` int(11) DEFAULT NULL)


INSERT INTO `store` (`StoreID`, `StoreName`, `StoreBranch`, `StoreManager`,
`Location`, `ContactInfo`, `ApprovalStatus`, `UserID`) VALUES
(1, 'Shoe Palace', 'Jonker Street', 'Azman Bakar', 'Melaka', '0198765432',
'Approved', 3),
(2, 'Footwear Haven', 'Banda Hilir', 'Faridah Abdullah', 'Melaka', '0164567890',
'Approved', 3),
(3, 'Sneaker Hub', 'Ayer Keroh', 'Kamarul Zaman', 'Melaka', '0137890123', '',
NULL),
(4, 'Sole City', 'Alor Gajah', 'Lim Teck Guan', 'Melaka', '0190123456', '', NULL),
(5, 'Stride & Step', 'Bukit Baru', 'Tan Siew Ling', 'Melaka', '0163456789', '',
NULL),
(6, 'Footwear Boutique', 'Ujong Pasir', 'Rashidah Yaakob', 'Melaka', '0136789012',
'', NULL),
(7, 'Shoe Emporium', 'Durian Tunggal', 'Hishamuddin Nor', 'Melaka',
'0162345678', '', NULL),
(8, 'Sneaker House', 'Masjid Tanah', 'Norliza Ibrahim', 'Melaka', '0199012345', '',
NULL),
(9, 'Kickz Central', 'Tanjung Kling', 'Shahrul Anuar', 'Melaka', '0161234567', '',
NULL),
(10, 'Elite Footwear', 'Bachang', 'Dev Kumar', 'Melaka', '0134567890', '', 3),
(665, 'Shoe UnderSea', 'Alor Gajah', 'Hayati', 'Melaka', '0189261866', 'Approved',
23),
(6677, 'Sasa', 'Durian Tunggal', 'Aieshura', 'Melaka', '0123456796', 'Rejected', 23),
(2147483647, 'Shushu', 'Ayer Keroh', 'Adilatul', 'Melaka', '011234534', 'Approved',
23);
```

78

```
CREATE TABLE `size` (
`SizeID` int(11) NOT NULL,
`Size` varchar(255) NOT NULL
```

```
)


INSERT INTO `size` (`SizeID`, `Size`) VALUES
(1, '38'),
(2, '39'),
(3, '40'),
(4, '41'),
(5, '42');



CREATE TABLE `shoe` (
`ShoeID` int(11) NOT NULL,
`Inventory` int(255) NOT NULL,
`ShoePicture` varchar(255) NOT NULL,
`Price` float NOT NULL,
`Description` text DEFAULT NULL,
`StoreID` int(11) DEFAULT NULL,
`BrandID` int(11) NOT NULL,
`CategoryID` int(11) DEFAULT NULL,
`ColorID` int(11) NOT NULL,
`SizeID` int(11) NOT NULL)

INSERT INTO `shoe` (`ShoeID`, `Inventory`, `ShoePicture`, `Price`,
`Description`, `StoreID`, `BrandID`, `CategoryID`, `ColorID`, `SizeID`) VALUES
(29, 153, 'images/nikesportred.jpg', 149, 'Nike sneakers in black color', 665, 1, 1, 5,
1),
(32, 80, 'images/asicssneakerwhite.jpg', 99, 'Adidas casual shoe in white color\r\n',
665, 9, 2, 6, 1),
(33, 78, 'images/highgray.jpg', 50, NULL, 1, 10, 3, 1, 2),
(35, 74, 'images/pumacasualblack.jpg', 219.99, NULL, 1, 3, 4, 4, 2),
(43, 11, 'images/pumasneakerwhite.jpg', 299.99, NULL, 1, 3, 5, 5, 2),
(47, 76, 'images/pumaformalblack.jpg', 219.99, NULL, 2, 3, 1, 5, 2),
(49, 658, 'images/adidascasualblack.jpg', 159.99, NULL, 2, 2, 2, 6, 3),
(51, 34, 'images/reeboksneakerwhite.jpg', 189.99, NULL, 6, 4, 3, 6, 3),
(59, 87, 'images/asicssneakerwhite.jpg', 219.99, NULL, 10, 9, 4, 6, 3),
(61, 26, 'images/underarmourhigheelsblack.jpg', 301.1, NULL, 9, 5, 5, 1, 4),
(63, 100, 'images/pic7.jpg', 68, NULL, 9, 3, 1, 5, 2),
(65, 54, 'images/pumasportwhite.jpg', 100.5, NULL, 8, 3, 1, 6, 2),
(73, 87, 'images/adidascasualblackwhite.jpg', 259.9, NULL, 10, 2, 3, 1, 5),
(77, 639, 'images/adidascasualblackwhite.jpg', 158, NULL, 7, 2, 4, 4, 5),
(85, 66, 'images/vansformalblack.jpg', 88, NULL, 7, 8, 5, 1, 5),
(93, 12, 'images/nikesportred.jpg', 299.9, NULL, 2, 1, 1, 1, 1),
```

```
(97, 218, 'images/pumacasualblack.jpg', 180.5, NULL, 3, 3, 2, 4, 2),
(101, 199, 'images/reeboksportblue.jpg', 55.5, NULL, 4, 4, 1, 4, 3),
(117, 78, 'images/reeboksportblue.jpg', 278.8, NULL, 5, 4, 1, 5, 4),
(119, 437, 'images/highyellow.jpg', 95, 'High heels that color in yellow', 10, 9, 5, 5,
3),
(120, 77, 'images/highyellow.jpg', 99, 'Beautiful shoe', 665, 9, 5, 5, 2),
(123, 60, 'images/newsneakerpink.jpg', 250, 'New Balance sneakers shoe in pink
color', 665, 6, 4, 9, 5),
(124, 80, 'images/nikesportred_1719190353.jpg', 149, 'Nike in red color', 665, 1, 1,
1, 4);



CREATE TABLE `shippingaddress` (
`ShippingAddressID` int(11) NOT NULL,
`UserID` int(11) NOT NULL,
`FullName` varchar(100) NOT NULL,
`AddressLine1` varchar(255) NOT NULL,
`AddressLine2` varchar(255) DEFAULT NULL,
`City` varchar(100) NOT NULL,
`State` varchar(100) NOT NULL,
`Postcode` varchar(20) NOT NULL,
`Country` varchar(100) NOT NULL,
`PhoneNumber` varchar(20) DEFAULT NULL)


INSERT INTO `shippingaddress` (`ShippingAddressID`, `UserID`, `FullName`,
`AddressLine1`, `AddressLine2`, `City`, `State`, `Postcode`, `Country`,
`PhoneNumber`) VALUES
(19, 1, 'Nurul Aida', 'Tasik Gelugur', '', 'Kubang Semang', 'Pulau Pinang', '13300',
'Malaysia', '011988643'),
(20, 1, 'Nurul Aida', 'Jalan Mahkota Tinggi', '', 'Butterworth', 'Pulau Pinang',
'13300', 'Malaysia', '011988643'),
(39, 2, 'Nurul Aida', 'Tasik Gelugur', '', 'Kubang Semang', 'Pulau Pinang', '13300',
'Malaysia', '11988643'),
(40, 26, 'Nurul Aida', 'Tasik Gelugur', '', 'Kubang Semang', 'Pulau Pinang', '13300',
'Malaysia', '11988643'),
(41, 23, 'Nurul Aida', 'Tasik Gelugur', '', 'Kubang Semang', 'Pulau Pinang', '13300',
'Malaysia', '119886435'),
(42, 27, 'Nurul Aida', 'Tasik Gelugur', '', 'Kubang Semang', 'Pulau Pinang', '13300',
'Malaysia', '119886435'),
```

```
(43, 27, 'Khairul', 'No 46, Kampung Air Merah', '', 'Kulim', 'Kedah', '09020',
'Malaysia', '01136893685');
```

80

```
CREATE TABLE `promotionshoe` (
`PromotionShoeID` int(11) NOT NULL,
`PromotionID` int(11) DEFAULT NULL,
`StoreID` int(11) NOT NULL,
`ShoeID` int(11) DEFAULT NULL)


INSERT INTO `promotionshoe` (`PromotionShoeID`, `PromotionID`, `StoreID`,
`ShoeID`) VALUES
(9, 4, 665, 32),
(13, 3, 665, 29),
(22, 5, 665, 32);
```

```
CREATE TABLE `promotion` (
`PromotionID` int(11) NOT NULL,
`PromotionName` varchar(255) NOT NULL,
`PromotionType` enum('Discount','BuyOneGetOneFree','BundleDeal','Other')
NOT NULL,
`PromotionDetails` varchar(255) DEFAULT NULL,
`Discount` decimal(5,2) DEFAULT NULL,
`DiscountPrice` float NOT NULL,
`StartDate` date NOT NULL,
`EndDate` date NOT NULL,
`ShoeID` int(11) DEFAULT NULL,
`StoreID` int(11) DEFAULT NULL)


INSERT INTO `promotion` (`PromotionID`, `PromotionName`, `PromotionType`,
`PromotionDetails`, `Discount`, `DiscountPrice`, `StartDate`, `EndDate`,
`ShoeID`, `StoreID`) VALUES
(3, 'Summer Sale', 'Discount', 'huhu', 100.00, 0, '2024-05-31', '2024-06-08', 29,
665),
(4, 'Holiday', 'BundleDeal', 'Holiday sales', 40.00, 57, '2024-06-08', '2024-06-28',
119, 665),
(5, 'Summer Sale', 'Discount', 'Summer Coming Soon', 60.00, 39.6, '2024-06-23',
'2024-07-02', 32, 665),
(7, 'Sy', 'Discount', 'Shu', 10.00, 225, '2024-06-24', '2024-07-05', 123, 665);
```

```
CREATE TABLE `orders` (
`OrderID` int(11) NOT NULL,
`UserID` int(11) NOT NULL,
`ShippingAddressID` int(11) NOT NULL,
`PaymentMethod` enum('Credit Card','Bank Transfer','Online Banking') NOT
```

81

```
NULL,
`OrderDate` timestamp NOT NULL DEFAULT current_timestamp(),
`TotalAmount` decimal(10,2) NOT NULL,
`ShoeID` int(11) DEFAULT NULL)


INSERT INTO `orders` (`OrderID`, `UserID`, `ShippingAddressID`,
`PaymentMethod`, `OrderDate`, `TotalAmount`, `ShoeID`) VALUES
(1, 1, 19, 'Credit Card', '2023-04-06 11:52:40', 899.40, 29),
(2, 1, 19, 'Credit Card', '2024-06-06 11:57:28', 899.40, 32),
(3, 1, 19, 'Credit Card', '2024-06-06 11:58:09', 899.40, NULL),
(4, 1, 19, 'Credit Card', '2023-04-06 11:59:29', 659.97, NULL),
(5, 1, 19, 'Credit Card', '2024-05-06 22:03:04', 449.70, NULL),
(6, 1, 19, 'Credit Card', '2023-04-08 01:53:14', 449.70, NULL),
(7, 1, 19, 'Credit Card', '2024-04-08 02:44:00', 299.80, NULL),
(8, 1, 19, 'Credit Card', '2024-06-08 06:34:15', 1179.76, NULL),
(9, 1, 19, 'Credit Card', '2024-03-08 08:18:07', 599.60, NULL),
(188, 2, 39, 'Credit Card', '2024-06-23 04:42:14', 198.00, NULL),
(239, 26, 40, 'Credit Card', '2024-06-21 08:57:13', 190.00, NULL),
(251, 1, 19, 'Credit Card', '2024-06-22 20:13:59', 219.99, NULL),
(264, 1, 19, 'Credit Card', '2024-06-21 02:33:05', 99.00, NULL),
(294, 1, 19, 'Credit Card', '2024-06-22 20:16:49', 289.00, NULL),
(320, 1, 19, 'Credit Card', '2024-06-21 08:57:13', 100.00, NULL),
(387, 26, 40, 'Credit Card', '2024-06-21 08:57:13', 95.00, NULL),
(395, 26, 40, 'Credit Card', '2024-06-21 08:57:13', 190.00, NULL),
(507, 23, 41, 'Credit Card', '2024-06-23 05:20:25', 149.00, 29),
(508, 1, 19, 'Credit Card', '2024-06-21 08:57:13', 50.00, NULL),
(523, 1, 19, 'Credit Card', '2024-04-21 02:46:27', 0.00, NULL),
(599, 1, 19, 'Credit Card', '2024-06-21 02:42:13', 149.00, NULL),
(627, 1, 19, 'Credit Card', '2024-04-21 08:57:13', 95.00, NULL),
(681, 1, 19, 'Credit Card', '2024-06-21 02:10:06', 50.00, NULL),
(703, 23, 41, 'Credit Card', '2024-06-23 05:15:45', 99.00, NULL),
(781, 1, 19, 'Credit Card', '2024-06-23 02:42:43', 95.00, NULL),
(822, 1, 19, 'Credit Card', '2024-05-23 02:24:08', 95.00, NULL),
(872, 1, 19, 'Credit Card', '2024-06-21 08:57:13', 50.00, NULL),
(939, 1, 19, 'Credit Card', '2024-06-21 02:26:12', 99.00, NULL),
(946, 26, 40, 'Credit Card', '2024-06-21 08:57:13', 361.00, NULL),
(960, 1, 19, 'Credit Card', '2024-06-21 02:15:11', 294.00, NULL),
```

```
(990, 1, 19, 'Credit Card', '2024-06-21 08:57:13', 696.50, NULL),
(1286, 1, 19, 'Credit Card', '2024-06-21 08:57:13', 50.00, NULL),
(3505, 27, 42, 'Credit Card', '2024-06-23 14:32:19', 206.00, NULL),
(4376, 23, 41, 'Credit Card', '2024-06-23 05:29:45', 188.60, NULL),
(4477, 27, 42, 'Credit Card', '2024-06-23 14:39:40', 3290.79, NULL),
(6583, 23, 41, 'Credit Card', '2024-06-24 00:57:32', 6098.40, NULL),
(9061, 27, 43, 'Credit Card', '2024-06-23 14:35:14', 439.98, NULL),
(135828, 1, 20, 'Credit Card', '2024-06-20 23:54:19', 219.99, NULL),
(484608, 1, 20, 'Credit Card', '2024-06-21 01:00:27', 219.99, NULL),
(490684, 1, 19, 'Credit Card', '2024-06-21 00:17:35', 95.00, NULL),
(553546, 1, 19, 'Credit Card', '2024-06-21 00:03:33', 50.00, NULL),
```



```
(563614, 1, 20, 'Credit Card', '2024-06-21 00:38:34', 99.00, NULL),
(582557, 1, 19, 'Credit Card', '2024-06-19 22:50:49', 318.99, NULL),
(614819, 1, 20, 'Credit Card', '2024-06-21 00:16:01', 50.00, NULL),
(615242, 1, 20, 'Credit Card', '2024-06-20 23:59:49', 99.00, NULL),
(615519, 1, 20, 'Credit Card', '2024-06-21 00:09:23', 99.00, NULL),
(631940, 1, 19, 'Credit Card', '2024-06-19 23:25:11', 160.75, NULL),
(672434, 1, 20, 'Credit Card', '2024-06-21 00:01:32', 0.00, NULL),
(715915, 1, 20, 'Credit Card', '2024-06-21 01:32:54', 250.00, NULL),
(783966, 1, 20, 'Credit Card', '2024-06-21 00:48:42', 99.00, NULL),
(802080, 1, 19, 'Credit Card', '2024-06-19 23:50:05', 314.99, NULL),
(839020, 1, 20, 'Credit Card', '2024-06-20 23:46:10', 118.00, NULL),
(862970, 1, 19, 'Credit Card', '2024-06-21 00:24:40', 99.00, NULL),
(890307, 1, 20, 'Credit Card', '2024-06-21 00:07:06', 50.00, NULL),
(945389, 1, 19, 'Credit Card', '2024-06-19 07:53:47', 448.00, NULL),
(945390, 1, 20, 'Credit Card', '2024-06-21 01:00:27', 219.99, NULL),
(945391, 1, 20, 'Credit Card', '2024-06-21 01:29:27', 250.00, NULL),
(945392, 1, 20, 'Credit Card', '2024-06-21 01:38:32', 301.10, NULL),
(945393, 1, 20, 'Credit Card', '2024-06-21 01:41:57', 439.98, NULL),
(945394, 1, 20, 'Credit Card', '2024-06-21 01:41:57', 439.98, NULL),
(945395, 1, 19, 'Credit Card', '2024-06-21 01:48:57', 198.00, NULL),
(945396, 1, 19, 'Credit Card', '2024-06-21 01:48:57', 198.00, NULL),
(945397, 1, 19, 'Credit Card', '2024-06-21 01:48:57', 198.00, NULL),
(945398, 1, 19, 'Credit Card', '2024-06-21 01:51:11', 99.00, NULL),
(945399, 1, 19, 'Credit Card', '2024-06-21 01:52:00', 158.00, NULL),
(945400, 1, 19, 'Credit Card', '2024-06-21 01:57:55', 210.00, NULL),
(945401, 1, 19, 'Credit Card', '2024-06-21 02:01:39', 100.00, NULL),
(2147483647, 1, 19, 'Credit Card', '2024-06-21 02:07:50', 379.98, NULL);
```

```
CREATE TABLE `orderitems` (
  `OrderItemID` int(11) NOT NULL,
  `OrderID` int(11) DEFAULT NULL,
  `ShoeID` int(11) DEFAULT NULL,
  `Quantity` int(11) DEFAULT NULL,
  `Price` decimal(10,2) DEFAULT NULL)


INSERT INTO `orderitems` (`OrderItemID`, `OrderID`, `ShoeID`, `Quantity`,
`Price`) VALUES
(1, 4376, 29, 1, 149.00),
(2, 4376, 32, 1, 39.60),
(3, 3505, 29, 1, 149.00),
(4, 3505, 119, 1, 57.00),
(5, 9061, 47, 2, 219.99),
(6, 4477, 49, 1, 159.99),
(7, 4477, 97, 4, 180.50),
```



```
(8, 4477, 61, 8, 301.10),
(9, 6583, 32, 154, 39.60);



CREATE TABLE `favorite` (
`FavoriteID` int(11) NOT NULL,
`DateAdded` datetime NOT NULL DEFAULT current_timestamp(),
`UserID` int(11) DEFAULT NULL,
`ShoeID` int(11) DEFAULT NULL)


INSERT INTO `favorite` (`FavoriteID`, `DateAdded`, `UserID`, `ShoeID`)
VALUES
(33, '2024-06-15 10:24:49', 1, 32),
(34, '2024-06-15 10:24:57', 1, 119),
(35, '2024-06-15 10:28:13', 1, 29),
(36, '2024-06-17 23:31:57', 1, 63),
(37, '2024-06-17 23:32:16', 1, 33),
(38, '2024-06-18 10:20:21', 26, 33),
(39, '2024-06-20 17:15:42', 23, 33),
(43, '2024-06-20 18:52:21', 23, 65),
(45, '2024-06-20 18:56:37', 23, 119),
(46, '2024-06-20 18:56:44', 23, 49),
```

```
(47, '2024-06-21 20:05:05', 2, 119),
(48, '2024-06-23 13:20:12', 23, 29);



CREATE TABLE `color` (
`ColorID` int(11) NOT NULL,
`ColorName` varchar(255) NOT NULL)


INSERT INTO `color` (`ColorID`, `ColorName`) VALUES
(1, 'Red'),
(2, 'Blue'),
(3, 'Green'),
(4, 'Yellow'),
(5, 'Black'),
(6, 'White'),
(7, 'Purple'),
(8, 'Orange'),
(9, 'Pink'),
(10, 'Gray');
```



```
CREATE TABLE `category` (
`CategoryID` int(11) NOT NULL,
`CategoryName` varchar(255) NOT NULL)


INSERT INTO `category` (`CategoryID`, `CategoryName`) VALUES
(1, 'Sports'),
(2, 'Casual'),
(3, 'Formal'),
(4, 'Sneakers'),
(5, 'High Heels');
```

```
ALTER TABLE `brand`
ADD PRIMARY KEY (`BrandID`);

ALTER TABLE `category`
ADD PRIMARY KEY (`CategoryID`);
```

```
ALTER TABLE `color`
ADD PRIMARY KEY (`ColorID`);

ALTER TABLE `favorite`
ADD PRIMARY KEY (`FavoriteID`),
ADD KEY `ShoeID` (`ShoeID`),
ADD KEY `FK_User_Favorite` (`UserID`);

ALTER TABLE `orderitems`
ADD PRIMARY KEY (`OrderItemID`),
ADD KEY `OrderID` (`OrderID`),
ADD KEY `ShoeID` (`ShoeID`);

ALTER TABLE `orders`
ADD PRIMARY KEY (`OrderID`),
ADD KEY `UserID` (`UserID`),
ADD KEY `ShippingAddressID` (`ShippingAddressID`),
ADD KEY `ShoeID` (`ShoeID`);

ALTER TABLE `promotion`
ADD PRIMARY KEY (`PromotionID`),
ADD KEY `FK_Promotion_Shoe` (`ShoeID`),
ADD KEY `FK_Promotion_Store` (`StoreID`);

ALTER TABLE `promotionshoe`
ADD PRIMARY KEY (`PromotionShoeID`),
ADD KEY `FK_PromotionShoe_Promotion` (`PromotionID`),
```



```
ADD KEY `FK_PromotionShoe_Shoe` (`ShoeID`);

ALTER TABLE `shippingaddress`
ADD PRIMARY KEY (`ShippingAddressID`),
ADD KEY `UserID` (`UserID`);

ALTER TABLE `shoe`
ADD PRIMARY KEY (`ShoeID`),
ADD KEY `FK_StoreID` (`StoreID`),
ADD KEY `FK_BrandID` (`BrandID`),
ADD KEY `FK_CategoryID` (`CategoryID`),
ADD KEY `FK_ColorID` (`ColorID`),
ADD KEY `FK_SizeID` (`SizeID`);

ALTER TABLE `size`
ADD PRIMARY KEY (`SizeID`);
```

```
ALTER TABLE `store`
ADD PRIMARY KEY (`StoreID`),
ADD KEY `FK_User_Store` (`UserID`);

ALTER TABLE `user`
ADD PRIMARY KEY (`UserID`),
ADD UNIQUE KEY `Username` (`Username`);

ALTER TABLE `brand`
MODIFY `BrandID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=11;


ALTER TABLE `category`
 MODIFY `CategoryID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;

ALTER TABLE `color`
MODIFY `ColorID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=11;

ALTER TABLE `favorite`
 MODIFY `FavoriteID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=49;

ALTER TABLE `orderitems`
 MODIFY `OrderItemID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=10;

ALTER TABLE `orders`
 MODIFY `OrderID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=2147483648;
```

86

```
ALTER TABLE `promotion`
 MODIFY `PromotionID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=8;

ALTER TABLE `promotionshoe`
 MODIFY `PromotionShoeID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=23;

ALTER TABLE `shippingaddress`
 MODIFY `ShippingAddressID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=44;
```

```
ALTER TABLE `shoe`
  MODIFY `ShoeID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=125;

ALTER TABLE `size`
  MODIFY `SizeID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=6;

ALTER TABLE `store`
MODIFY `StoreID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=2147483648;

ALTER TABLE `user`
MODIFY `UserID` int(11) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=30;


ALTER TABLE `favorite`
ADD CONSTRAINT `FK_User_Favorite` FOREIGN KEY (`UserID`)
REFERENCES `user` (`UserID`),
  ADD CONSTRAINT `favorite_ibfk_2` FOREIGN KEY (`ShoeID`)
REFERENCES `shoe` (`ShoeID`);

ALTER TABLE `orderitems`
ADD CONSTRAINT `orderitems_ibfk_1` FOREIGN KEY (`OrderID`)
REFERENCES `orders` (`OrderID`),
  ADD CONSTRAINT `orderitems_ibfk_2` FOREIGN KEY (`ShoeID`)
REFERENCES `shoe` (`ShoeID`);

ALTER TABLE `orders`
ADD CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`UserID`)
REFERENCES `user` (`UserID`),
ADD CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`ShippingAddressID`)
REFERENCES `shippingaddress` (`ShippingAddressID`) ON DELETE NO
ACTION ON UPDATE NO ACTION,
ADD CONSTRAINT `orders_ibfk_3` FOREIGN KEY (`ShoeID`)
```



```
REFERENCES `shoe` (`ShoeID`);

ALTER TABLE `promotion`
ADD CONSTRAINT `FK_Promotion_Shoe` FOREIGN KEY (`ShoeID`)
REFERENCES `shoe` (`ShoeID`),
  ADD CONSTRAINT `FK_Promotion_Store` FOREIGN KEY (`StoreID`)
REFERENCES `store` (`StoreID`);
```

```
ALTER TABLE `promotionshoe`
ADD CONSTRAINT `FK_PromotionShoe_Promotion` FOREIGN KEY
(`PromotionID`) REFERENCES `promotion` (`PromotionID`),
ADD CONSTRAINT `FK_PromotionShoe_Shoe` FOREIGN KEY (`ShoeID`)
REFERENCES `shoe` (`ShoeID`);

ALTER TABLE `shippingaddress`
ADD CONSTRAINT `shippingaddress_ibfk_1` FOREIGN KEY (`UserID`)
REFERENCES `user` (`UserID`);

ALTER TABLE `shoe`
ADD CONSTRAINT `FK_BrandID` FOREIGN KEY (`BrandID`)
REFERENCES `brand` (`BrandID`),
ADD CONSTRAINT `FK_CategoryID` FOREIGN KEY (`CategoryID`)
REFERENCES `category` (`CategoryID`),
ADD CONSTRAINT `FK_ColorID` FOREIGN KEY (`ColorID`)
REFERENCES `color` (`ColorID`),
ADD CONSTRAINT `FK_SizeID` FOREIGN KEY (`SizeID`) REFERENCES
`size` (`SizeID`),
ADD CONSTRAINT `FK_StoreID` FOREIGN KEY (`StoreID`) REFERENCES
`store` (`StoreID`);

ALTER TABLE `store`
ADD CONSTRAINT `FK_User_Store` FOREIGN KEY (`UserID`)
REFERENCES `user` (`UserID`);
COMMIT;
```

88

4.6 GUI Design

Register Page

register.php

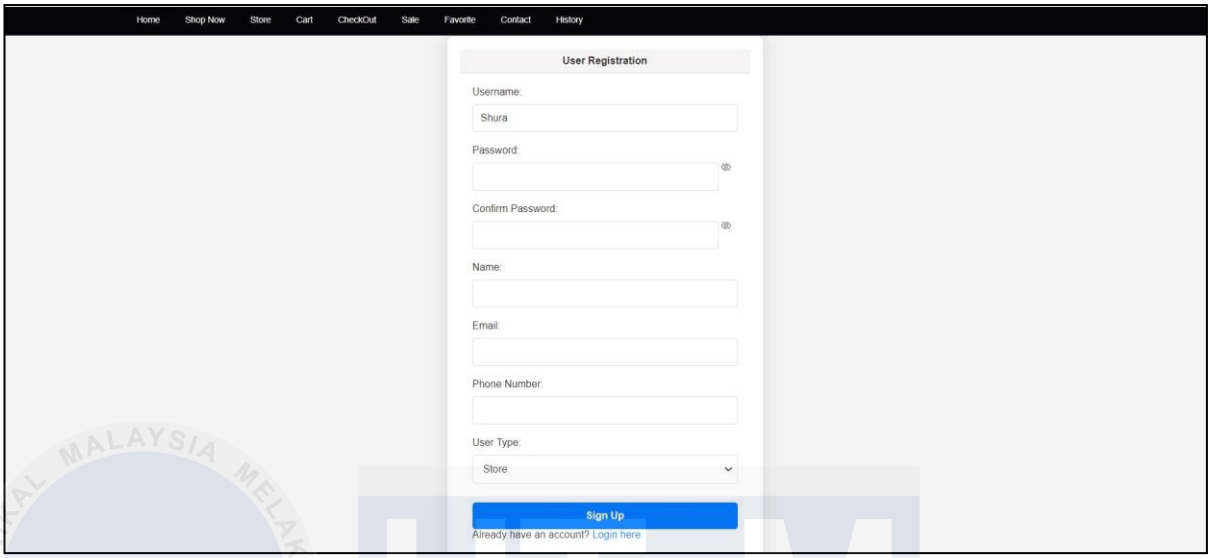


Figure 4.22 Register

Login Page

login.php

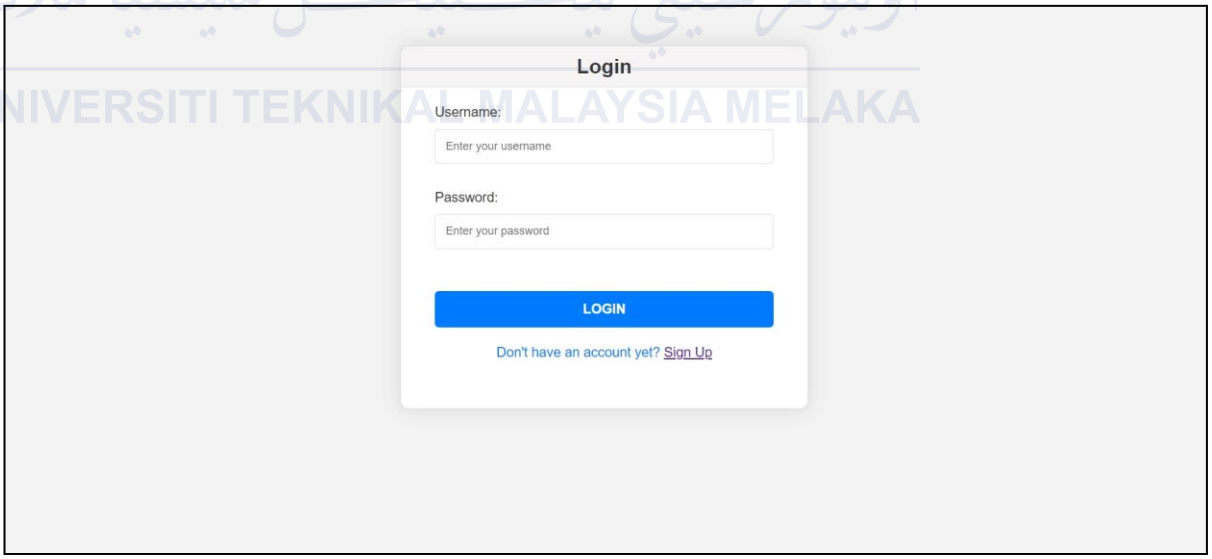


Figure 4.23 Login

Shoe Enthusiast Module

shoeenthusiast_acc.php

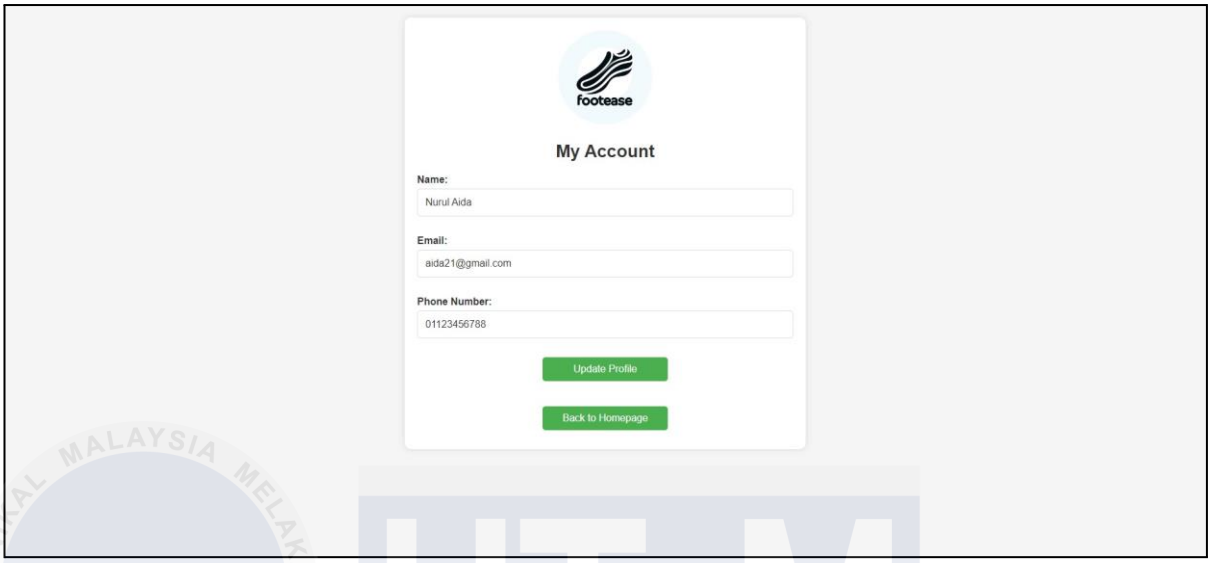


Figure 4.24 Shoe Enthusiast Account

index.php

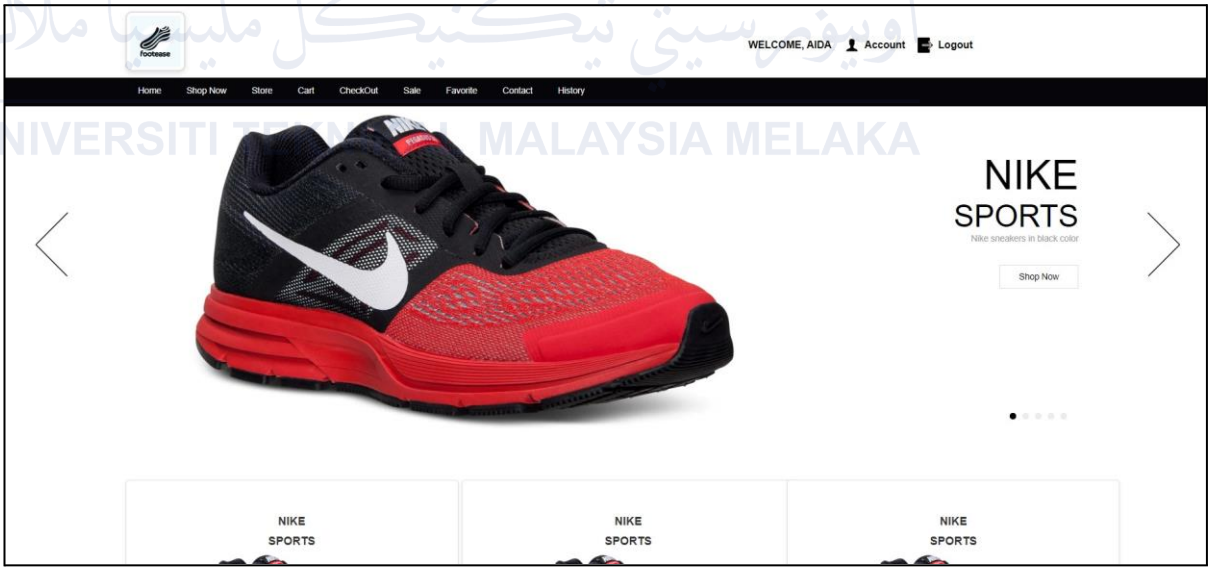


Figure 4.25 Shoe Enthusiast Homepage

shop.php

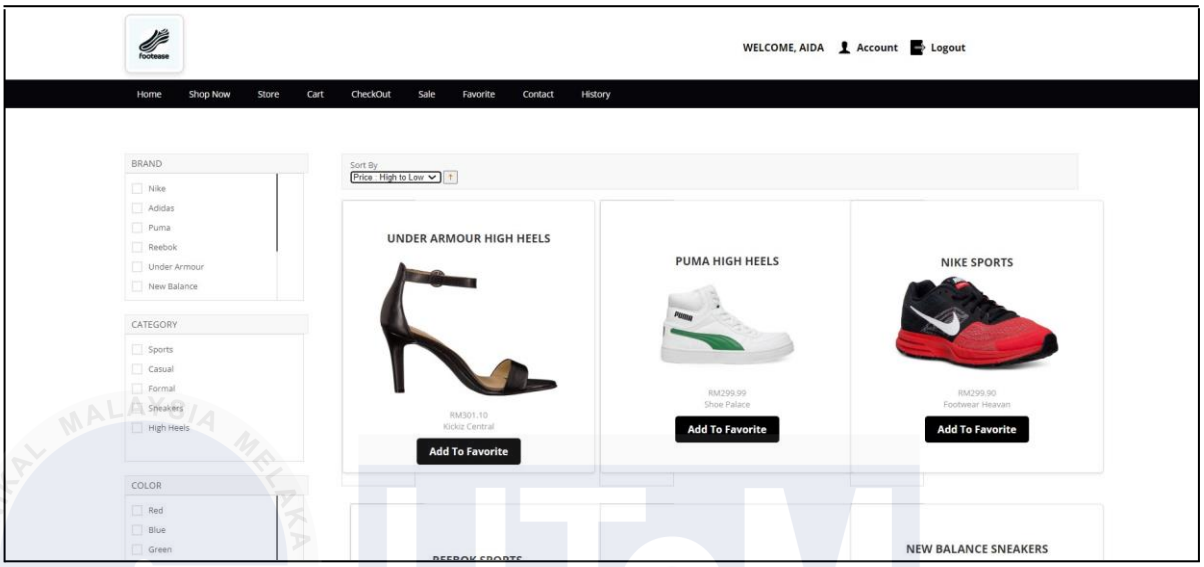


Figure 4.26 Shop Now

store.php

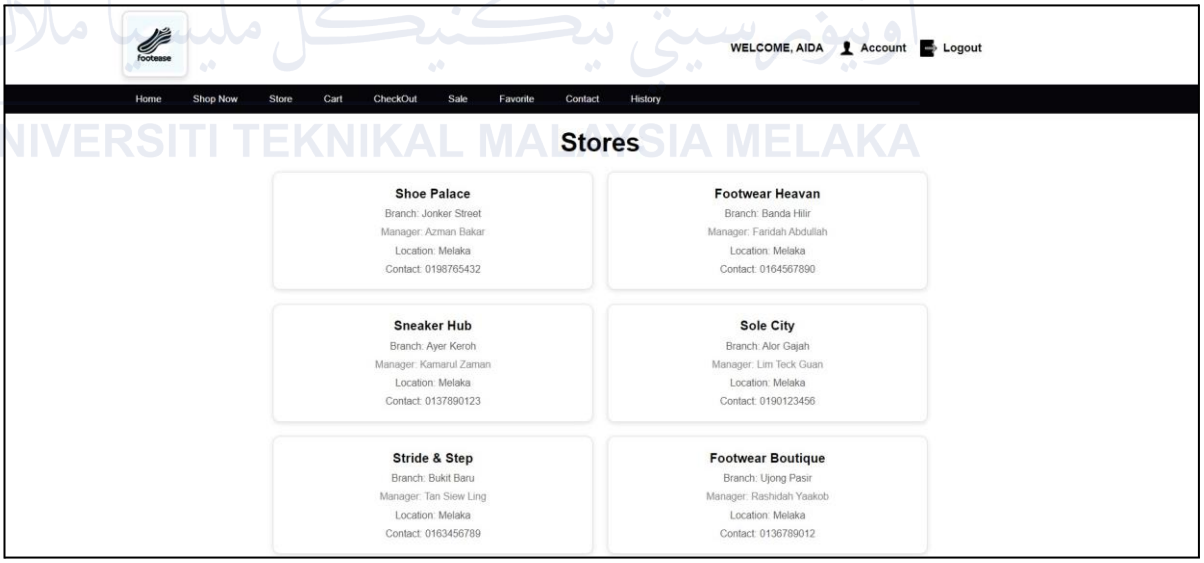


Figure 4.27 Store

store_details.php

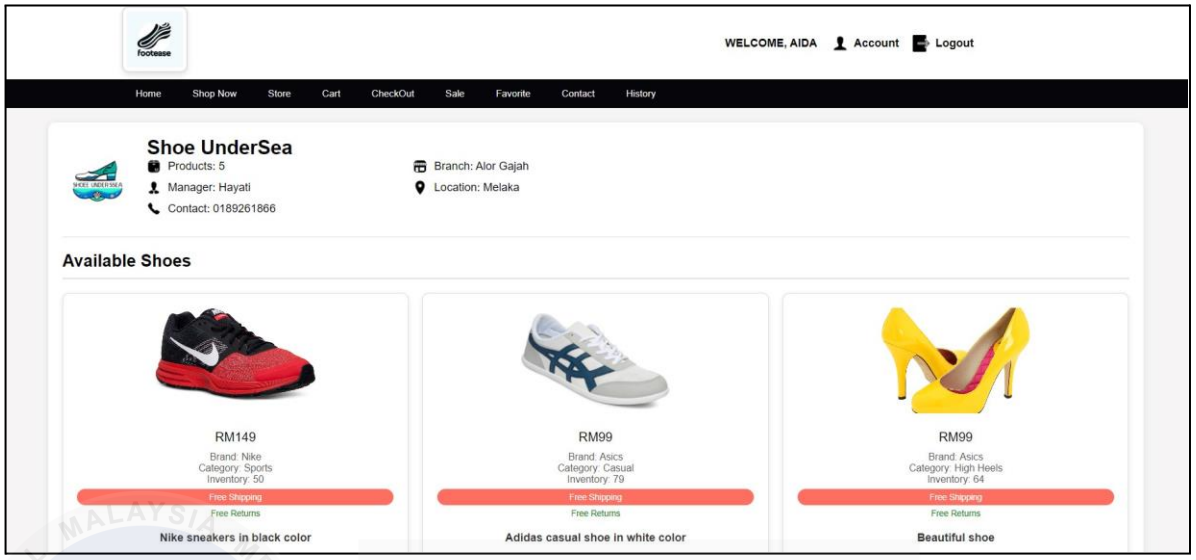


Figure 4.28 Store Details

single.php

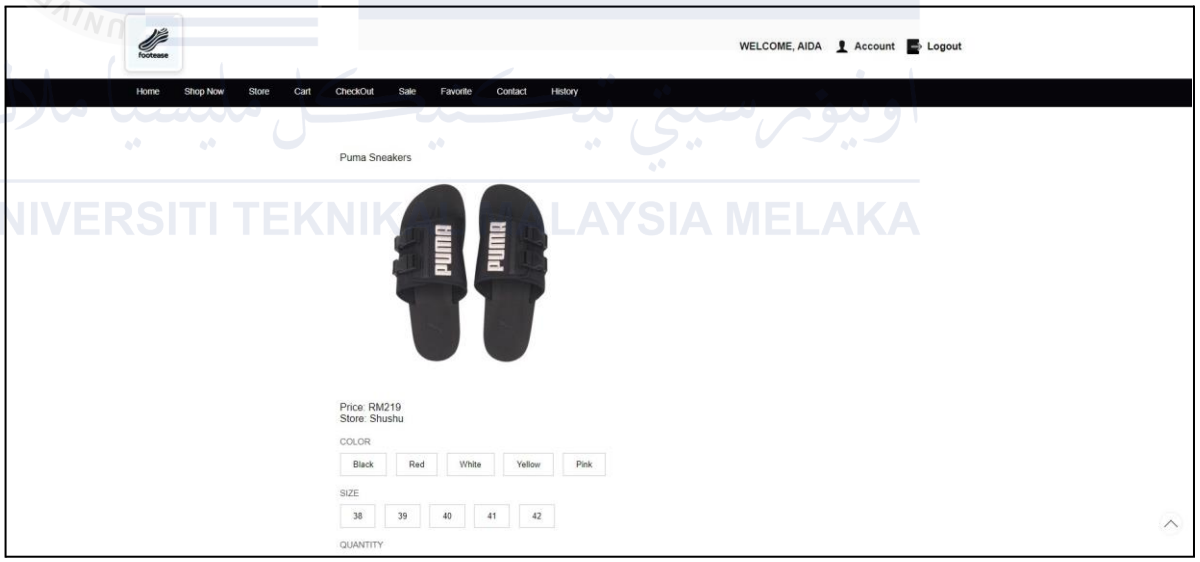


Figure 4.29 Shoe Details

continue single.php

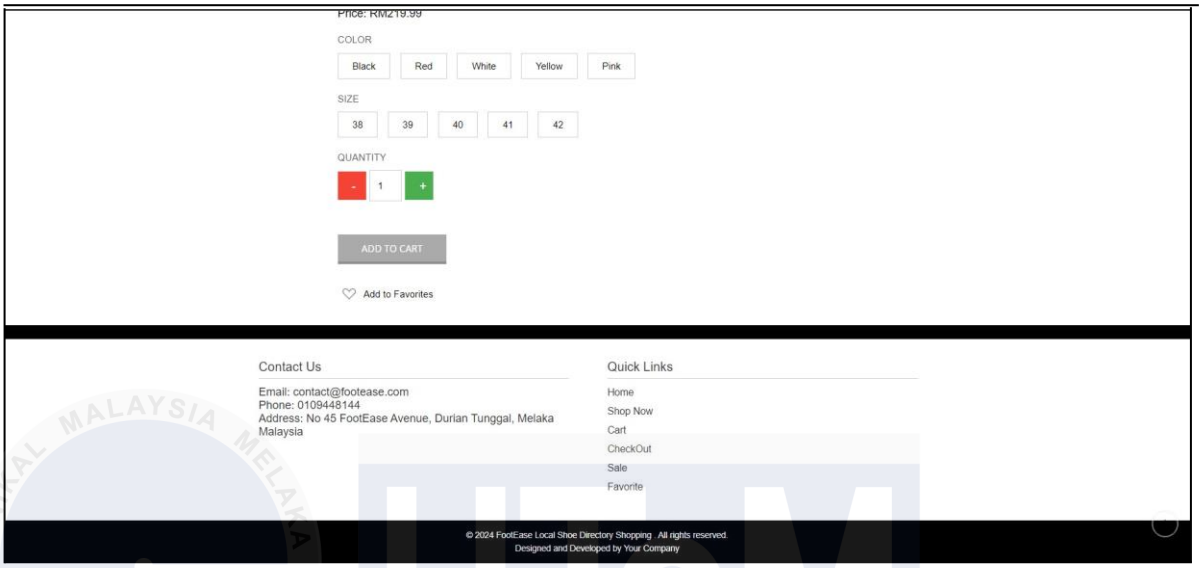


Figure 4.30 Continue Shoe Details

cart.php

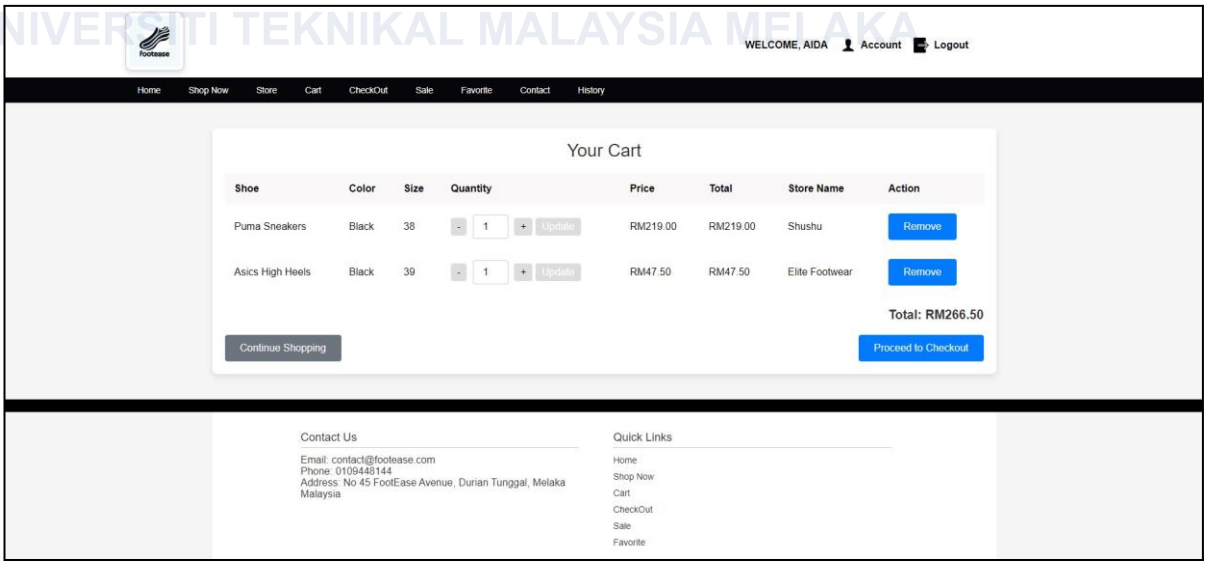


Figure 4.31 Cart

checkout.php

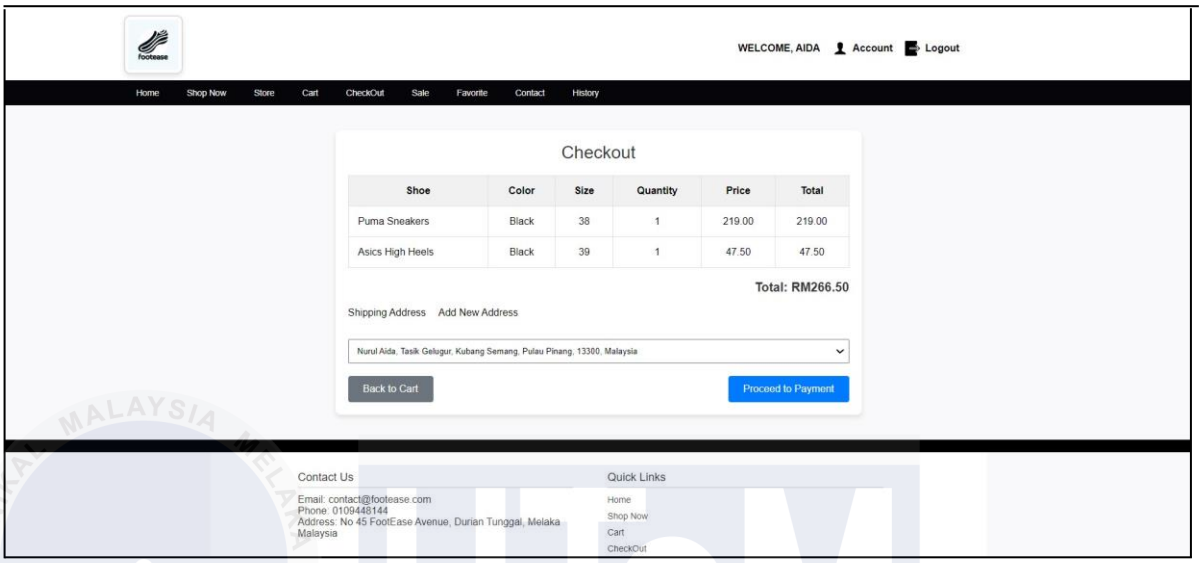


Figure 4.32 Checkout

address_selection.php

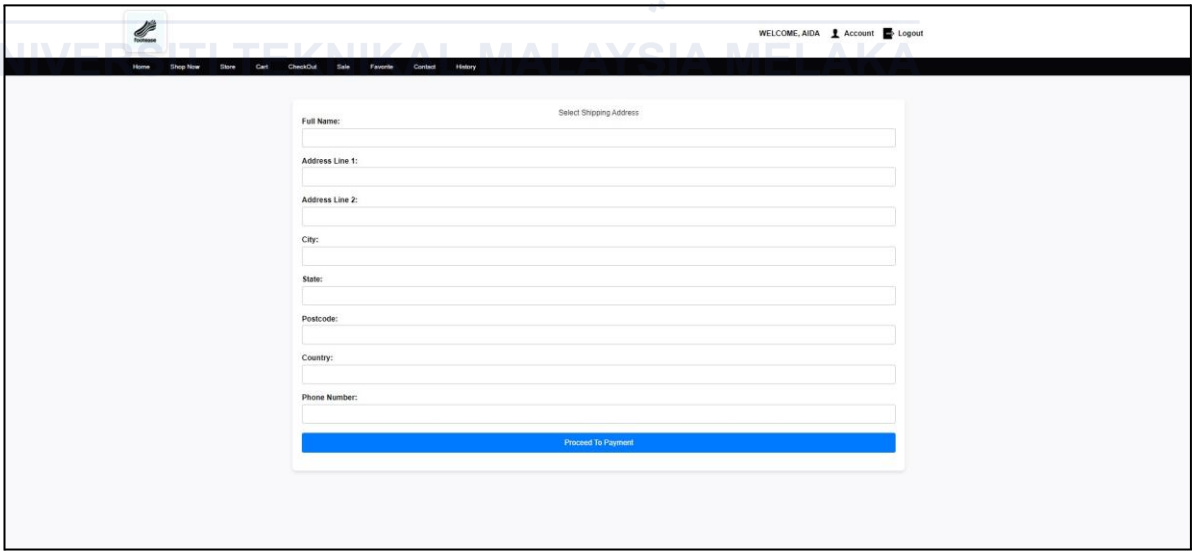


Figure 4.33 Add Address

payment_selection.php

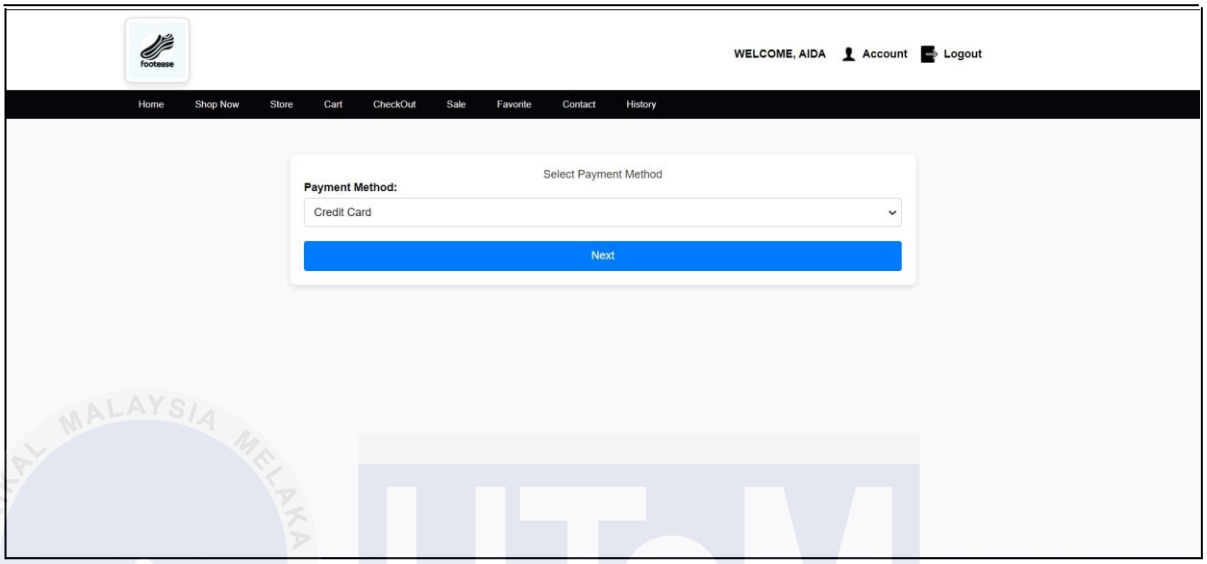


Figure 4.34 Payment Method Credit Card

credit_card_payment.php

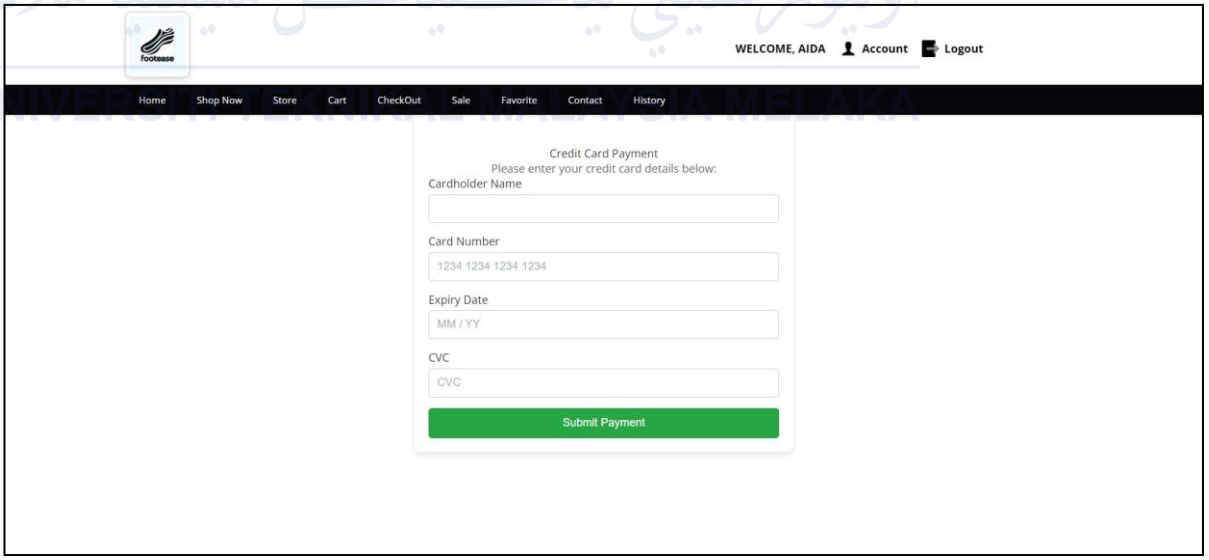


Figure 4.35 Credit Card Form

information credit_card_payment.php

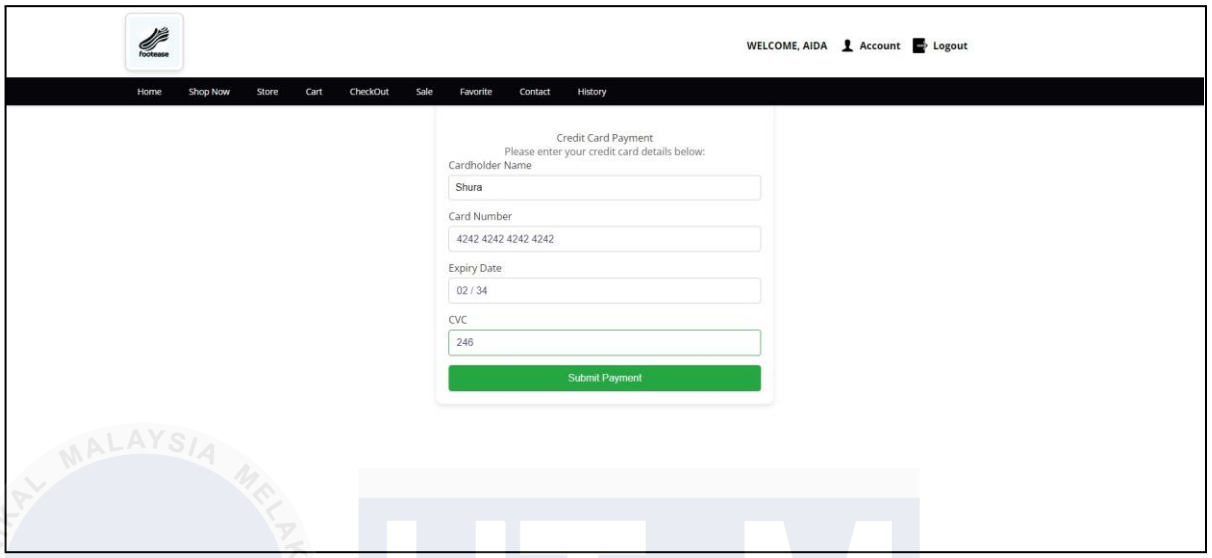


Figure 4.36 Credit Card Details

payment_selection.php

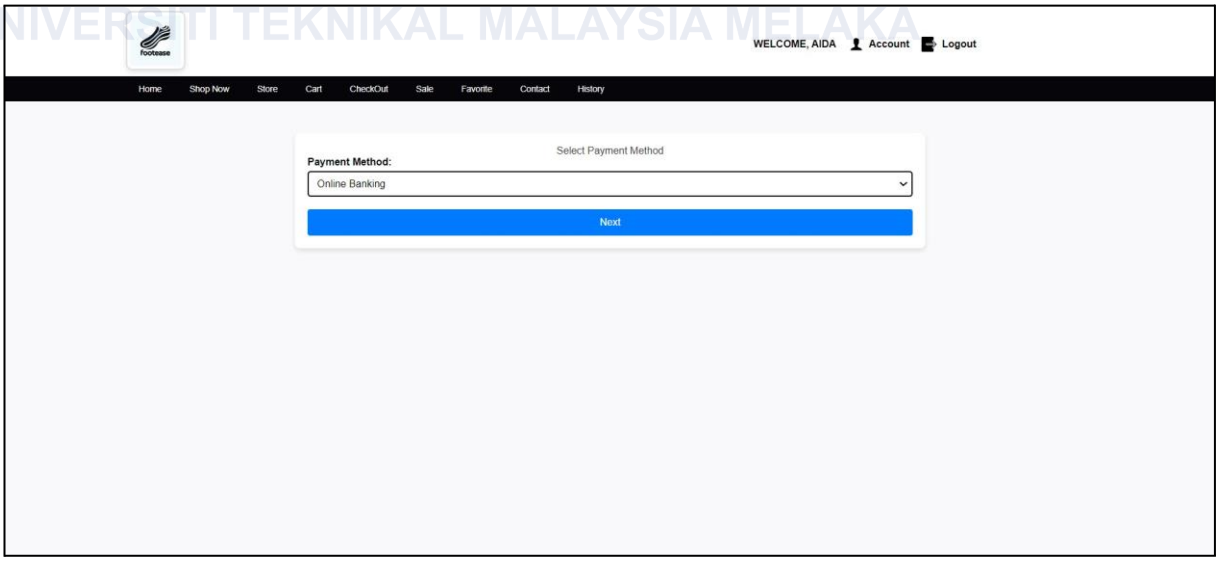


Figure 4.37 Payment Method Online Banking

online_banking.php

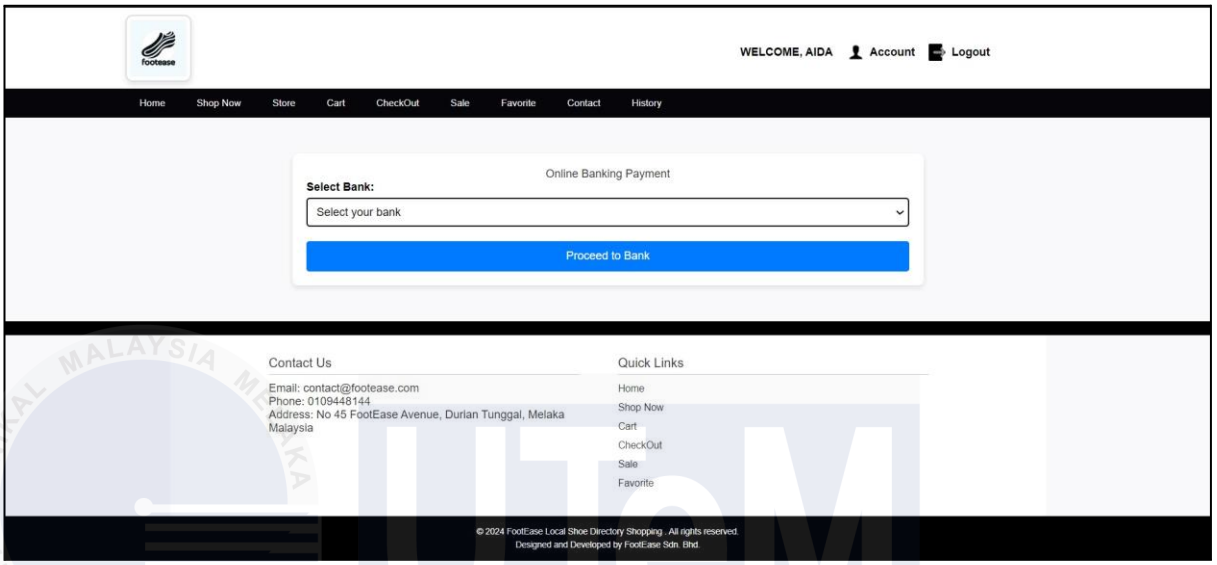


Figure 4.38 Select Bank

selecting online_banking.php

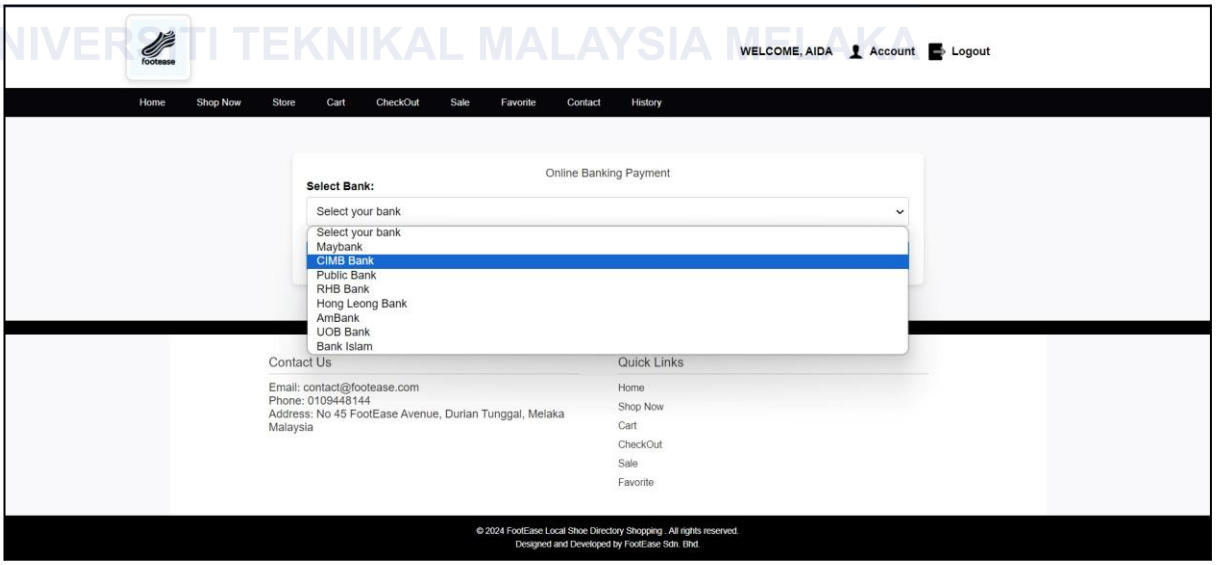


Figure 4.39 Select Bank CIMB

96

https://www.cimbclicks.com.my/

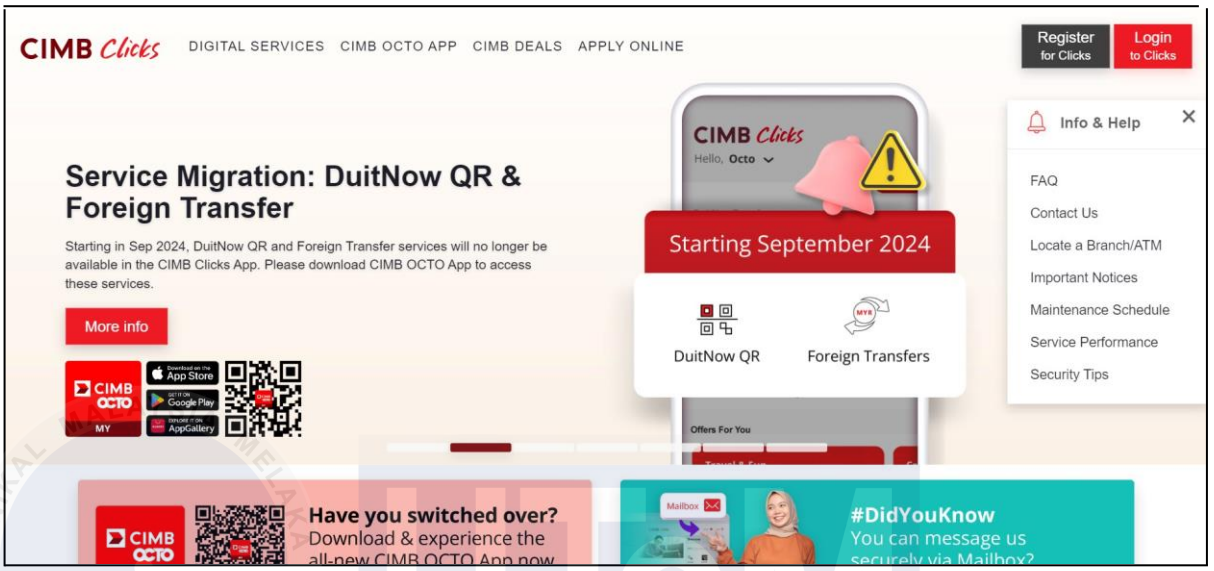


Figure 4.40 CIMB Homepage

order_confirmation.php

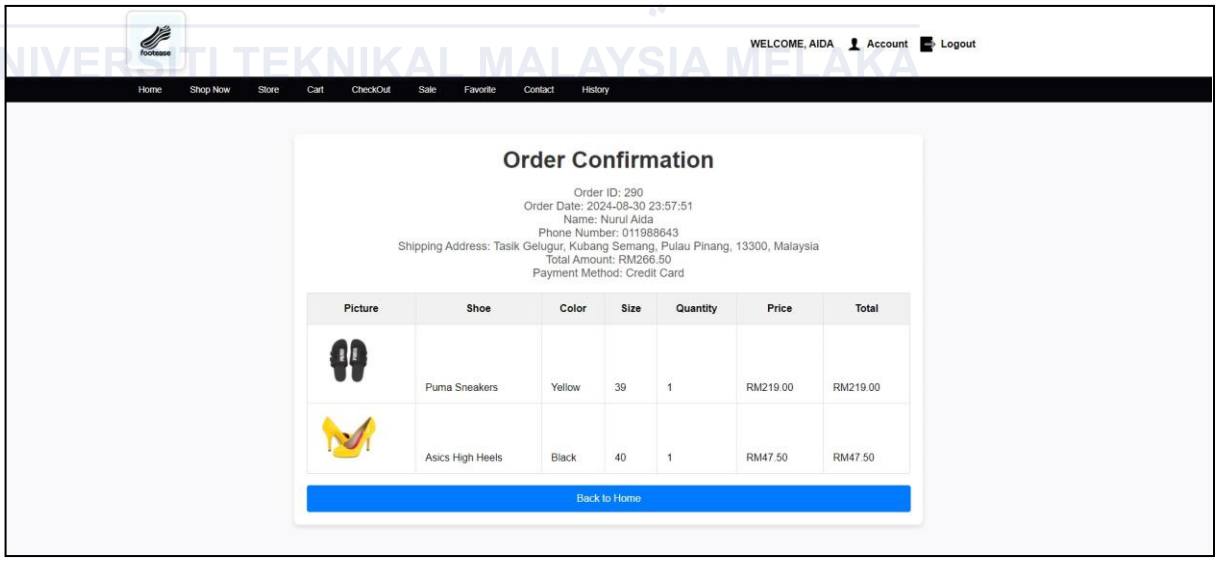


Figure 4.41 Order Confirmation

sale.php

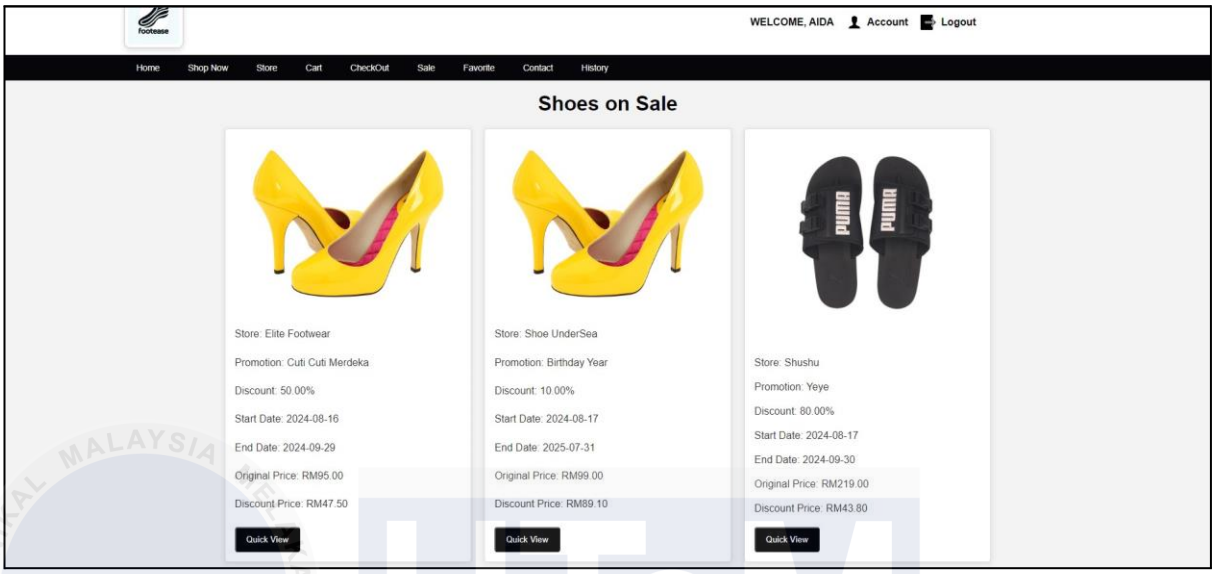


Figure 4.42 Sale

favorite.php

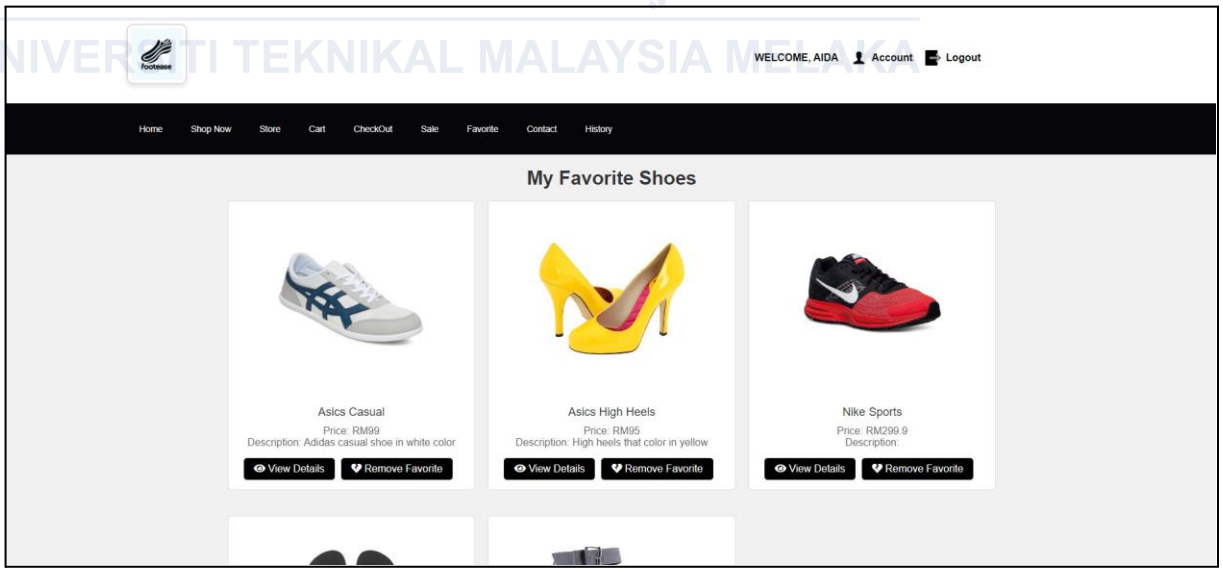


Figure 4.43 Favorite

contact.php

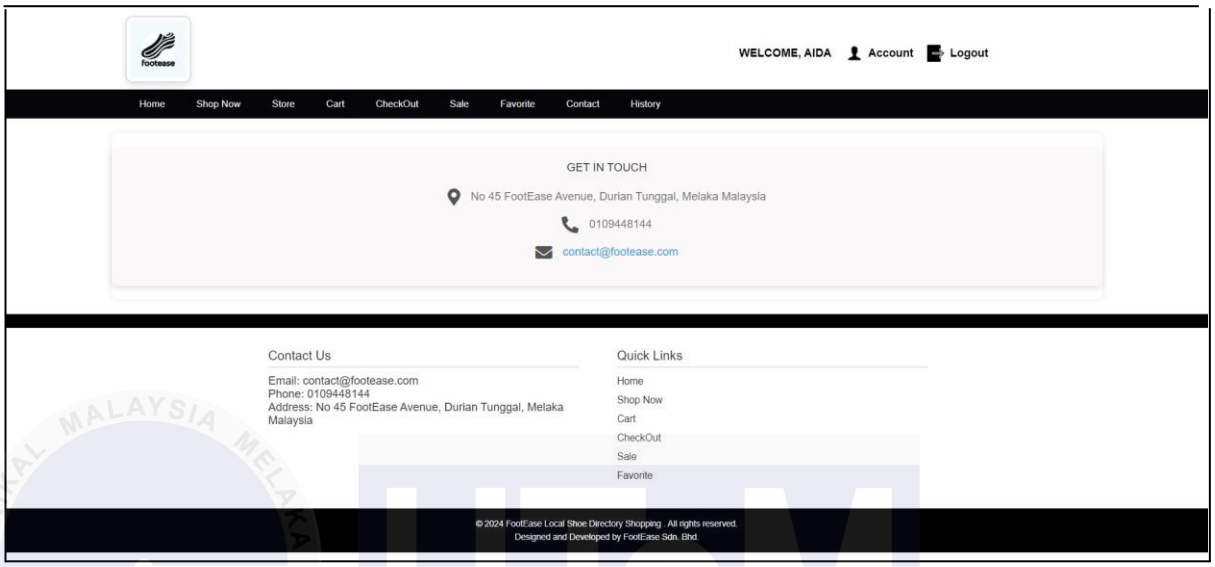


Figure 4.44 Contact

history.php

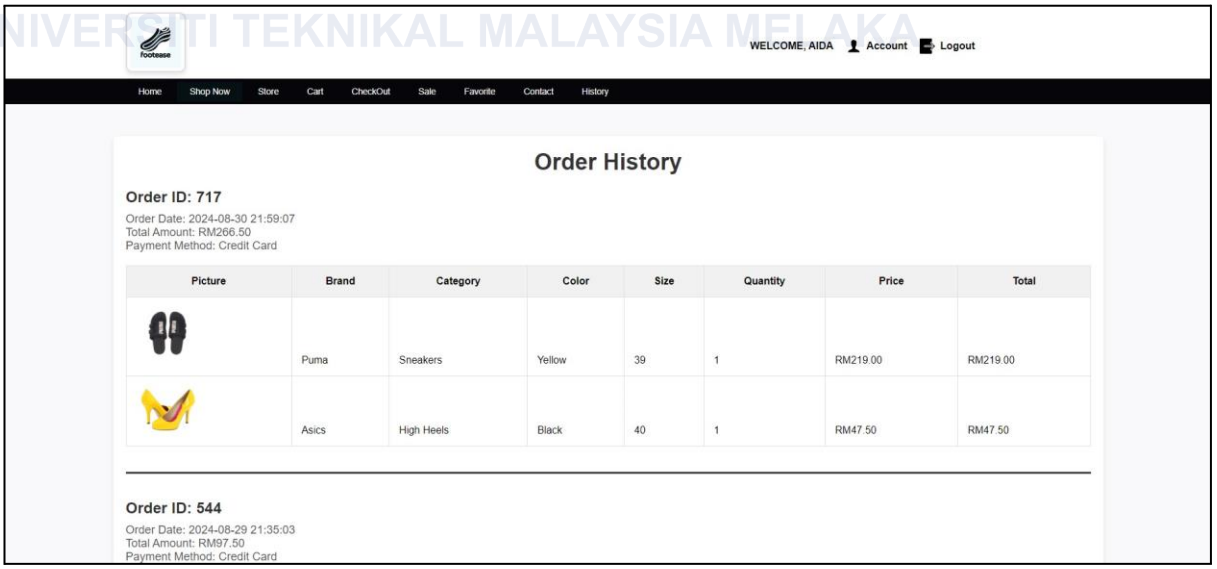


Figure 4.45 Order History

Store Owner Module

store_acc.php

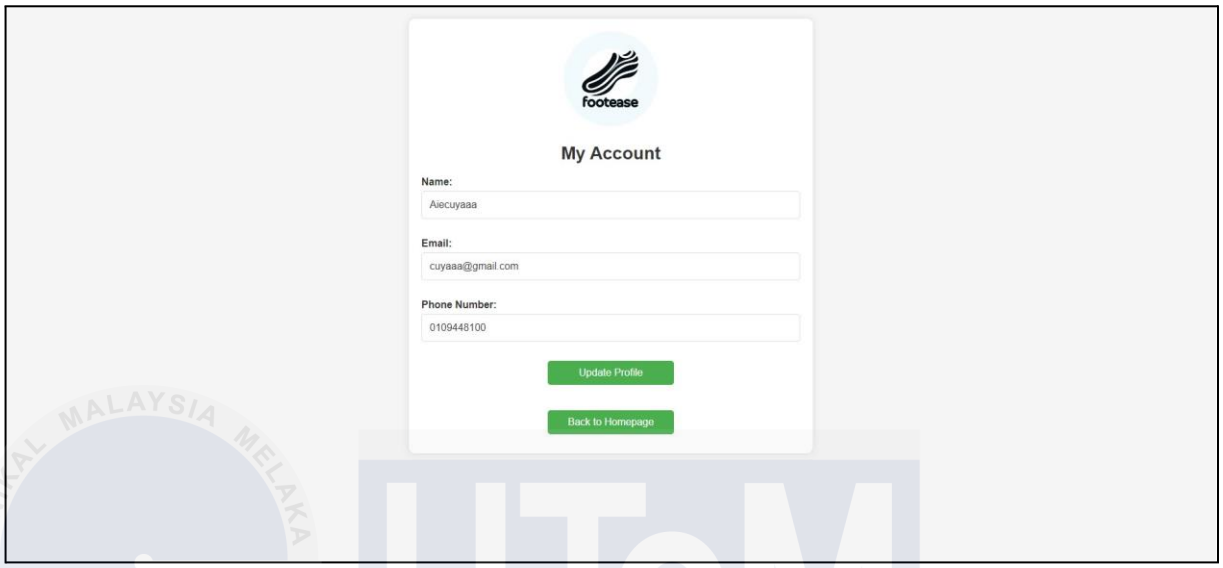


Figure 4.46 Store Owner Account

homepage_store.php

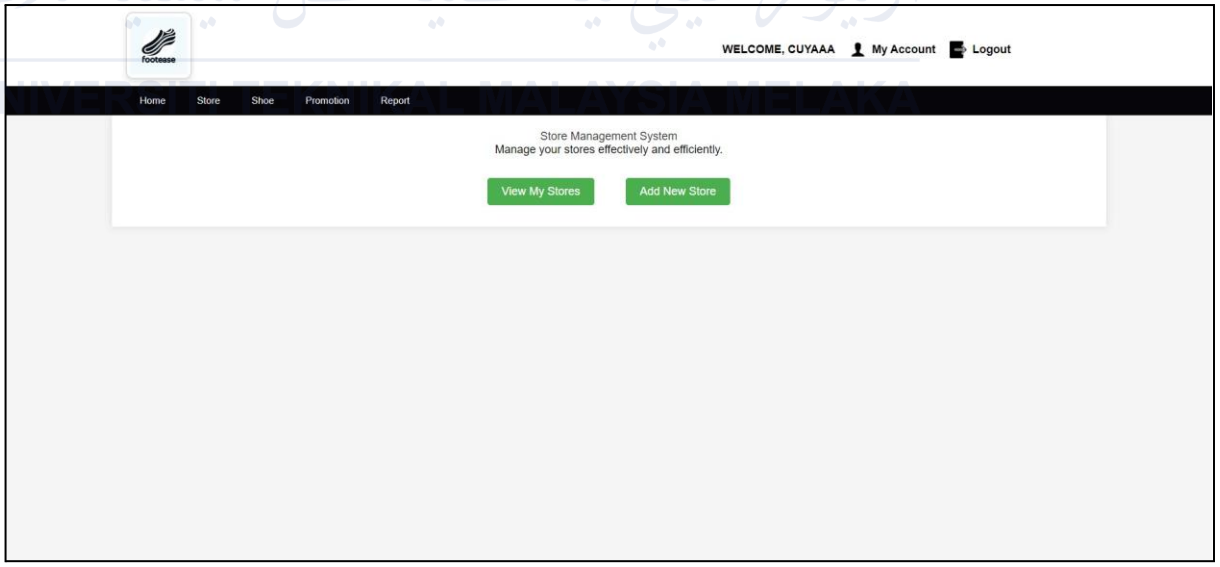


Figure 4.47 Store Owner Homepage

store_owner_index.php

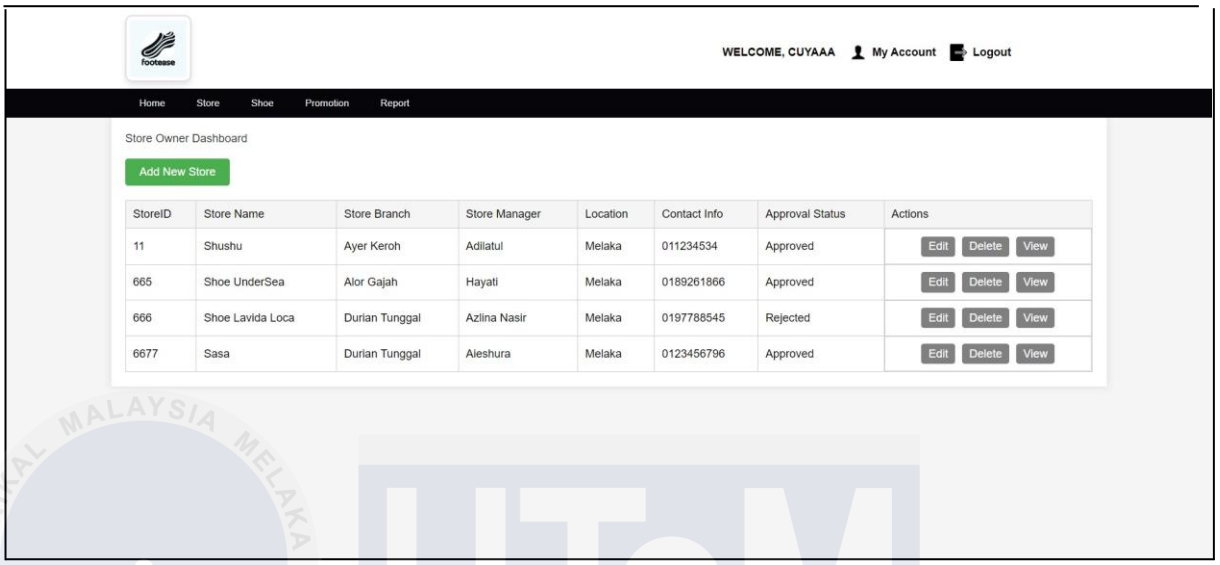


Figure 4.48 Store Management

edit_store.php

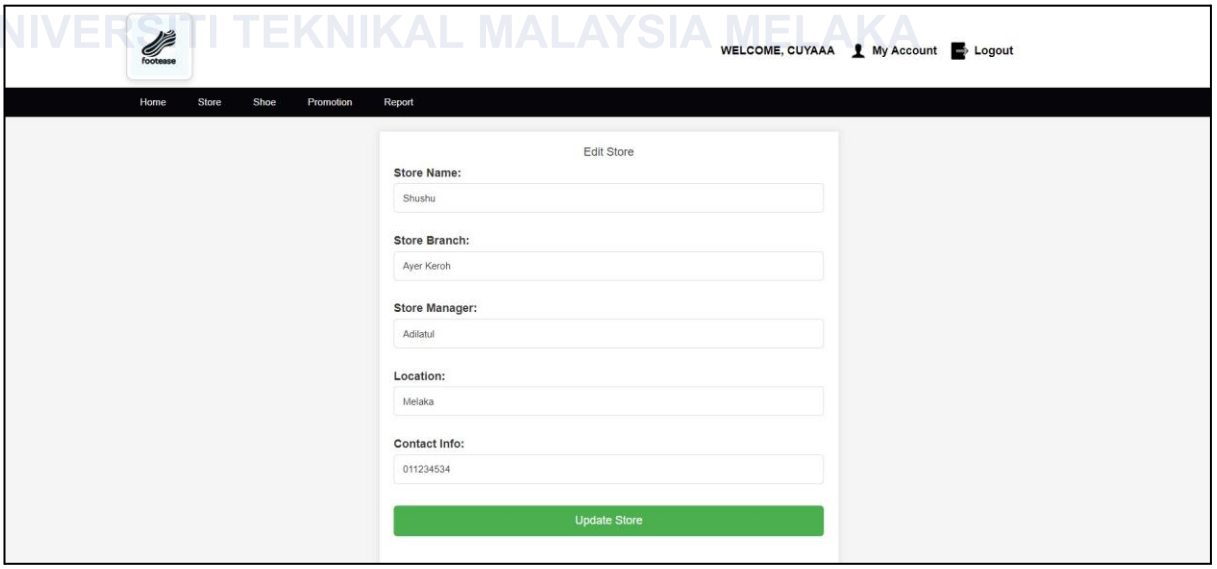


Figure 4.49 Edit Store

add_store.php

Figure 4.50 Add Store

view_shoes.php

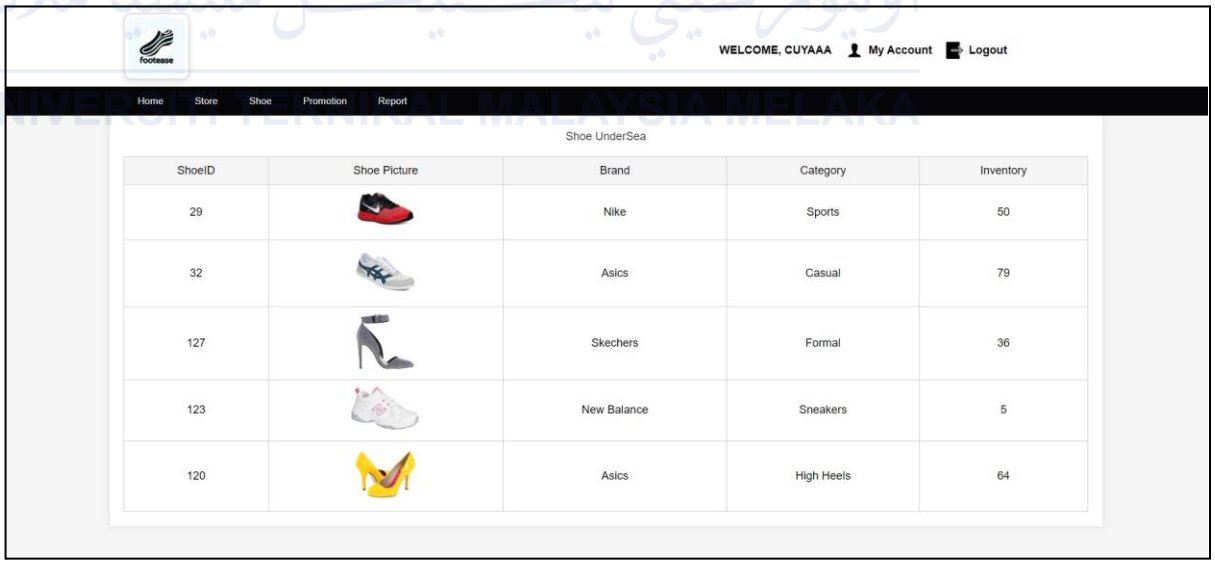


Figure 4.51 View Shoes

store_shoe.php

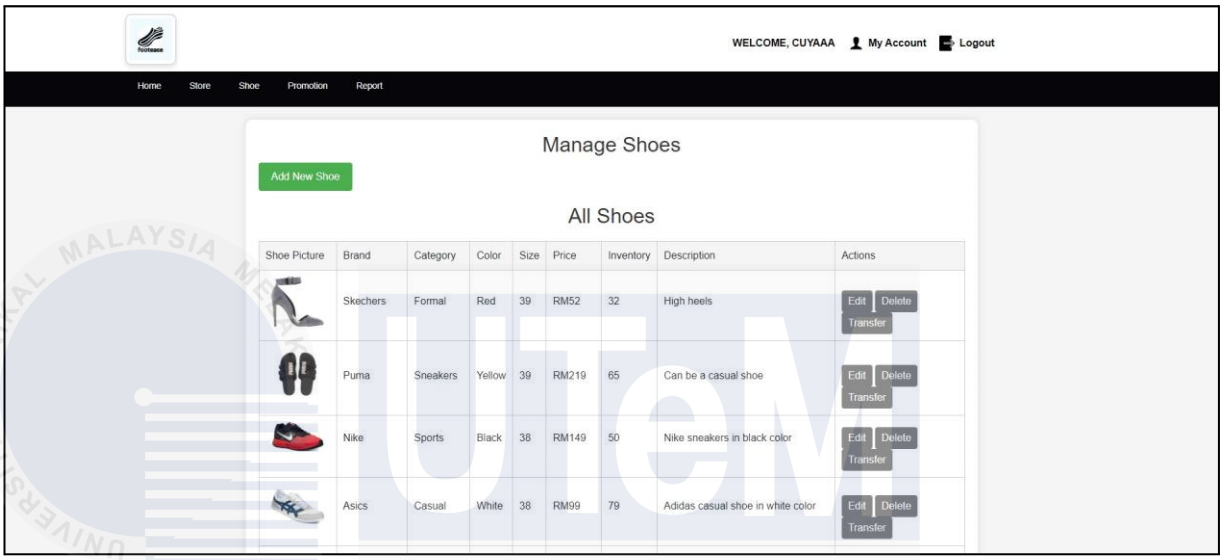


Figure 4.52 Shoe Management

add_shoe.php

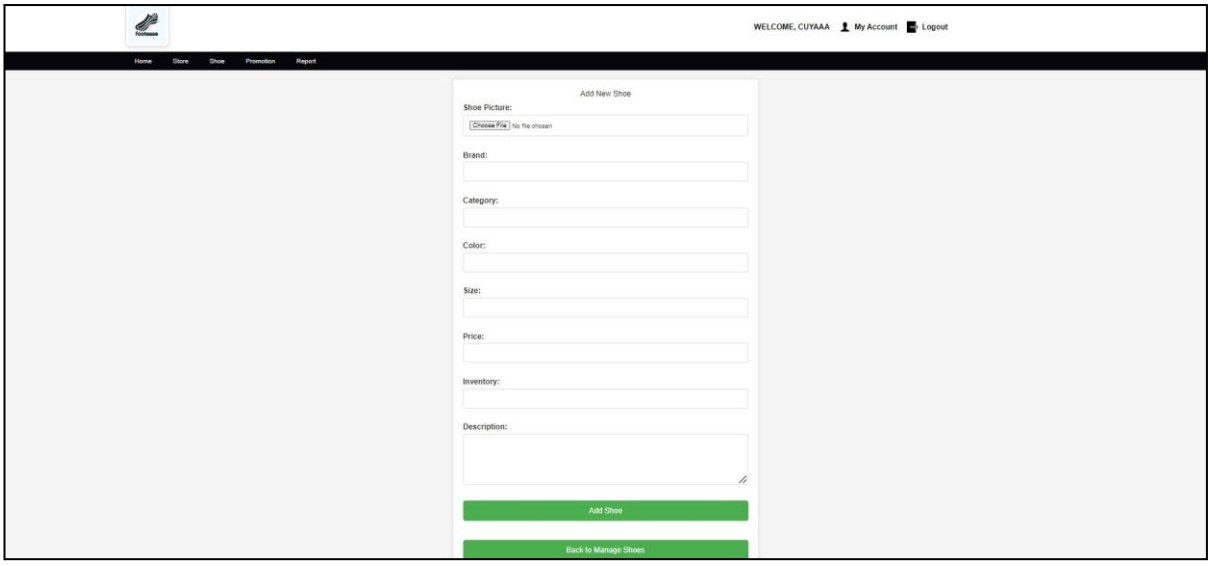


Figure 4.53 Add Shoe

edit_shoe.php

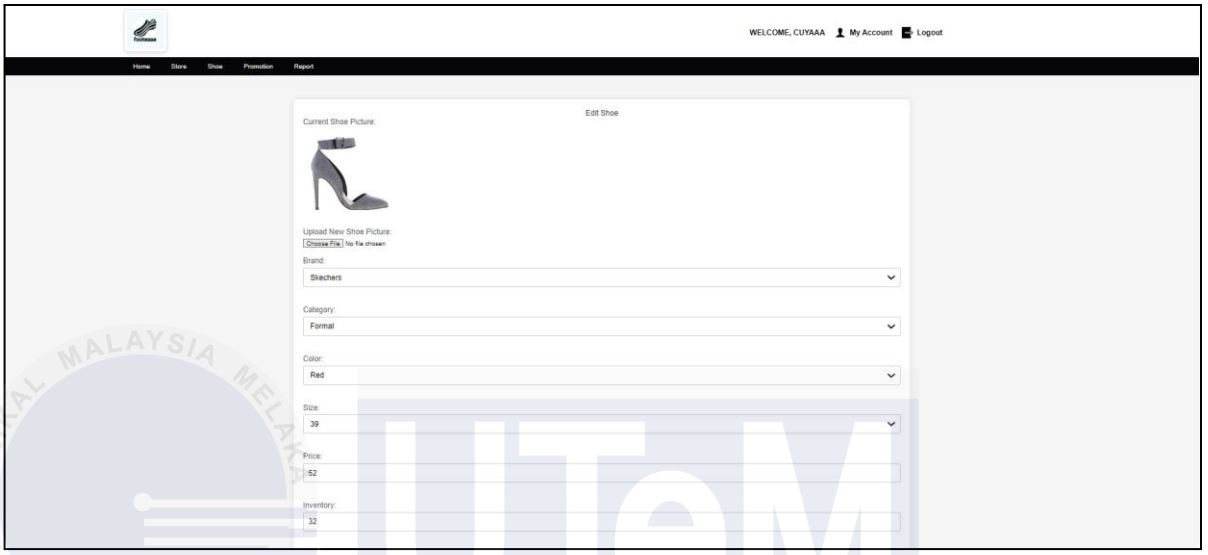


Figure 4.54 Edit Shoe

transfer_shoes.php

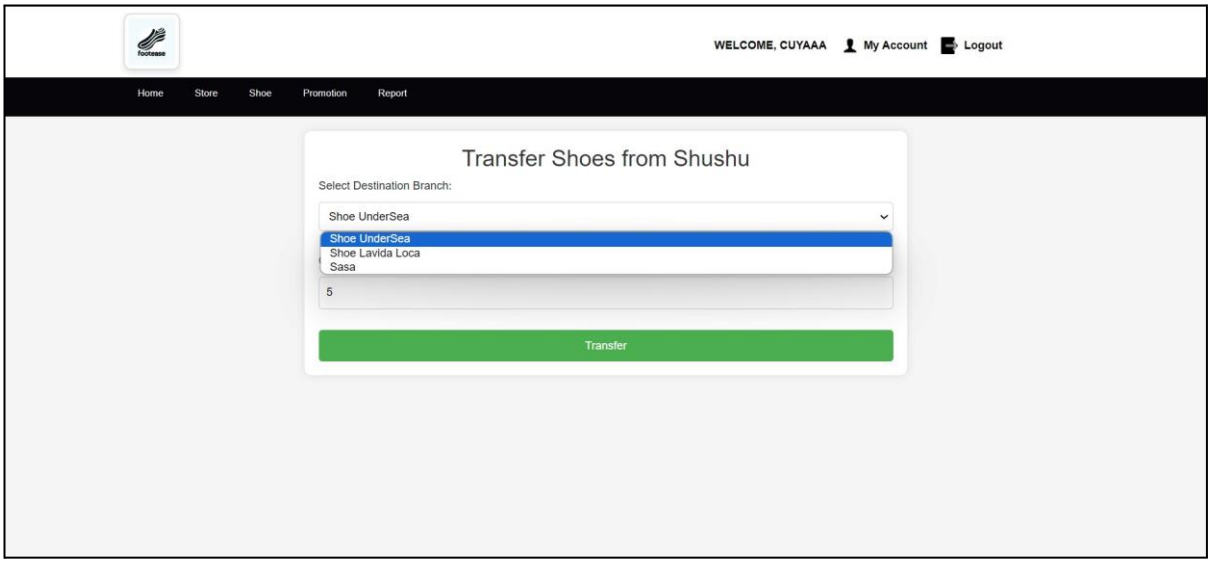


Figure 4.55 Transfer Shoe

store_promotion.php

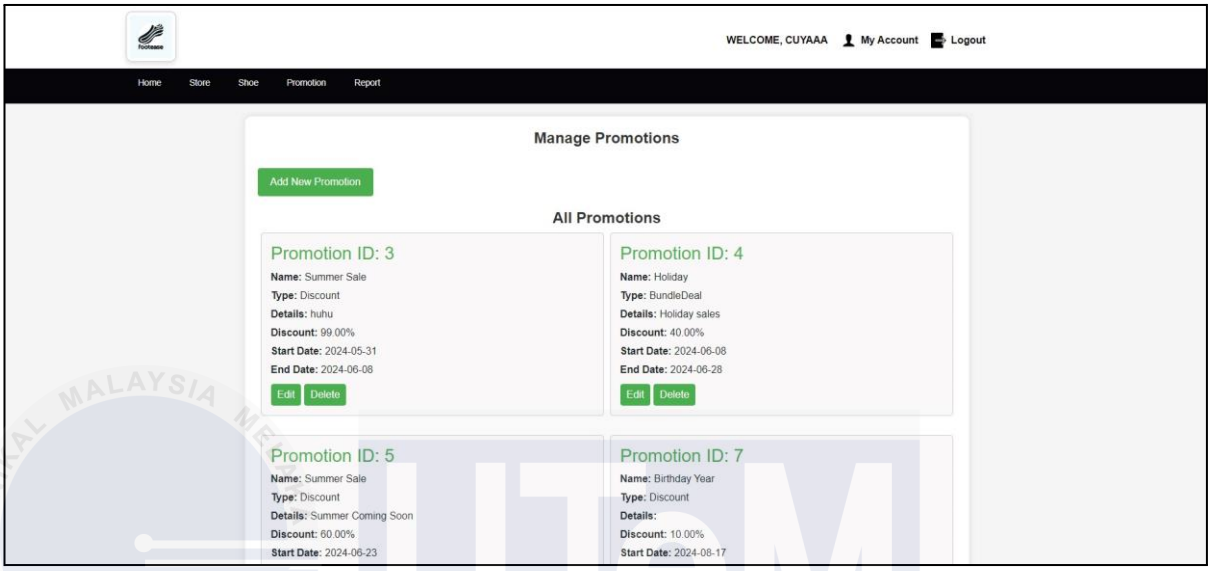


Figure 4.56 Promotion

edit_promotion.php

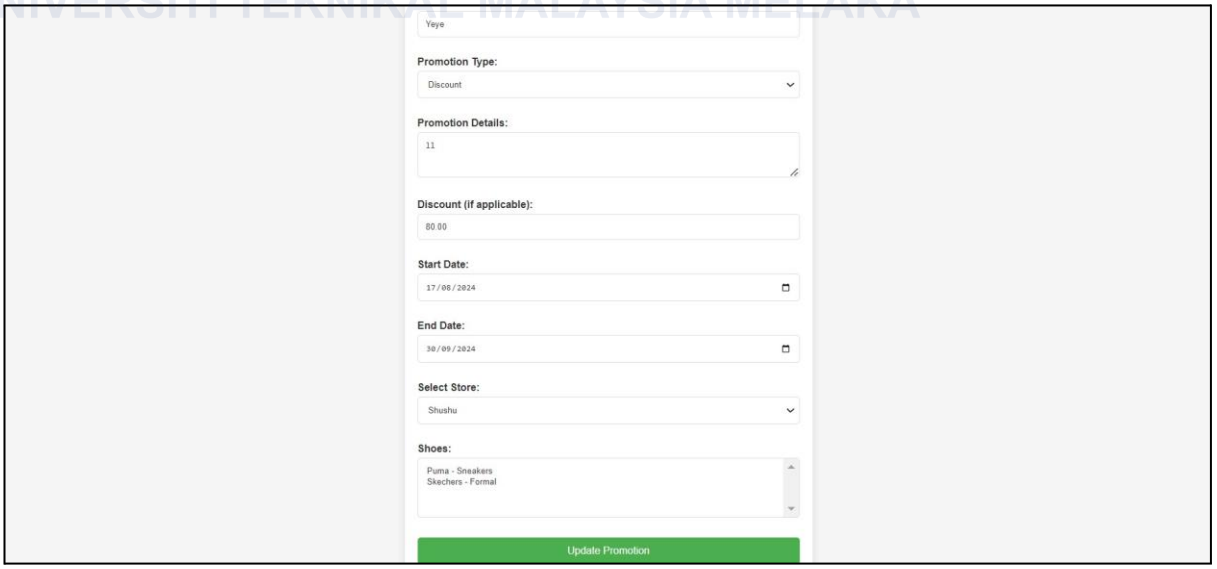


Figure 4.57 Edit Promotion

add_promotion.php

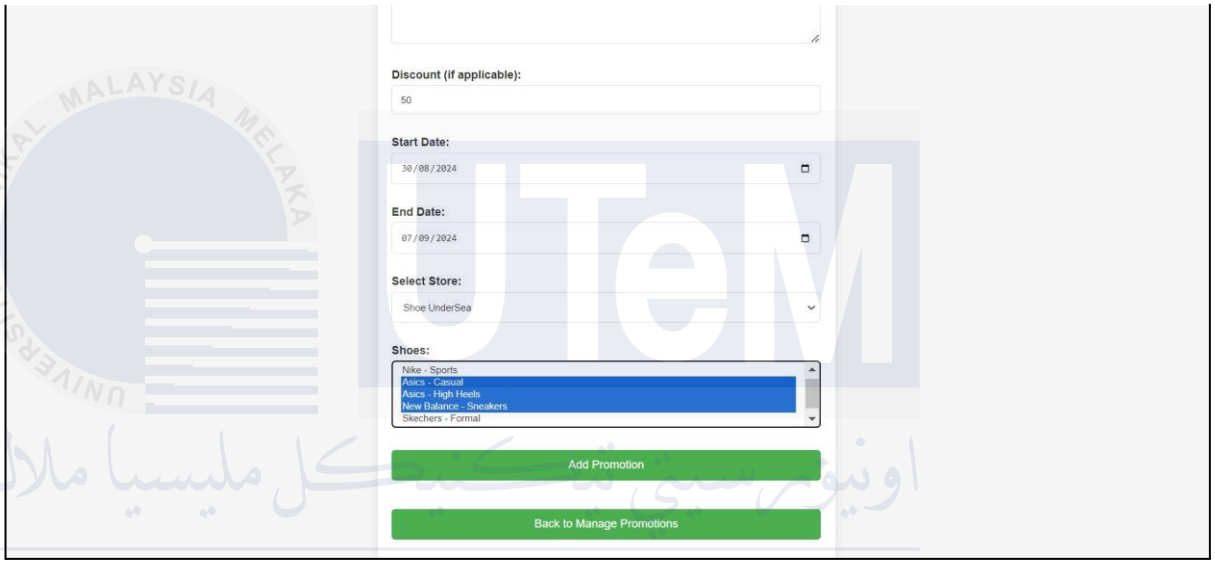


Figure 4.58 Add Promotion

report_home_store.php

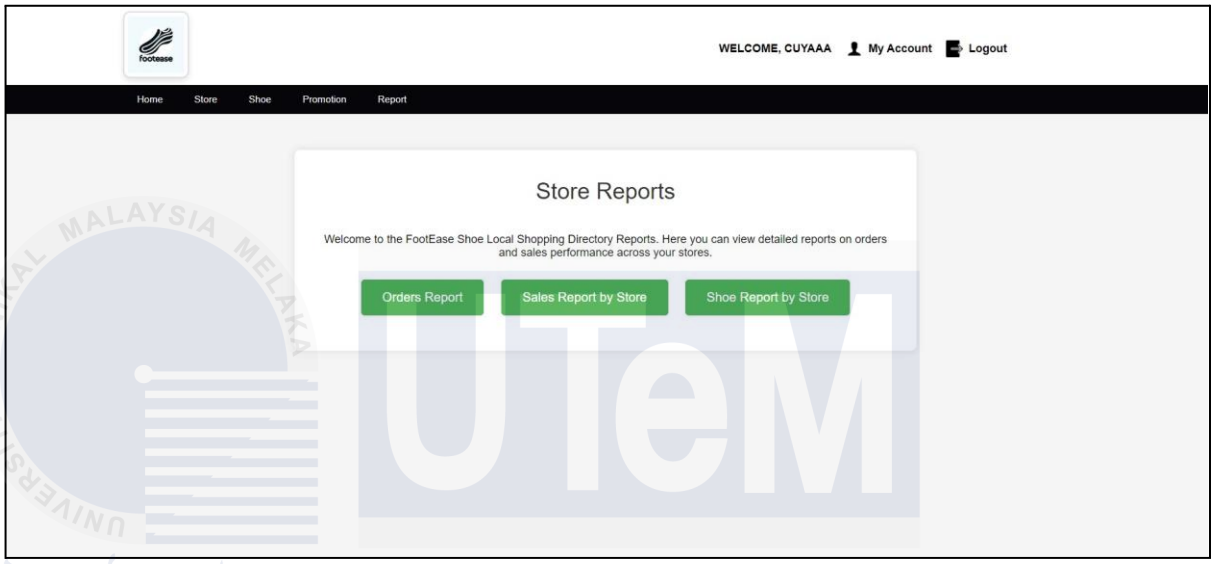


Figure 4.59 Store Reports

107

orders_report_store.php

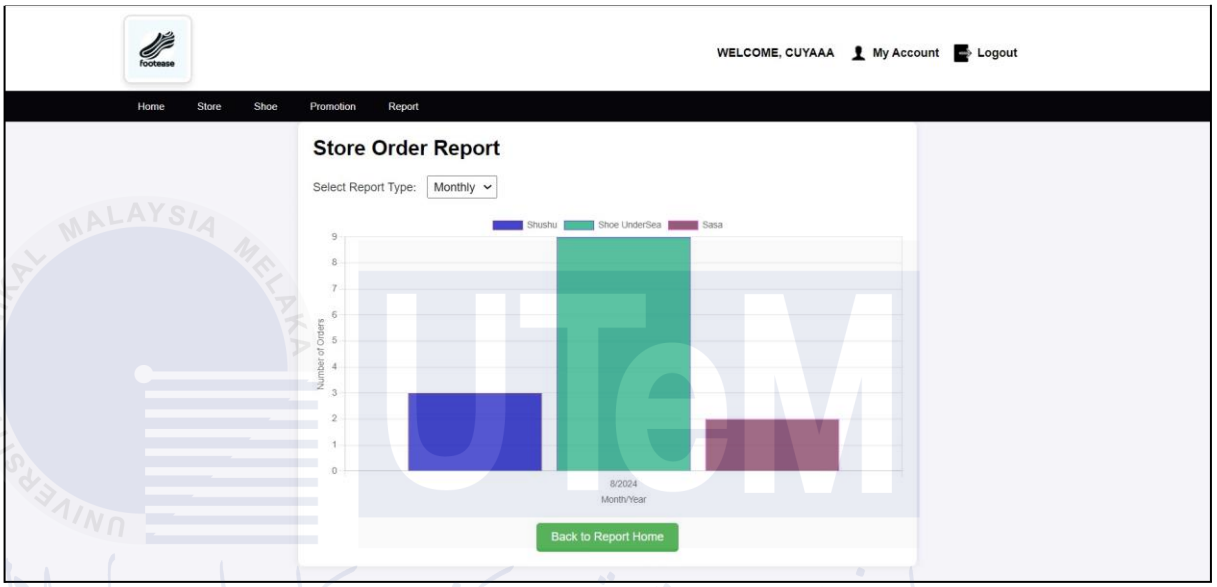


Figure 4.60 Order Reports Store

sale_report_store.php

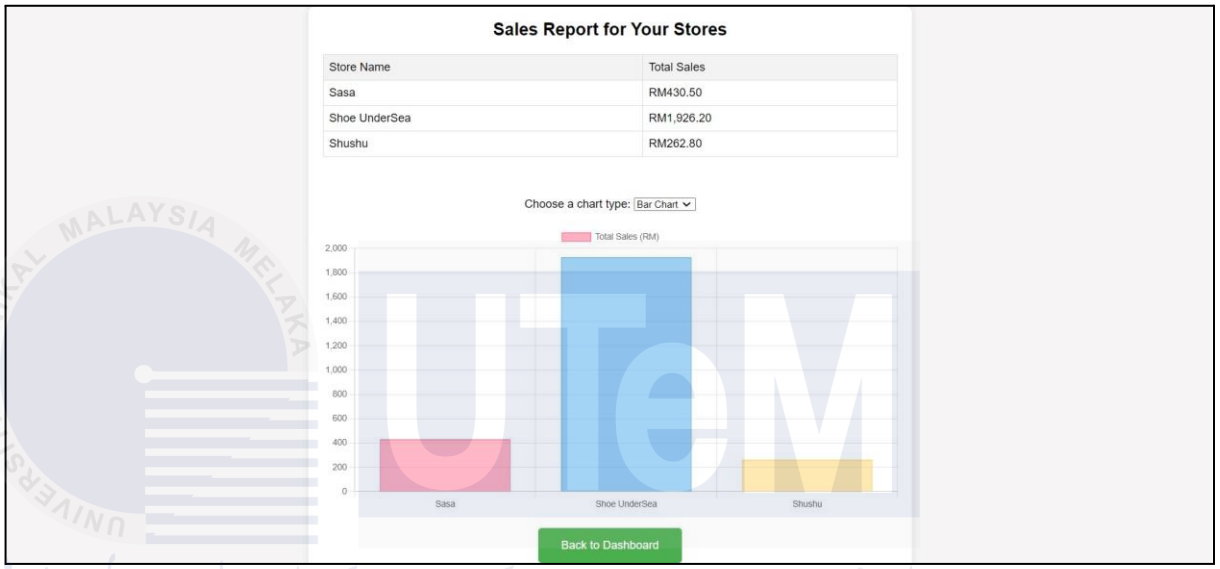


Figure 4.61 Sale Reports Store

shoe_report_store.php

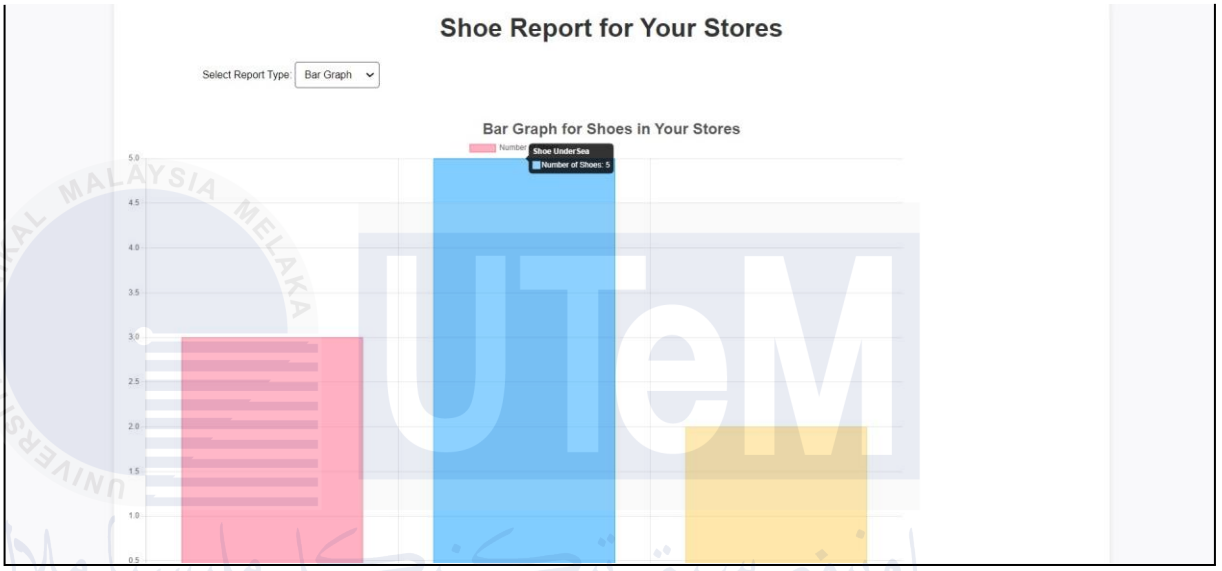


Figure 4.62 Shoe Reports Store

Admin Page

homepage_admin.php

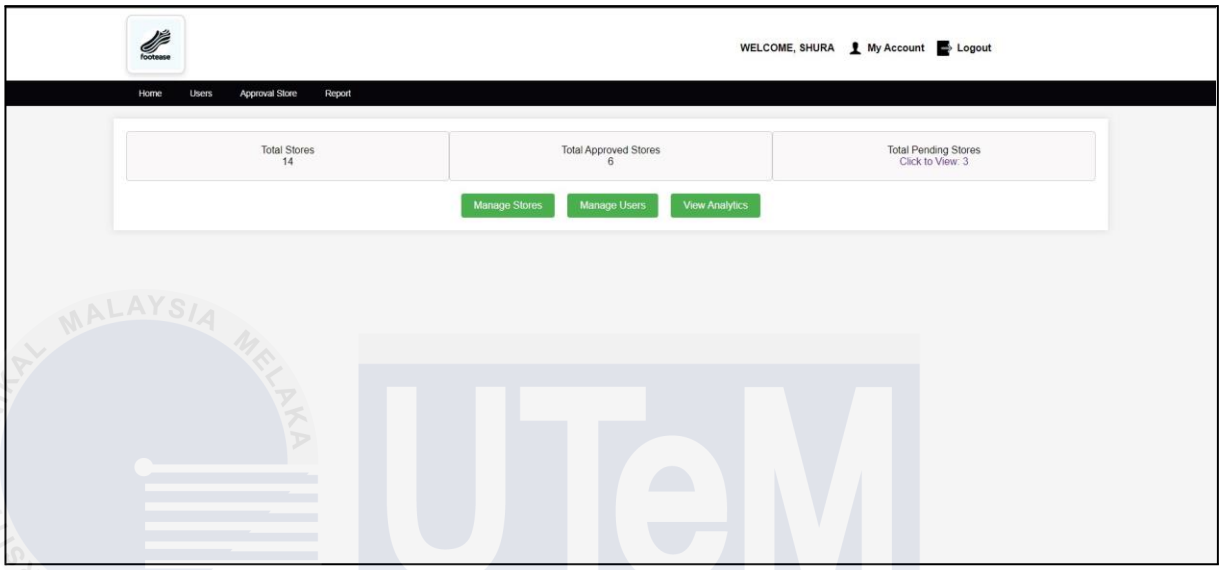


Figure 4.63 Admin Homepage

analytics.php

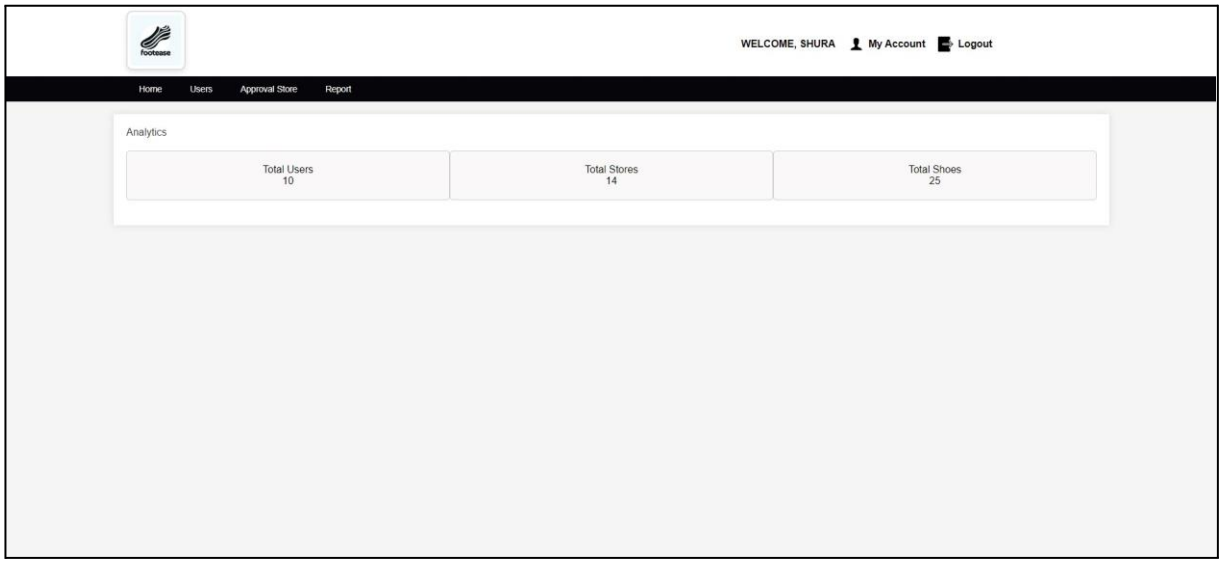


Figure 4.64 Analytics

admin.php

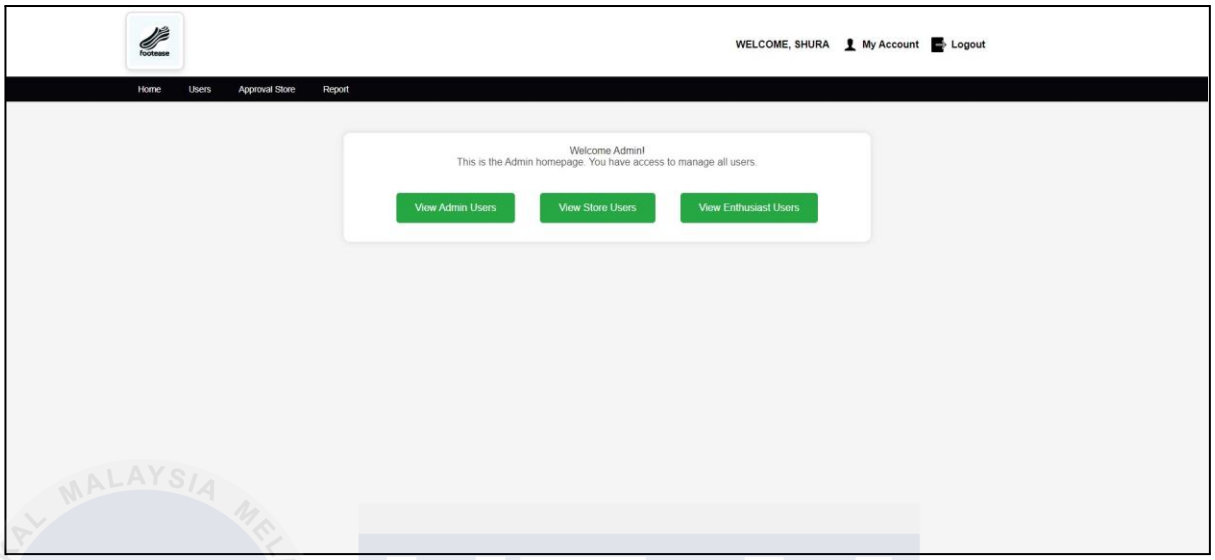


Figure 4.65 Users Management

admin_users.php

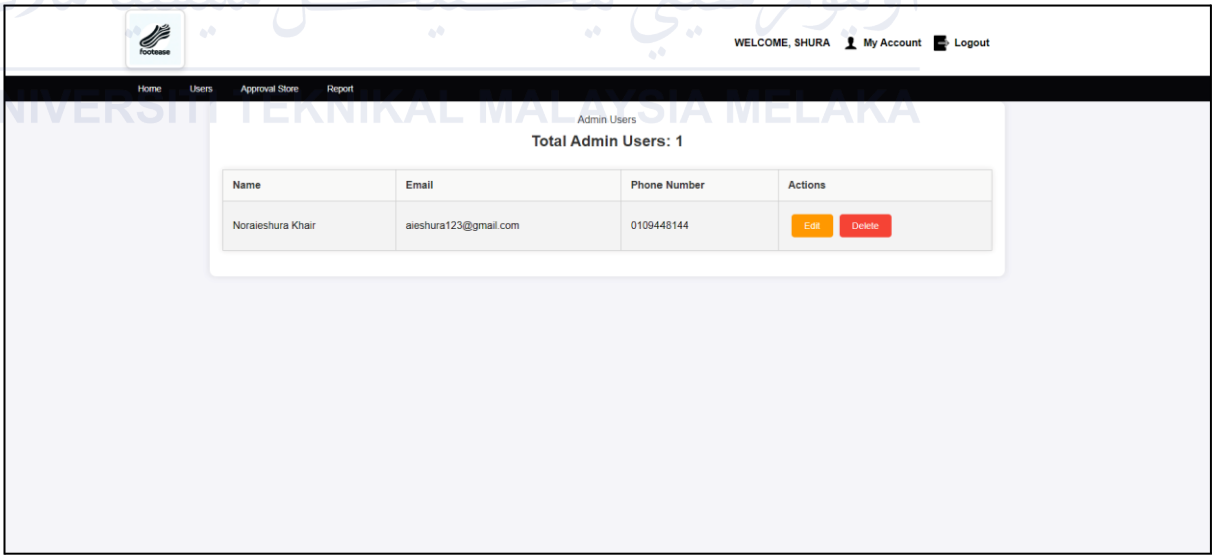


Figure 4.66 Admin Users

store_users.php

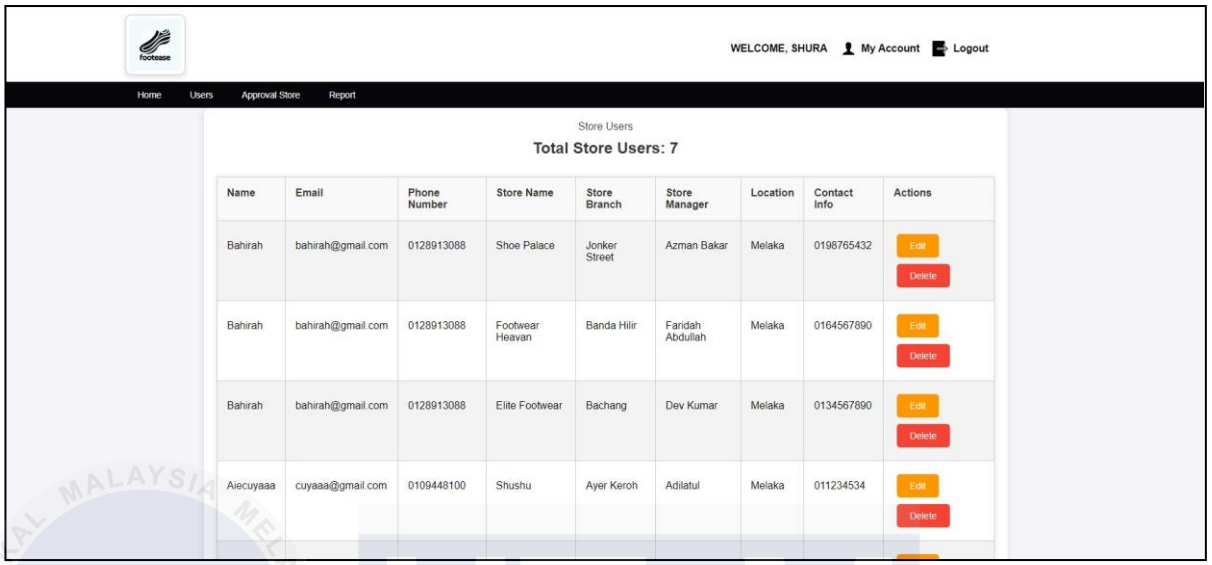


Figure 4.67 Store Owner Users

shoeenthusiast_users.php

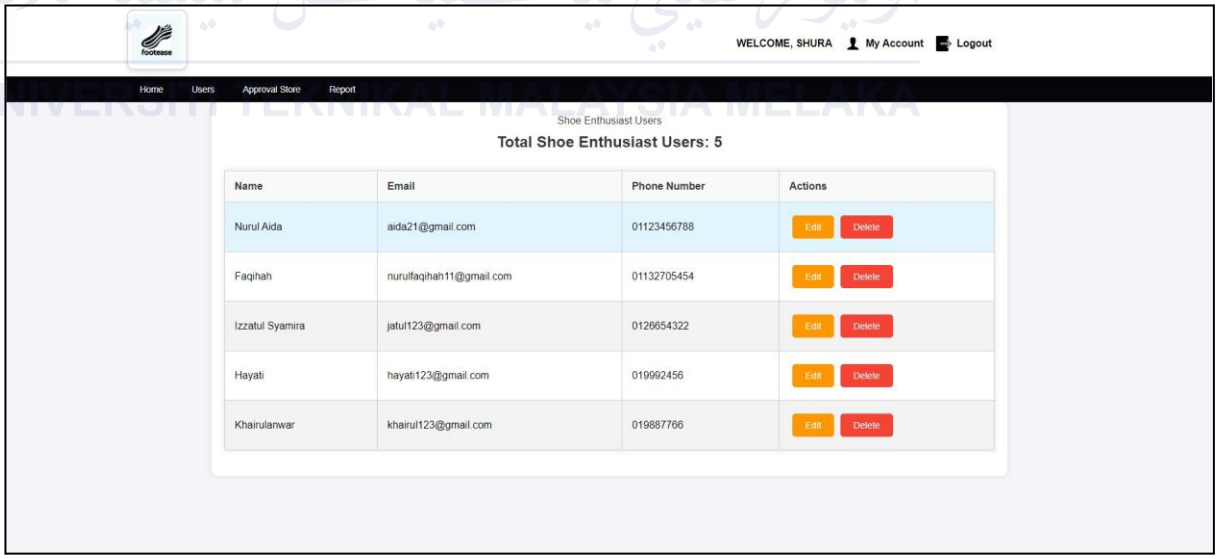


Figure 4.68 Shoe Enthusiast Users

admin_approval.php

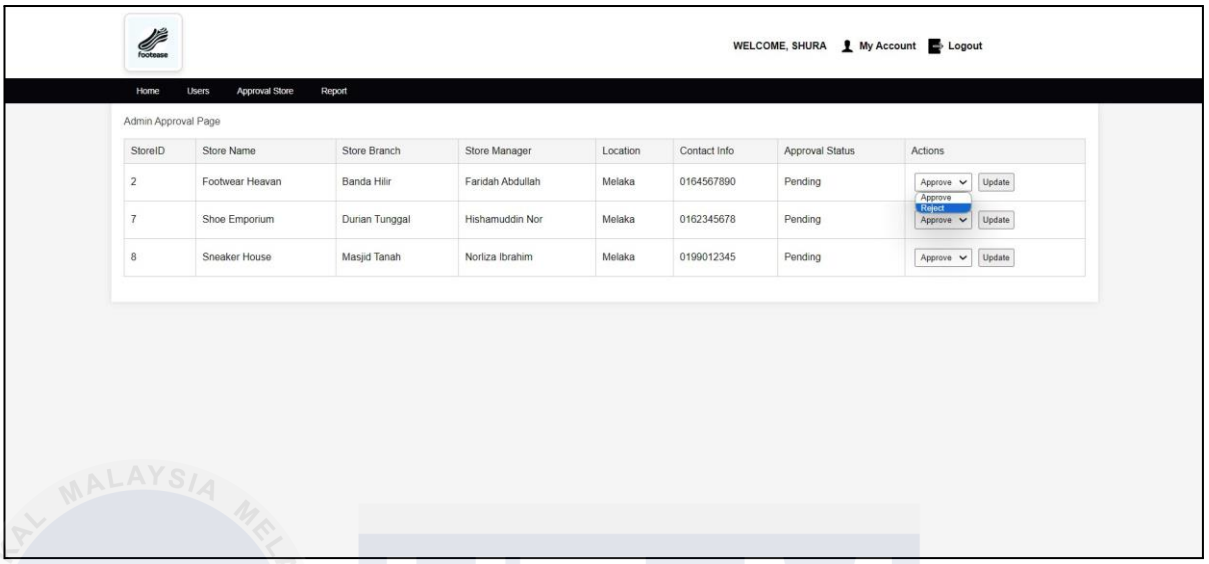


Figure 4.69 Admin Approval

admin_approval.php

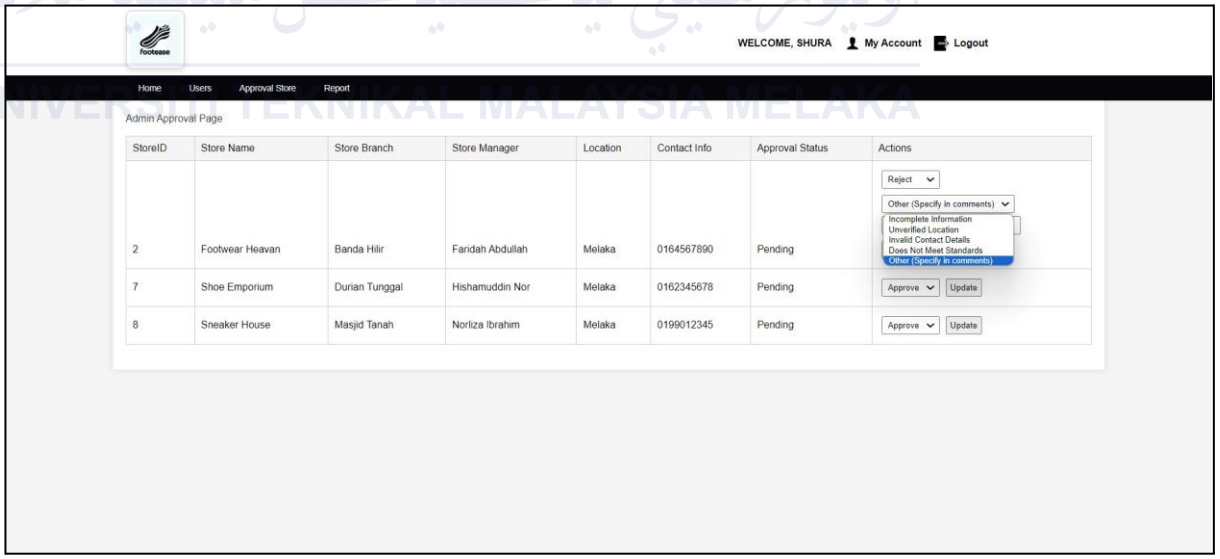


Figure 4.70 Admin Rejection

report_home.php

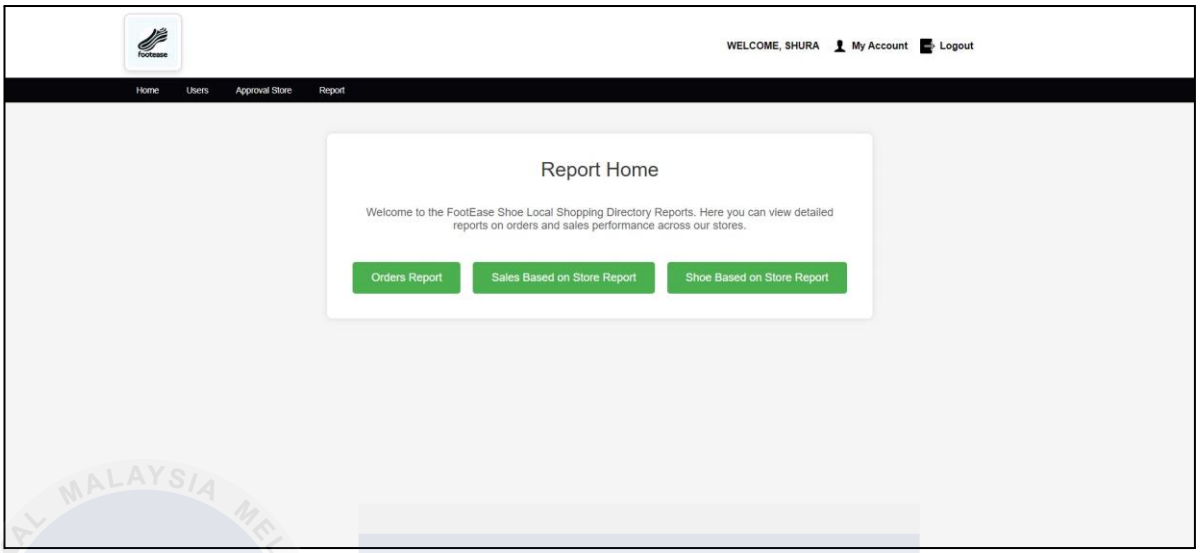


Figure 4.71 Admin Reports

orders_report.php

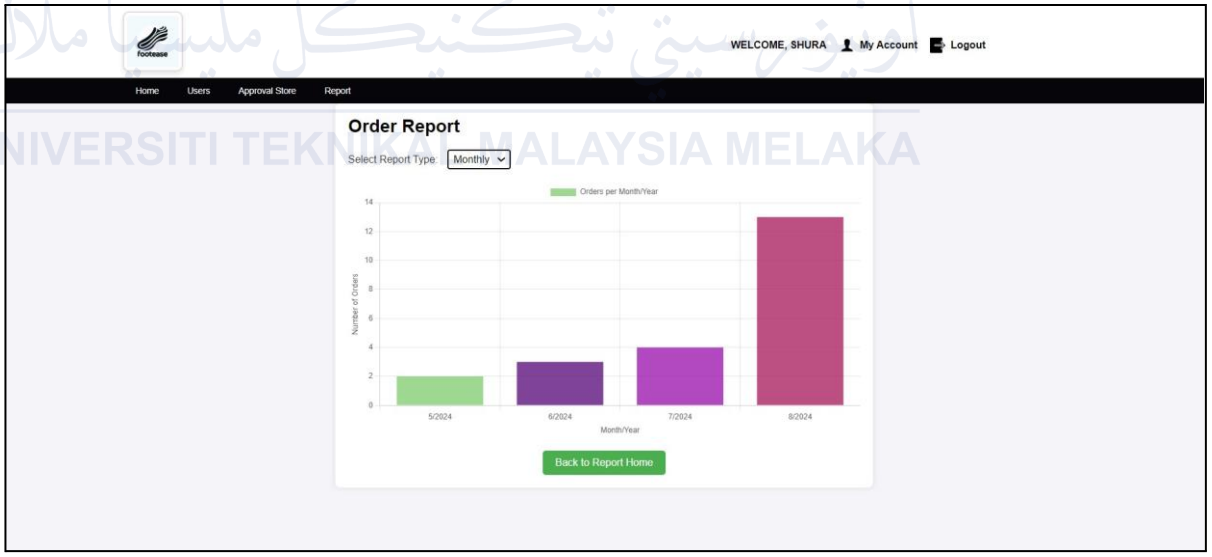


Figure 4.72 Order Report Admin Monthly

orders_report.php

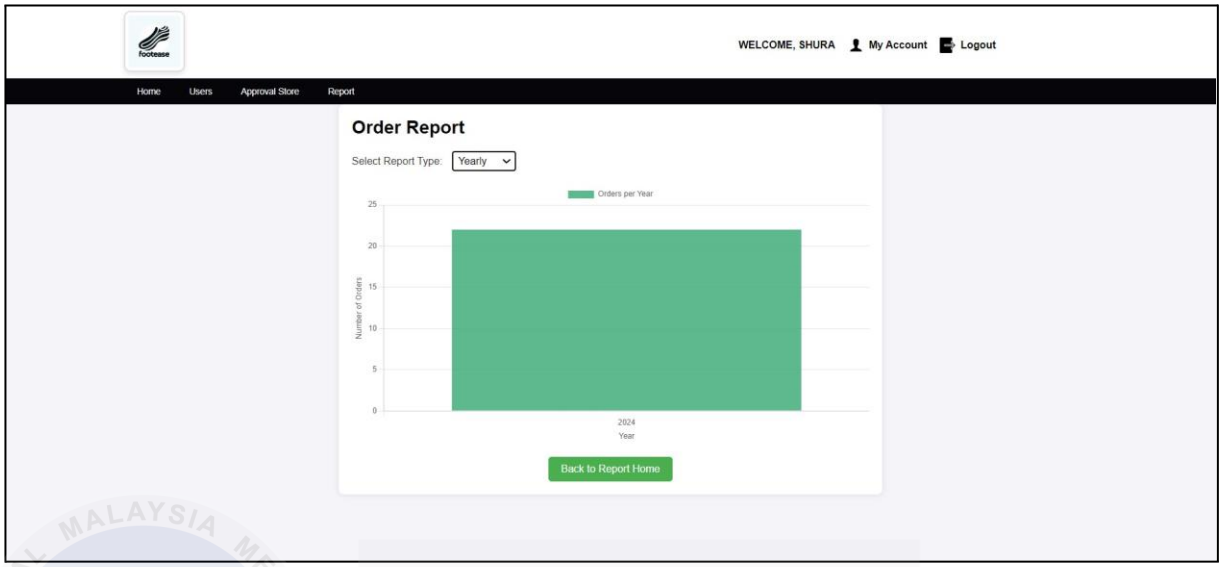


Figure 4.73 Order Report Admin Yearly

sale_report.php

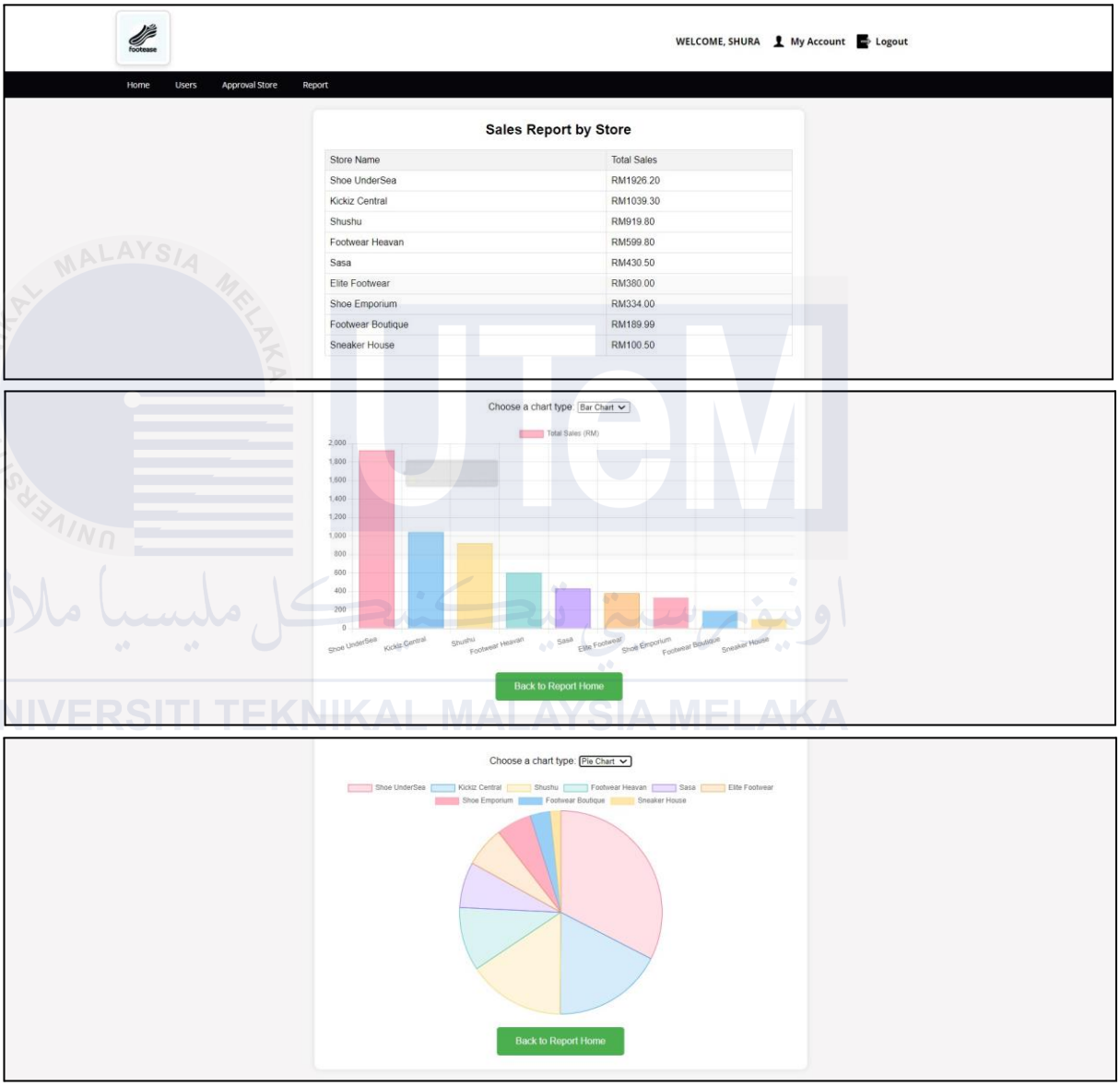


Figure 4.74 Sale Report Admin

shoe_report.php

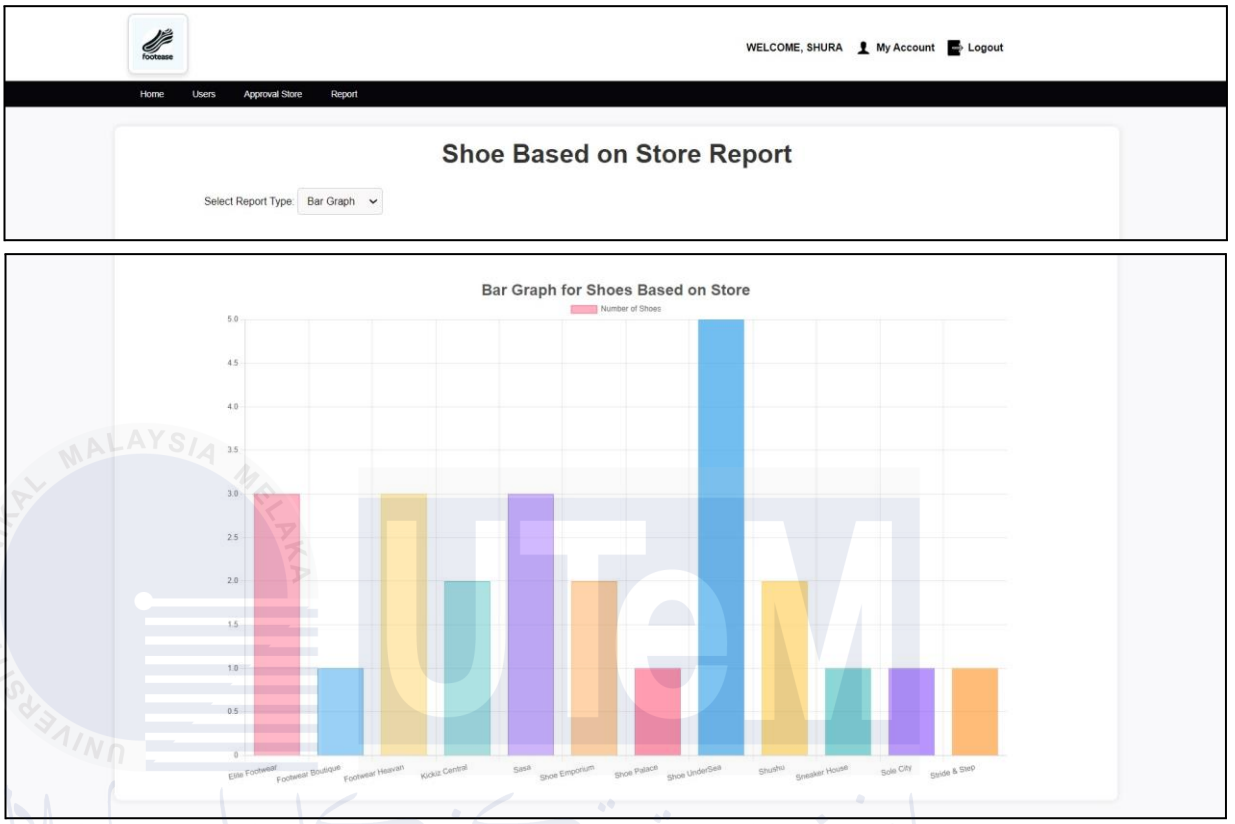


Figure 4.75 Shoe Report Admin Bar Graph

shoe_report.php

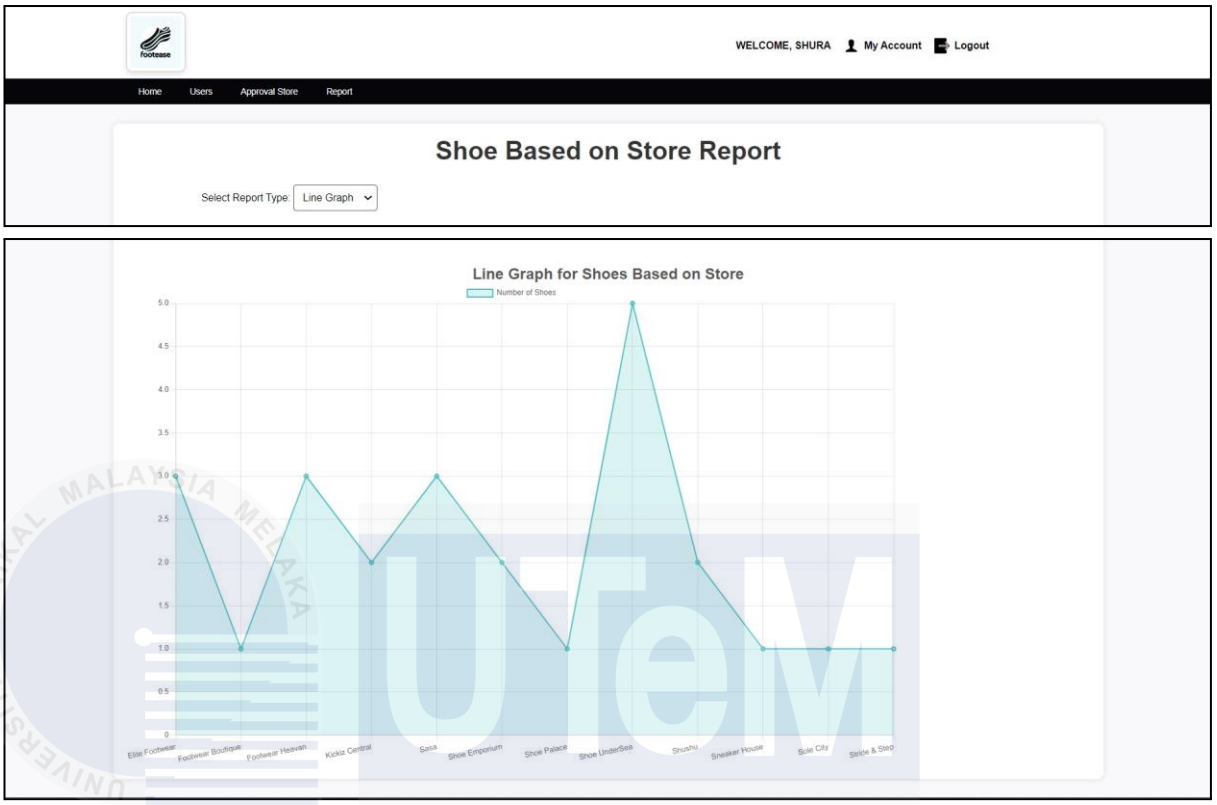


Figure 4.76 Shoe Report Admin Line Graph

4.7 Conclusion

The process of designing a database is thoroughly in Chapter 4. An introduction and a flowchart explaining the general structure are presented first. After that, the chapter explores logical database architecture, which includes a thorough data dictionary, and conceptual database design. Lastly, it discusses the data description language (DDL) and physical database design, highlighting the change from theoretical ideas to real-world application.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter covers the practical elements of putting the system's application modules into practice, with a particular focus on the functions of three important modules which are the admin, the store owner, and the shoe enthusiast and will describe how careful coding and integration may transform business requirements into workable solutions. This chapter will walk through the steps involved in creating each

module and show how to integrate certain design requirements into a functional solution. This also will go beyond simple coding to examine more general aspects of system implementation, such as how to make sure all the parts function together harmoniously to provide a productive and user-friendly experience.

5.2 Software Development Environment Setup

Software Development Environment Setup refers to the process of configuring and organizing the necessary tools and components required to develop, test, and deploy software applications efficiently. This setup ensures that developers have everything they need to write code, manage databases, and run applications in a consistent and controlled manner. The client app is the part users interact with, whether it's on the web, a mobile device, or a desktop. It needs to handle tasks like managing accounts, searching for products, and showing promotions. Using the right tools and frameworks is important for building and testing these features effectively.

The server manages the core functions and processes requests from the client app, connecting it to the database. It handles user accounts, generates reports for Admin tasks, and supports product searches, shopping, and store management for Shoe Enthusiasts and Store modules. Setting up the server involves choosing an operating system, configuring a web server, and using server-side frameworks, with APIs or endpoints for smooth data transfer.

The database stores all key information like user details, product data, and sales records. It's essential to design the database schema to support all modules and ensure it's optimized for quick access and efficient data handling.

In summary, the client app, server, and database must be set up and linked correctly. The client app interacts with users, the server handles requests and business logic, and the database organizes and stores data. Proper setup and testing of these parts ensure that the application functions well for users, store management, and administrative tasks.

5.2.1 Software Development Environment Setup for Comprehensive Module Integration

1. **Define Development Requirements**

- **Admin Modules**: Includes user management, store management, and reporting functionalities. The environment should support features for managing user accounts, handling store data, and generating various reports.
- **Shoe Enthusiasts Modules**: Includes user registration, shoe discovery, shopping, and promotions. The environment should be equipped to handle user interactions, search functionality, and shopping processes.

- **Store Modules**: Involves store registration, shoe management, and promotions. Ensure the environment supports store-specific operations, including inventory and promotion management.

2. Hardware and Software Requirements

Hardware:

- **Servers:** You'll need strong servers to manage back-end processes, handle databases, and efficiently run reporting tasks.
- **Development Workstations:** These are the computers you'll use to develop and test your project. They should be powerful enough to run your development tools smoothly.

Software:

- **Operating System:** Use Windows 11 as the main operating system for your workstations.
- **Development Tools:**
    - **IDE (Integrated Development Environment):** Use Microsoft Visual Studio Code for writing and editing your code.
    - **Version Control System:** Use Git to manage your source code and collaborate with others
- **Database Management System:**
    - **SQL Databases:** MySQL is used to manage and store your relational data.
- **Web Server:** Apache is the web server that hosts your website or application.

3. Development Environment Configuration

Project Structure:

- **Create Repositories**: Set up version-controlled repositories for each module.
    - **Admin Modules**: Include repositories for managing user accounts, store management, and reports.
    - **Shoe Enthusiasts Modules**: Repositories for user registration, search, shopping functionalities, and promotions.
    - **Store Modules**: Repositories for store management, shoe inventory, and promotions.

Environment Setup:

- **Local Development**: Configure local servers and databases to mirror production environments as closely as possible.
- **Dependencies**: Install necessary libraries and frameworks for each module.
    - **Admin Modules**: Dependencies for user management, reporting libraries.
    - **Shoe Enthusiasts Modules**: Libraries for search functionality, user authentication, and shopping carts.
    - **Store Modules**: Dependencies for store management, inventory handling, and promotions.
- **Configuration Files**: Set up configuration files (.env, config.json) to manage environment-specific settings such as database connections and API keys.

5.3 Database Implementation

To use a working database for your application through XAMPP Apache Mysql PHP, one first needs to install XAMPP, this is a package that contains all the necessities for a developer to set the development environment. This is made up of Apache, MySQL and PHP thus when you want to manage these components on your local machine then you do it through XAMPP. After the installation of xampp, Apache and MySQL needs to be set up manually by opening the xampp control panel and running these two services. Apache will be responsible for web server kinds of tasks while MySQL will be responsible for all database related tasks.

Second, you will proceed on creating and configuring a database through phpMyAdmin which is also available in the XAMPP project. Access phpMyAdmin through your browser at `http:>`http://localhost/phpmyadmin` and then create a new database ”foot” in which you will input data needed for your application. Type in your database schema from the table layouts and their linkage basing yourself on the needs

of your Admin, Shoe Enthusiasts, and Store. For instance, the tables one could build include, the user management table, store information table, shoe stock table, and promotion table. These tables should be pre-filled with data using a tool: SQL or phpMyAdmin (using insert data feature).

Basically, incorporating the database with the PHP involves the use of scripts to manage the relationship between your application and the database. Use PHP scripts to implement purely functional methods for querying, updating, adding and removing data in the database. For instance, connect a database to an html using php with a script that contains parameters such as the server name, user name and the name of the database. More PHP code connecting to the database and for executing more SQL queries to for read or write or any kind required by the application depending on its modules.

It should also be noted that testing and debugging of the scripts written in PHP dialect is a mandatory step in order to make sure that they can interact with the database in the desired manner. Testing can also follow the methodology whereby a user checks whether data handling is in order by inputting additional records, altering some entries and removing other records to ensure that all functions properly and use debugging tools plus error messages to solve any occurrences. Further, incorporate security solutions as a part of the database, for example, sanitize the user's input to avoid SQL injections or manage users' privileges in the database.

This is a snippet code PHP to connect to MySQL

```
<?php
$servername = "localhost";
$username = "root"; // default XAMPP username
$password = ""; // default XAMPP password (blank) $dbname
= "foot";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection if
($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

5.3.1 Procedures and Triggers

1. Procedures

A precompiled collection of one or more SQL statements stored in a database. Stored procedures can perform operations like querying data, inserting new records, updating existing records, or deleting data.

a) getShoeData

The `getShoeData` procedure is designed to retrieve comprehensive information from several key tables in a shoe-related database, specifically the `Size`, `Category`, `Brand`, `Color`, and `Store` tables. When executed, this stored procedure consolidates data from these tables, providing detailed information about each shoe, including its size, category (such as sneakers or boots), brand, color, and the store where it is available. By combining data from these related tables into a single, unified dataset, `getShoeData` simplifies the process of fetching and displaying all relevant shoe information in an application, eliminating the need to write multiple SQL queries each time this data is required.

Figure 5.1 getShoeData

b) getShoePurchaseData

The `getShoePurchaseData` procedure is designed to retrieve and display all relevant shoe information after a successful payment has been made. Specifically, it gathers data from the `Size`, `Category`, `Brand`, `Color`, and `Store` tables. Once a payment is completed, this procedure consolidates details about the purchased shoe, including its size, category (like sneakers or boots), brand, color, and the store from which it was bought. This information can then be displayed to the customer, providing them with a complete overview of their purchase.

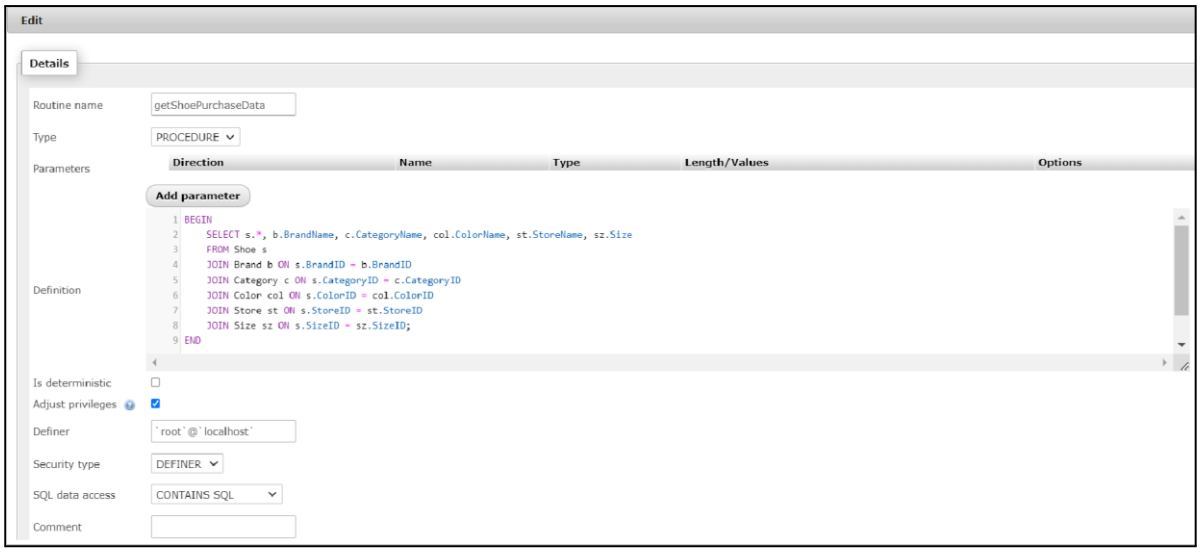


Figure 5.2 getShoePurchaseData

2. Triggers

   A trigger is a special type of stored procedure in a database that automatically executes in response to specific events on a particular table or view. Triggers are commonly used to enforce business rules, maintain data integrity, or automatically update related data when certain actions occur, such as inserting, updating, or deleting records.

c) After Delete Store

The after delete trigger on the `store` table is designed to automatically log information into the `storelogs` table whenever a store record is deleted. When a deletion occurs, this trigger is triggered by the database and inserts relevant details about the deleted store—such as its StoreID, StoreName, and the timestamp of the deletion—into the `storelogs` table. This process ensures that even though the store record is removed from the `store` table, a record of the deletion is preserved. The trigger uses the `OLD` keyword to reference the values of the deleted record, allowing it to capture and log the specific details. This automated logging is crucial for maintaining an audit trail, helping track changes, and providing a way to review or restore deleted information if necessary.

Figure 5.3 After Delete Store

5.4 Programming Technique

There are several programming techniques used in the system.

a. IF Statement

```
// Fetch store details
$storeSql = "SELECT * FROM Store WHERE StoreID = $storeId";
$storeResult = $connection->query($storeSql);
if (!$storeResult || $storeResult->num_rows == 0) {
    echo "Store not found.";
    exit;
}
$store = $storeResult->fetch_assoc();
```

b. IF… Else Statement

```
// Check if image file is an actual image or fake image
    $check = getimagesize($_FILES['homepage_image']['tmp_name']);
    if ($check !== false) {
        $uploadOk = 1;
    } else {
        $imageError = "File is not an image.";
        $uploadOk = 0;
    }
```

c. Nested If Statement

```
// Check if $uploadOk is set to 0 by an error
    if ($uploadOk == 0) {
        $imageError = "Sorry, your file was not uploaded.";
```

```
    } else {
        // If everything is ok, try to upload file
                                                                    if
(move_uploaded_file($_FILES['homepage_image']['tmp_name'],
$targetFile)) {
            // Store image path in session
            $_SESSION['homepage_image'] = $targetFile;
        } else {
                $imageError = "Sorry, there was an error uploading
your file.";
        }
    }
```

d. Switch Statement

```
switch ($row['UserType']) {
                case 'Admin':
                    header('location:homepage_admin.php');
                    exit();
                case 'ShoeEnthusiast':
                    header('location:index.php');
                    exit();
                case 'Store':
                    header('location:homepage_store.php');
                    exit();
                default:
                    $error[] = 'Invalid user type!';
                    break;
            }
```

e. Foreach Loop Statement

```
        <?php
        if (isset($error)) {
            foreach ($error as $err) {
                        echo '<span class="error-msg">' . $err .
'</span><br>';
            }
        }
        ?>
```

5.3 Error Handling

Error handling is a vital component of software development that involves devising strategies for managing and responding to system errors when they occur. The primary goal is to give consumers understandable and consistent alerts when system failures occur. Appropriate error management guarantees that the program can carry on with regular activities without interruption, even in the event of problems, and aids in providing users with insightful feedback. Developers may enhance the robustness and overall user experience of their program, making it more dependable and user-friendly, by foreseeing and correcting possible mistakes.

5.4.1 register.php

a. Password Should be at Least 8 Characters

Users must enforce a minimum password length of 8 characters, enhancing security by making passwords more resistant to brute force attacks.

```php
// Check password strength
if (empty($password)) {
    $error[] = 'Password is required!';
} elseif (strlen($password) < 8) {
    $error[] = 'Password should be at least 8 characters!';
} elseif (!$uppercase || !$lowercase || !$number || !$specialChars) {
    $error[] = 'Password should contain at least one uppercase letter, one lowercase letter,
    one number, and one special character!';
} elseif ($password !== $cpassword) {
    $error[] = 'Passwords do not match!';
}
```


Figure 5.4 Code Password

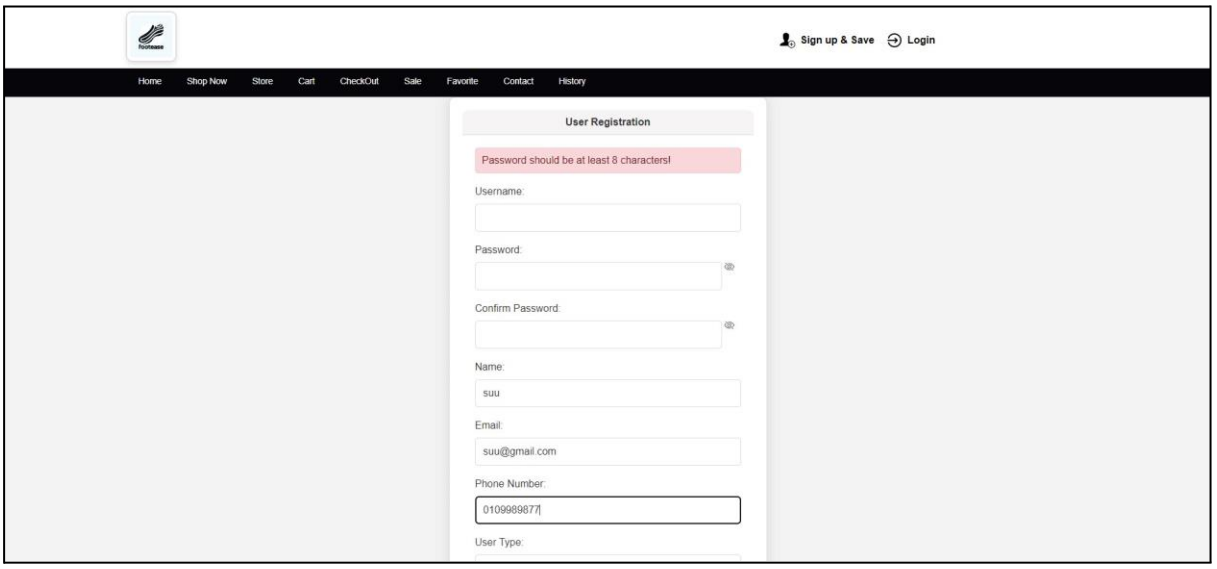


Figure 5.5 Password At Least 8 Characters

b. Password should contain at least one uppercase letter, one lowercase letter, one number, and one special character

Users must ensure that passwords contain at least one uppercase letter, one lowercase letter, one number, and one special character to enhance security by increasing the complexity and making them harder to crack.

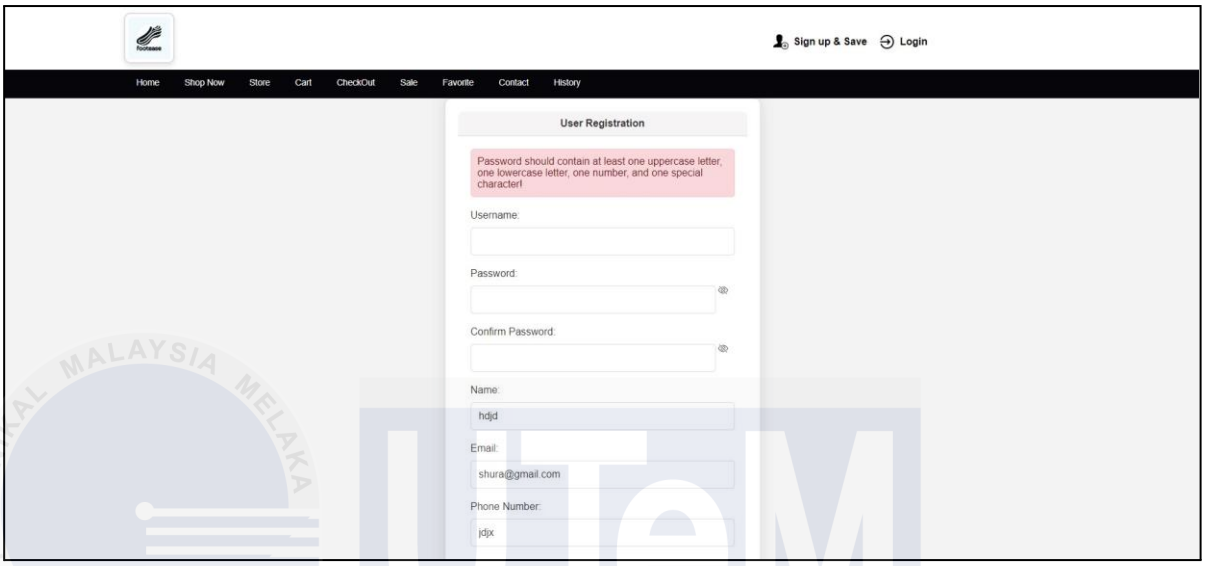


Figure 5.6 Password Format

c. Passwords do not Match

Users must ensure that both password entries match exactly to avoid errors and confirm that the correct password has been entered.

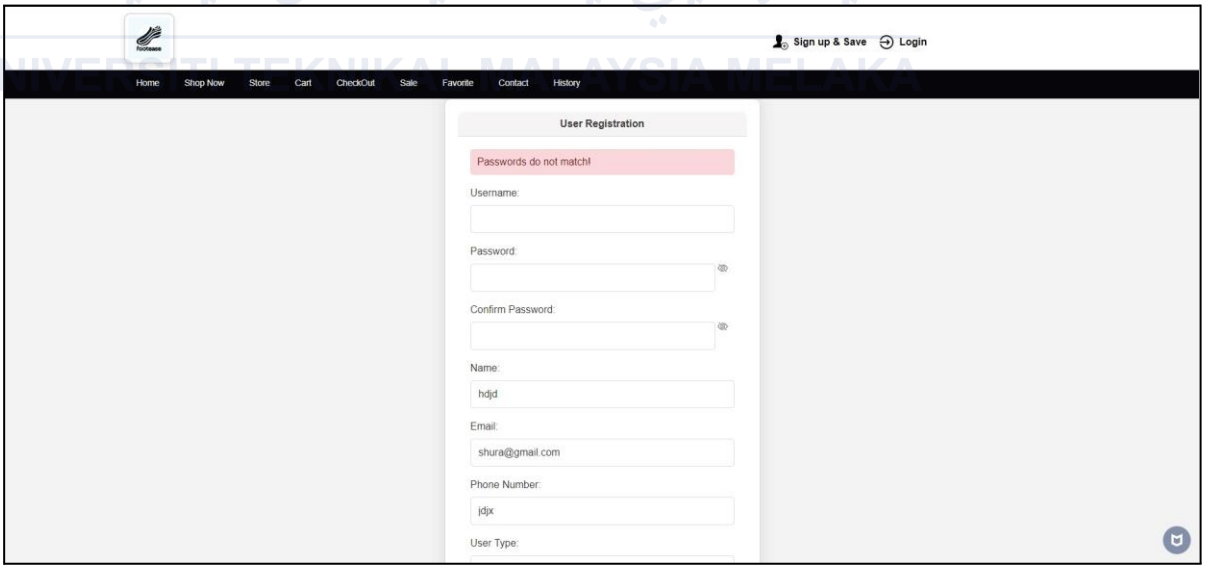


Figure 5.7 Password Do Not Match

d. User Already Exists

Users must ensure that the username is unique, as the chosen username already exists in the system, preventing duplication.

```php
if (mysqli_num_rows($result) > 0) {
    $error[] = 'User already exists!';
} else {
    // Call InsertNewUser stored procedure
    $insert = "CALL InsertNewUser('$username', '$hashed_pass', '$name', '$email', '$phonenumber', '$usertype')";
    if (mysqli_query($connection, $insert)) {
        // Call GetUserByEmail stored procedure to retrieve the user_id
        $selectUserID = "CALL GetUserByUsername('$username', @UserID)";
        mysqli_query($connection, $selectUserID);

        // Retrieve the user_id from the session variable
        $resultUserID = mysqli_query($connection, "SELECT @UserID as UserID");
        $rowUserID = mysqli_fetch_assoc($resultUserID);
        $UserID = $rowUserID['UserID'];

        header("Location:login.php");
        exit(); // Terminate the script after redirecting
    } else {
        $error[] = 'Error creating user: ' . mysqli_error($connection);
    }
}
```

Figure 5.8 Code User Exist

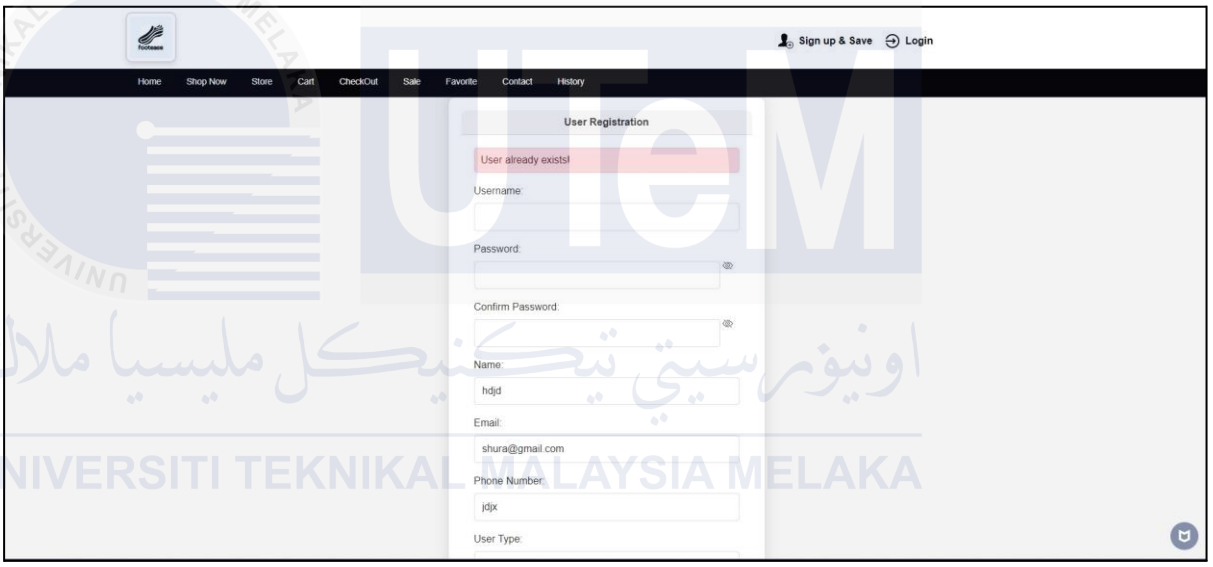


Figure 5.9 User Exist

5.3.2 login.php

a.User Does Not Exist. Please Register First

If a user enters invalid username, an error message “User does not exist. Please register first.” will be displayed. Users must ensure that they are registered before attempting to log in, as the entered username does not exist in the system. Please register first.

```
} else {
    // Display a message indicating that the user does not exist
    $error[] = 'User does not exist. Please register first.';
}
```

Figure 5.10 Code User Not Exist

Figure 5.11 User Not Exist

b. Incorrect Password

If a user enters invalid password, an error message “Invalid password” will be displayed. Users must ensure that the correct password is entered, as the provided password is incorrect. Please try again.

```
} else {
    // Display incorrect password message
    $error[] = 'Incorrect password!';
}
```

Figure 5.12 Code Incorrect Password

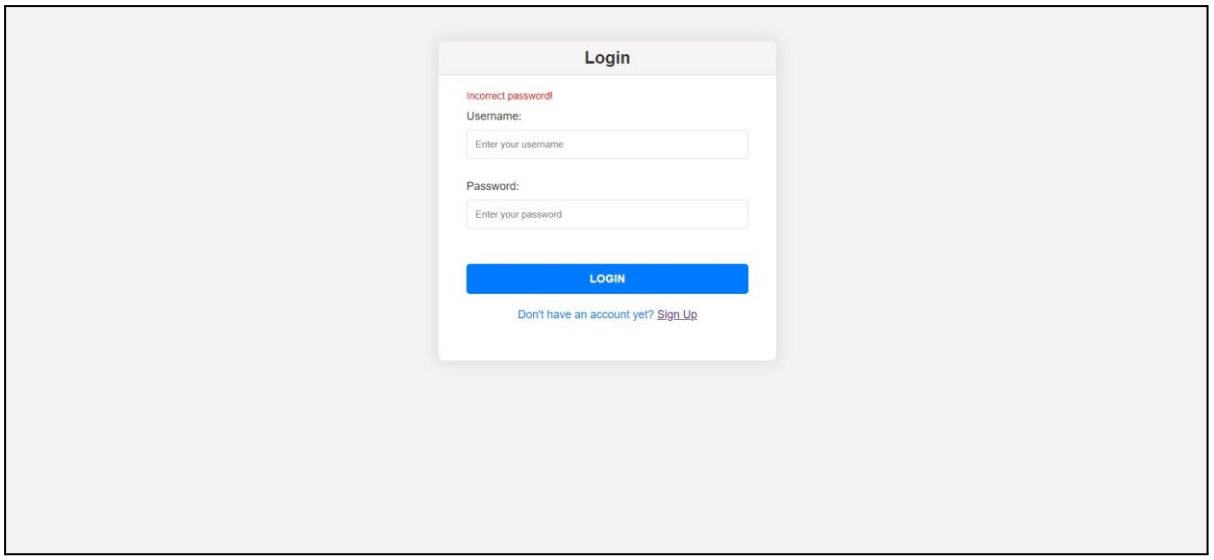


Figure 5.13 Incorrect Password

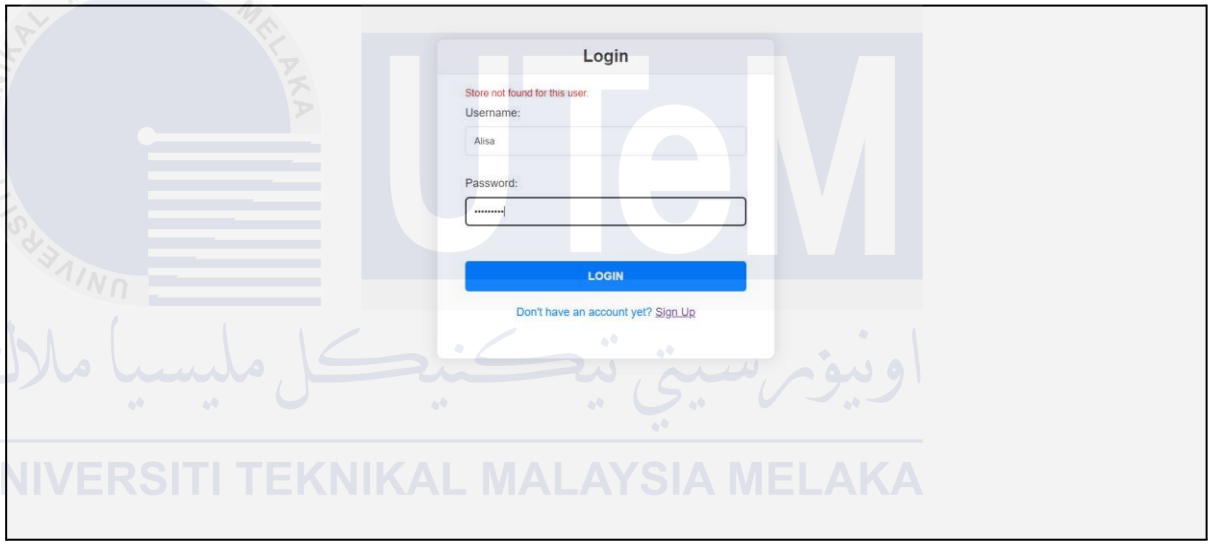


Figure 5.14 Store Does Not Exist

5.3.3 cart.php, checkout.php, favorite.php and history.php

b. Access Denied

Access is denied if the user is not logged in, ensuring that only authenticated users can view or interact with secure content. This protects sensitive information and resources from unauthorized access. Users must log in to proceed.

```
// Check if UserID is set in session
if (!isset($_SESSION['UserID'])) {
    echo '<!DOCTYPE html>
    <html>
    <head>
        <title>Access Denied</title>
        <style>
            .error-container {
                font-family: Arial, sans-serif;
                display: grid;
                place-items: start center;
                height: 100vh;
                width: 100%;
                margin: 0;
                padding: 0;
                background-color: #f8f9fa;
            }
            .message-box {
                text-align: center;
                background-color: #fff;
                padding: 20px;
                box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
                border-radius: 8px;
                max-width: 400px;
                margin-top: 20px;
            }
```

```
        .message-box h1 {
            color: #d9534f;
            font-size: 24px;
            margin-bottom: 10px;
        }
        .message-box p {
            color: #555;
            font-size: 16px;
            margin-bottom: 20px;
        }
        .btn {
            padding: 10px 20px;
            background-color: #007bff;
            color: white;
            text-decoration: none;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            transition: background-color 0.3s;
            font-size: 16px;
        }
        .btn:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div class="error-container">
        <div class="message-box">
            <h1>Access Denied</h1>
            <p>You must log in first.</p>
            <a href="login.php" class="btn">Log In</a>
        </div>
    </div>
</body>
</html>';
exit();
}
```

Figure 5.15 Code Access Denied

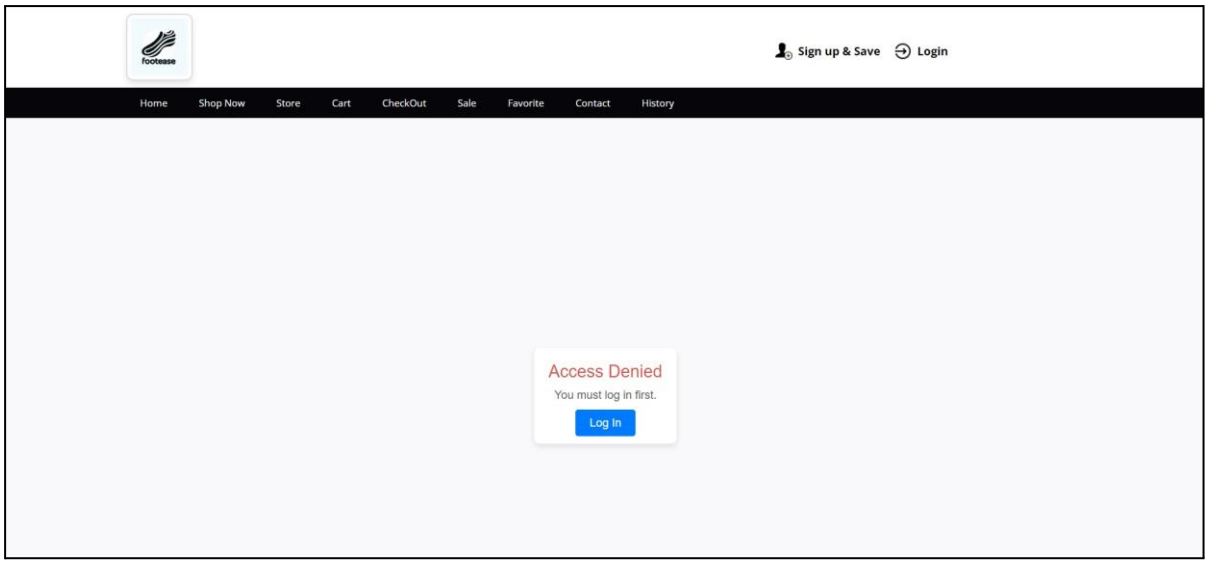


Figure 5.16 Access Denied

5.3.4 single.php

a. Quantity exceeds available inventory

Users must ensure that the requested quantity does not exceed the available inventory. If the quantity exceeds the stock on hand, the system will prevent the order to maintain accurate inventory levels.

```
quantityInput.addEventListener('change', function() {
    const color = selectedColorInput.value.trim();
    const size = selectedSizeInput.value.trim();
    const inventory = <?php echo json_encode($inventory); ?>;
    const maxQuantity = inventory[color][size] || 0;
    let quantity = parseInt(quantityInput.value);
    if (quantity > maxQuantity) {
        quantity = maxQuantity;
        quantityInput.value = quantity;
        alert('Selected quantity exceeds available inventory.');
    }
    quantityHiddenInput.value = quantity;
});
```

Figure 5.17 Code Quantity Exceeds Available Inventory

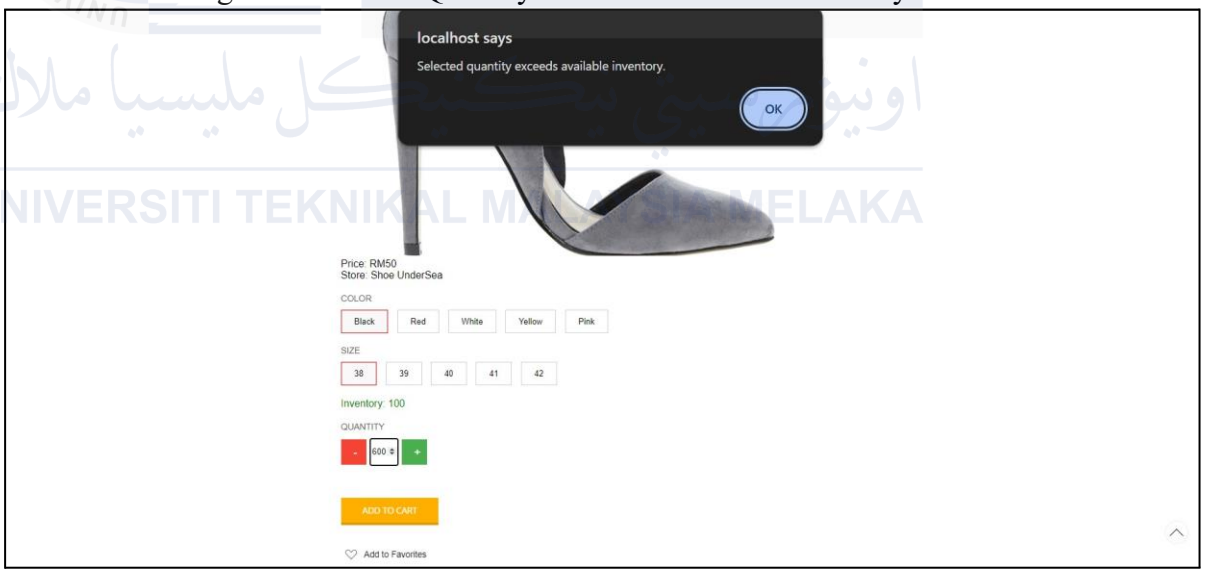


Figure 5.18 Quantity Exceeds Available Inventory

b. Color and size are out of stock

Users must select an available option, as the chosen color and size are currently out of stock. This ensures that only in-stock items can be added to the cart for purchase.

```
// Function to update inventory display
function updateInventory() {
    const color = selectedColorInput.value.trim();
    const size = selectedSizeInput.value.trim();
    if (color && size) {
        const inventory = <?php echo json_encode($inventory); ?>;
        const quantity = inventory[color][size] || 0;
        inventoryQuantity.innerText = quantity;
        inventoryMessage.style.display = 'block';
        if (quantity === 0) {
            disableButtons();
            alert('Selected color and size are out of stock.');
        } else {
            enableButtons();
        }
    }
}
```

Figure 5.19 Code Out Of Stock

Figure 5.20 Out Of Stock

5.3.5 credit_card_payment.php

```
<script>
document.addEventListener("DOMContentLoaded", function() {
    const stripe = Stripe('pk_test_51PPmCJRwk5noFcUyDbgjyi6VmM4EJSY9TwKnjuZF2Sl2u0ASAendQ71YDop8K8CFxTwjKVwRXV8jwXBv1gCu19FI00Y6hx1oEg');
    const elements = stripe.elements();

    const style = {
        base: {
            color: "#32325d",
            fontFamily: '"Helvetica Neue", Helvetica, sans-serif',
            fontSmoothing: "antialiased",
            fontSize: "16px",
            "::placeholder": {
                color: "#aab7c4"
            }
        },
        invalid: {
            color: "#fa755a",
            iconColor: "#fa755a"
        }
    };

    const cardNumber = elements.create('cardNumber', {style: style});
    const cardExpiry = elements.create('cardExpiry', {style: style});
    const cardCvc = elements.create('cardCvc', {style: style});
```

```
    cardNumber.mount('#card-number');
    cardExpiry.mount('#card-expiry');
    cardCvc.mount('#card-cvc');

    const form = document.getElementById('payment-form');

    form.addEventListener('submit', async function(event) {
        event.preventDefault();

        const {token, error} = await stripe.createToken(cardNumber);

        if (error) {
            console.log('Error creating token:', error);
            document.getElementById('card-errors').textContent = error.message;
        } else {
            stripeTokenHandler(token);
        }
    });
```

```
    function stripeTokenHandler(token) {
        const hiddenInput = document.createElement('input');
        hiddenInput.setAttribute('type', 'hidden');
        hiddenInput.setAttribute('name', 'stripeToken');
        hiddenInput.setAttribute('value', token.id);
        form.appendChild(hiddenInput);

        // Disable the submit button to prevent multiple submissions
        document.getElementById('card-button').disabled = true;

        form.submit();
    }
});
</script>
```

Figure 5.21 Code Payment

Card

Users

a. Expiration Year is in the Past

must provide a valid card expiration year, as the entered year is in the past. A current or future expiration date is required to process the payment.

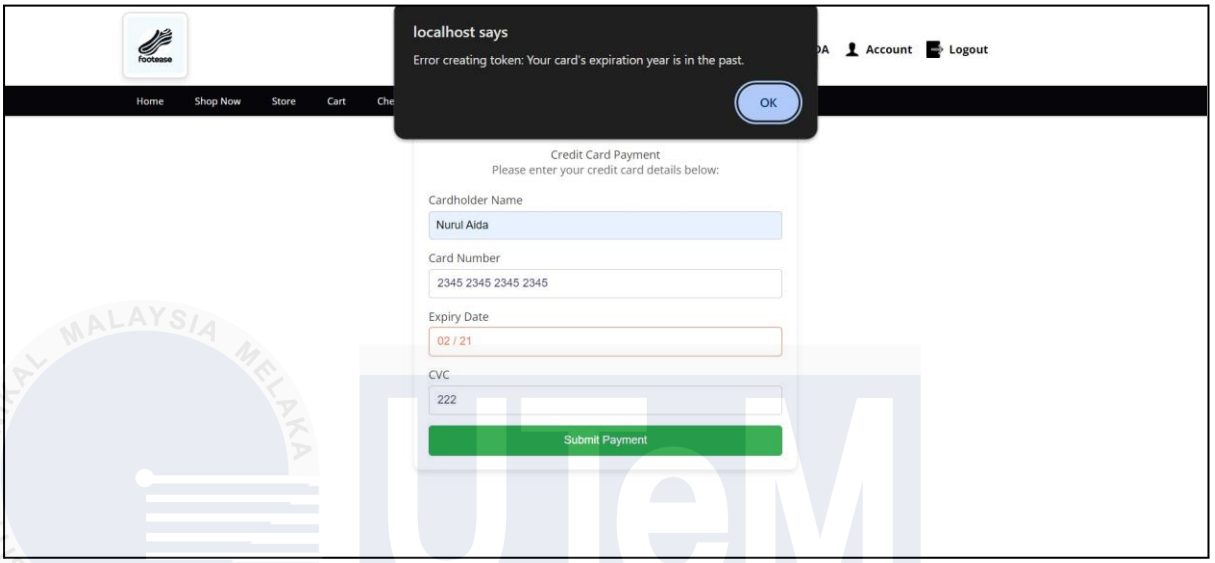


Figure 5.23 Card Expired

b. Card Number Invalid

Users must provide a valid card expiration year, as the entered year is in the past. A current or future expiration date is required to process the payment.

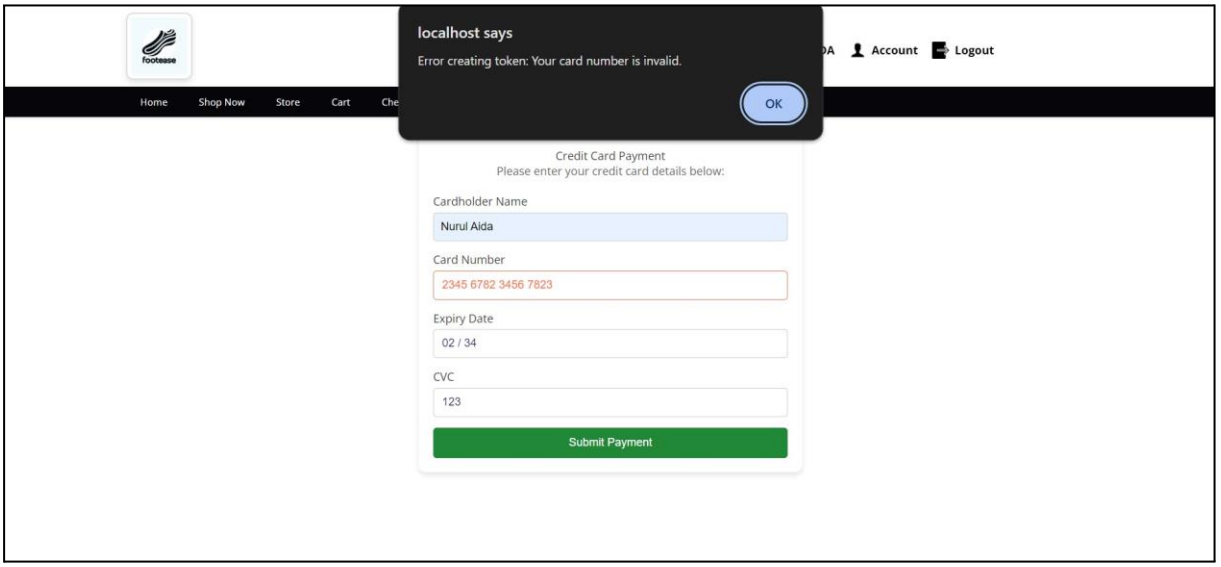


Figure 5.24 Card Number Invalid

Card

Users

c. Security Code is Incomplete must enter the full card security code, as the provided code is incomplete. A complete and accurate security code is required to verify the card and process the transaction.

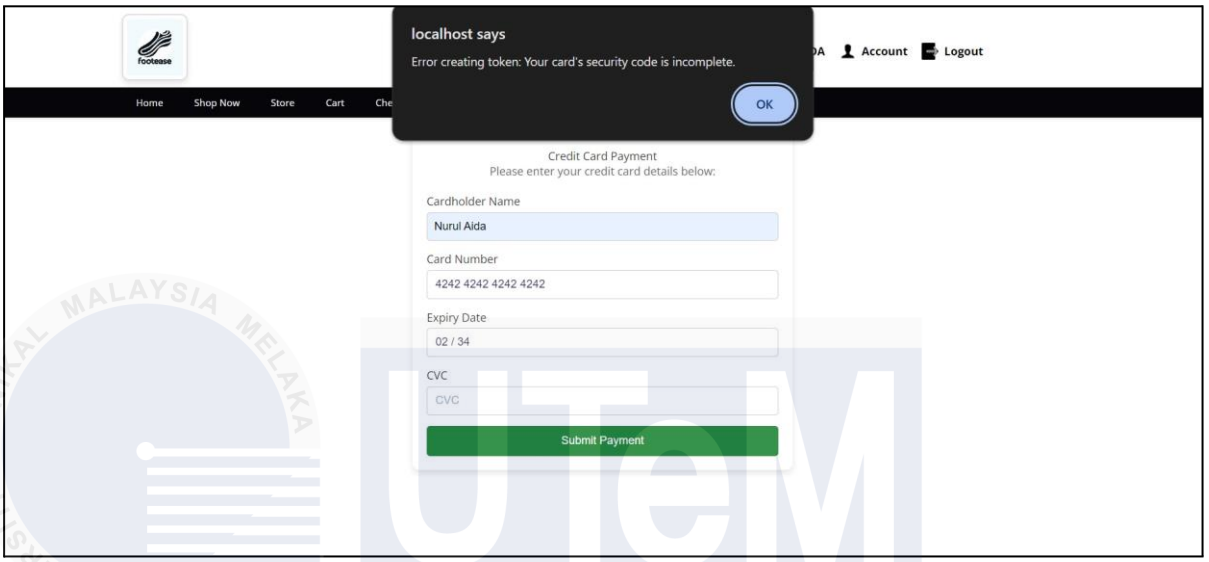
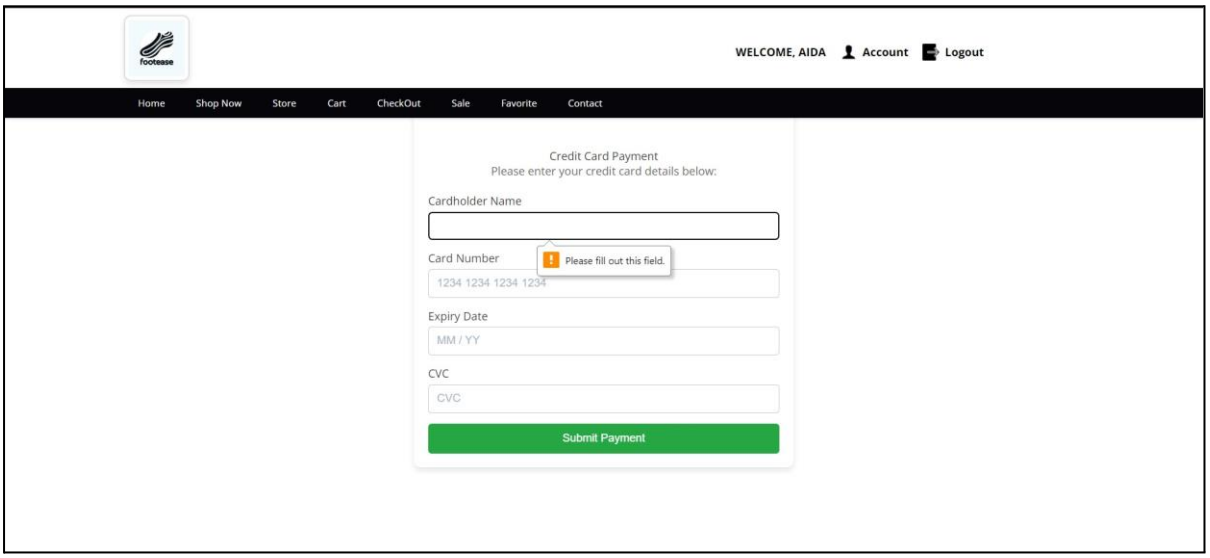


Figure 5.25 Card Security not Complete

d. Required Fields are Filled in

Users must ensure that all required fields are filled in, as one or more fields are currently incomplete. Completing all fields is necessary to proceed.



Card

Users

Figure 5.26 Empty Required Fields

e. Number is Incomplete must enter the full card number, as the provided card number is

incomplete. A complete card number is required to process the payment.

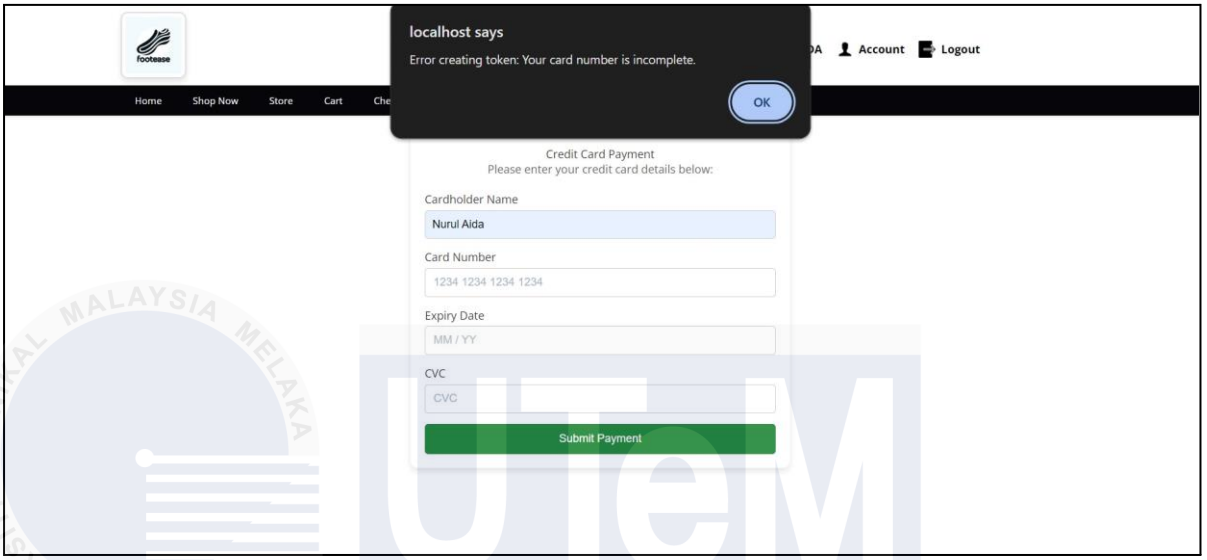


Figure 5.27 Card Number Incomplete

f. Card Expiration date is Incomplete

Users must enter both the month and year for the card expiration date, as the provided expiration date is incomplete. A complete expiration date is necessary to process the payment.

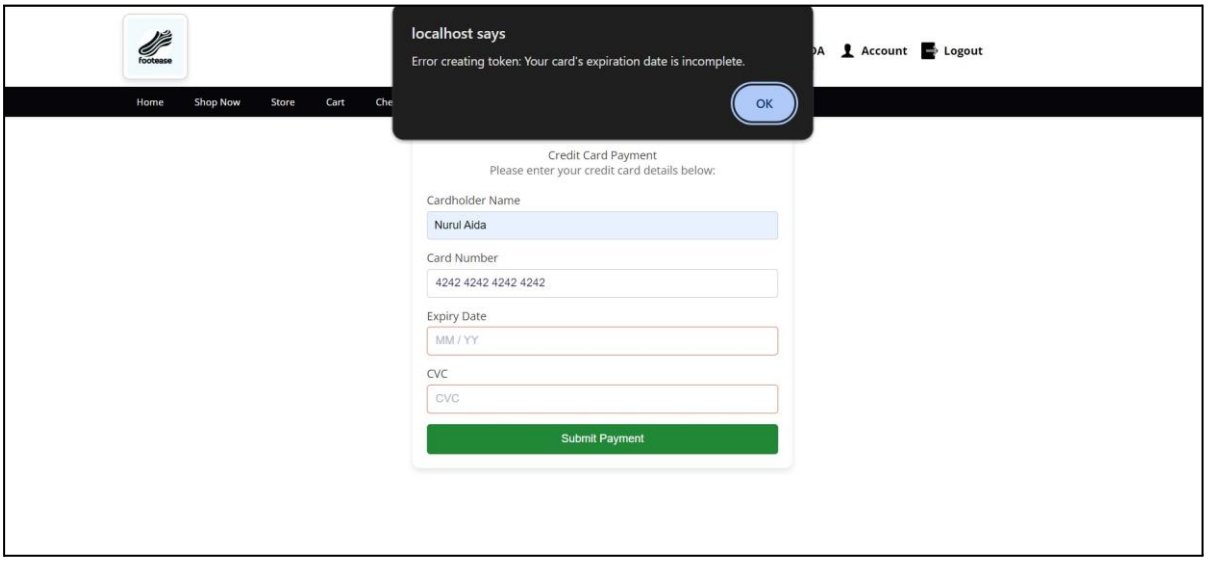


Figure 5.28 Card Expiration Incomplete

attempts

5.3.6 delete_store.php

a. Delete Store

When the user to delete a store, a popup message should appear asking, "Are you sure you want to delete this store?" This confirmation prompt helps prevent accidental deletions by requiring the user to confirm their action before proceeding.

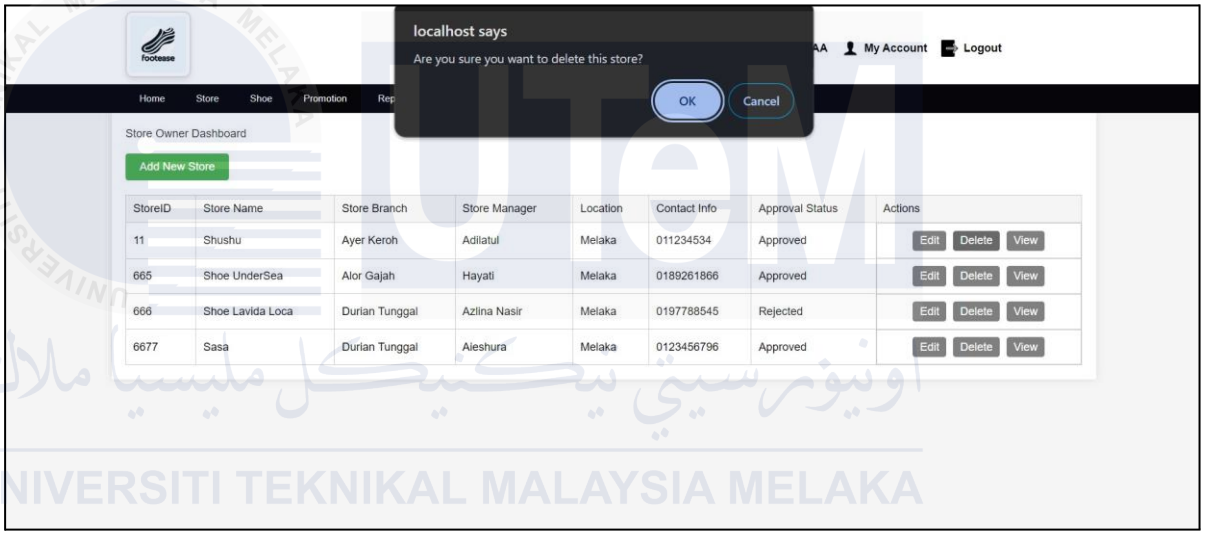


Figure 5.29 Delete Store

5.3.7 delete_shoe.php

a. Delete Shoe

When the user to delete a shoe, a popup message should appear asking, "Are you sure you want to delete this shoe?" This confirmation prompt helps prevent accidental deletions by requiring the user to confirm their action before proceeding.

attempts

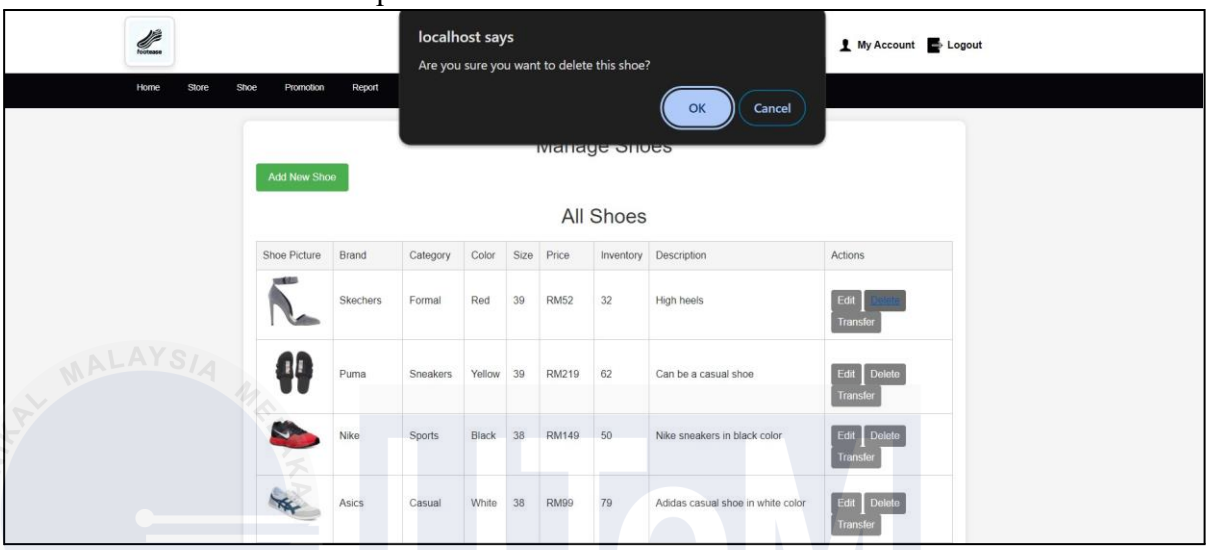


Figure 5.30 Delete Shoe

5.3.8 delete_promotion.php

a. Delete promotion

When the user to delete a promotion, a popup message should appear asking, "Are you sure you want to delete this promotion?" This confirmation prompt helps prevent accidental deletions by requiring the user to confirm their action before proceeding.

attempts

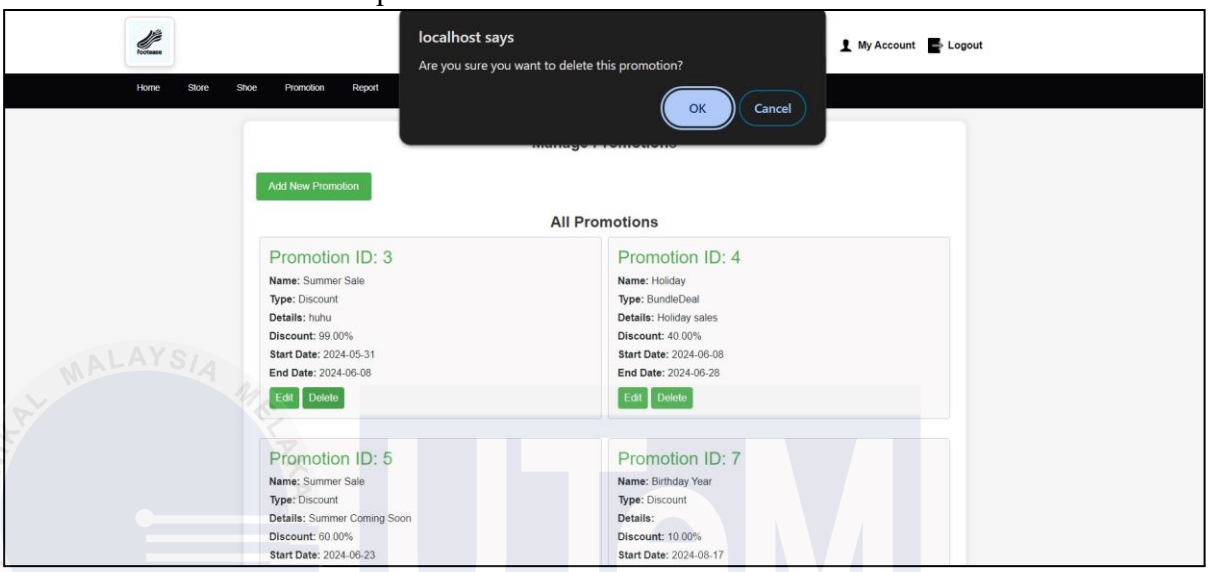


Figure 5.31 Delete Promotion

5.4 Conclusion

In summary, this project's implementation phase successfully converts business requirements into useful application modules, with an emphasis on the admin, store owner, and shoe enthusiast. It has created a reliable and easy-to-use system by utilizing a variety of programming methods, including as switch statements, foreach loops, nested IF statements, IF...ELSE expressions, and IF statements. The smooth operation and communication between various application components is ensured by the integration of these modules. Comprehensive error handling procedures have also been included to improve the system's overall dependability and user experience by giving users clear and consistent signals. This meticulous implementation strategy highlights the project's dedication to providing a well-thought-out and effective

CHAPTER 6: TESTING

6.1 Introduction

This chapter provides an overview of the testing phase of the project. During this phase, various testing activities will be conducted to ensure the system meets the specified requirements and performs as expected. The testing strategy adopted in this project includes a combination of manual and automated testing, focusing on functional, security, and performance aspects. The goal is to identify and resolve any defects or issues before the system is deployed for production use.

6.2 Test Plan

A test plan is a document that outlines how software will be tested. It defines the goals of testing, what parts of the software will be tested, and what won't. It includes details on the tools and team members needed, the schedule for testing activities, and the specific scenarios that will be tested. The plan also sets criteria to determine if the software meets the required standards. Essentially, a test plan helps ensure that the software is thoroughly checked for issues before it is released.

6.2.1 Test Organization

The testing team consists of key personnel who are responsible for executing the test plan. This includes a Test Manager, who oversees the testing process; Test Analysts, who design and document the test cases; and Test Engineers, who execute the test cases and document the results. Additionally, there is a Quality Assurance (QA) Lead, who ensures that the testing processes adhere to the project's quality standards.

Table 6.1 List of the user and their responsibility

Tester ID	Title/Post	Responsibility
Tester 1	System Developer	Designs, builds, and maintains software systems. Write code, test and debug software, and collaborate with other developers and stakeholders to create efficient and effective solutions.

Tester 2	Owner System	Oversees the operation and management of a system. Responsible for ensuring the system meets business needs, managing updates and maintenance, and addressing any issues that arise.
Tester 3	Admin	Manages and maintains the administrative aspects of a system. This includes user management, system configuration, troubleshooting issues, and ensuring system security and performance.
Tester 4	Store Owner	Operates and manages a retail store. Responsibilities include overseeing daily operations, managing inventory and ensuring the store meets the sales.
Tester 5	Shoe Enthusiast	A person who has a deep interest and passion for shoes. This may involve collecting, reviewing, or discussing various types of footwear, keeping up with trends, sharing knowledge with others who have similar interests, and buying the shoes they prefer.

6.2.2 Test Environment

Testing will be conducted in a controlled environment that stimulates the production setup. The test environment is located in the company's testing lab, equipped with the necessary hardware, such as servers, workstations, and network equipment. The firmware configurations will match the production environment to ensure accurate test results. Prior to testing, the team will undergo training to familiarize themselves with the testing tools and procedures. All necessary software and hardware will be prepared and configured according to the test plan.

a. Environment Setup Specification

Table 6.2 Environment setup specification

Environment Specification	**Description**

Device Name	HP Pavilion x360 Convertible 14
Processor	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
Random Access Memory (RAM)	8GB
Database	phpMyAdmin
Server Scripting	MySQL

b. Software Application

Table 6.3 Software Application

Environment Specification	**Description**
Operating system	**Windows 11 Home Single Language**: The latest Microsoft operating system with a new design, centered Start menu, improved multitasking, Snap Layouts, and built-in Teams for easier use and communication.
Text Editor	**Microsoft Visual Studio Code**: A free, lightweight code editor that supports various programming languages with features like syntax highlighting, debugging, and version control.
Web Server	**XAMPP**: A free tool for setting up a local web server, including Apache, MySQL, PHP, and Perl, for developing and testing web applications.

Database Management System (DBMS)	**MySQL**: An open-source database system that stores data in tables and uses SQL for managing and querying data, commonly used in web applications.
Web Browser	**Google Chrome**: A fast, secure web browser by Google, known for its speed, simplicity, and extensive extensions, with features like synchronization and security tools.
	Microsoft Edge: A web browser by Microsoft, built on Chromium, offering integration with Microsoft services, privacy tools, and a reading mode.
Documentation Tools	**Draw.io**: A free online tool for creating and sharing diagrams and flowcharts directly from your browser. **Google Docs**: A cloud-based word processor for creating, editing, and collaborating on documents in real-time, with features for sharing and commenting

6.2.3 Test Schedule

The testing phase will consist of multiple test cycles. Each cycle is planned to last two weeks, with a total of three cycles. The first cycle will focus on functional testing, the second on performance and stress testing, and the final cycle on security and regression testing. The duration of each cycle is based on the complexity of the tests and the availability of resources.

Table 6.4 Test Schedule for Administrator

Module/ Component	Test Type	Duration (Day)	Start Date	End Date	Tester
System Login	Security Testing	1	15/8/2024	15/8/2024	1,2,3
Registration	Integration Testing	1	15/8/2024	15/8/2024	1,2,3
User Management	Functional Testing	1	18/8/2024	18/8/2024	1,2,3
Store Management	Functional Testing	1	16/8/2024	16/8/2024	1,2,3
Shoe Management	Functional Testing	1	25/8/2024	25/8/2024	1,2,3
Report	Performance Testing	1	28/8/2024	28/8/2024	1,2,3

Table 6.5 Test Schedule for Store Owner

Module/ Component	Test Type	Duration (Day)	Start Date	End Date	Tester
System Login	Security Testing	1	15/8/2024	15/8/2024	1,2,4
Registration	Integration Testing	1	15/8/2024	15/8/2024	1,2,4
Store Management	Functional Testing	3	16/8/2024	18/8/2024	1,2,4
Shoe Management	Usability Testing	1	22/8/2024	22/8/2024	1,2,4
Promotion Management	Functional Testing	1	23/8/2024	23/8/2024	1,2,4

Report	Performance Testing	1	28/8/2024	28/8/2024	1,2,4

Table 6.6 Test Schedule for Shoe Enthusiast

Module/ Component	Test Type	Duration (Day)	Start Date	End Date	Tester
System Login	Security Testing	1	15/8/2024	15/8/2024	1,2,5
Registration	Integration Testing	1	15/8/2024	15/8/2024	1,2,5
Search and Discovery	Usability Testing	1	16/8/2024	16/8/2024	1,2,5
Shopping	Functional Testing	1	17/8/2024	17/8/2024	1,2,5
Store Exploration	Usability Testing	1	19/8/2024	19/8/2024	1,2,5
Shoe Exploration	Usability Testing	1	23/8/2024	23/8/2024	1,2,5
Promotion Section	Functional Testing	1	23/8/2024	23/8/2024	1,2,5
User Preference (favorite)	Functional Testing	1	25/8/2024	25/8/2024	1,2,5
User Account	Functional Testing	1	25/8/2024	25/8/2024	1,2,5
Payment	Functional Testing	1	26/8/2024	26/8/2024	1,2,5
History	Integration Testing	1	28/8/2024	28/8/2024	1,2,5

6.3 Test Strategy

A test strategy is a blueprint which describes how testing will be conducted for a given project. It encompasses the expectations of testing, what is to be tested and what is not, and the classification of types of tests to be done such as, checking if a feature is properly functional and if the system is fast among others. In this section, the approach to testing will be outlined, who and what will be involved in testing and how test cases will be developed. The strategy also defines the plan on how defects will be controlled and monitored, how the measures of progress will be assessed, and the plan on the testing timeline for the given software. In other words, it assists to make certain that testing is systematic and comprehensive so that a quality product is achieved. The testing strategy which has been used in this project is a blend of the bottom up and top down strategies.

a. Top-down Test Strategy

Top-down testing starts by testing the system at a high level and progressively breaks it down into smaller parts. The idea is to test the overall functionality and integration of the system as a whole, before focusing on lower-level components. The top-down approach will then be applied to test the integration and overall system functionality.

6.3.1 Testing Technique

a. Black-box Test Technique

Black-box testing focuses on evaluating the functionality of a software application without any knowledge of its internal code or structure. The tester only interacts with the system from the user's perspective. Types of tests that include in this black-box testing are functional testing, system testing, integration testing, and acceptance testing.

Table 6.7 Black Box

Module	Description

System Login	Verifying that users can log in with correct credentials and receive appropriate error messages with invalid credentials.
Registration	Ensuring that user registration works correctly, capturing all required details, and validating inputs like email format or password strength.
Store Exploration	Testing whether users can view stores and shoes correctly, apply filters, and see accurate information.
Shoe Exploration	Testing whether users can view stores and shoes correctly, apply filters, and see accurate information.
Shopping	Checking if users can browse and purchase shoes, ensuring that the checkout process works as intended.

Promotions Section	Ensuring that users can view and access current promotions and that the terms and details are accurately displayed.
Report	Verifying that the reports generated reflect accurate sales, orders, and shoe data.
User Preference (favorite)	Tests whether users can mark items as favorites and manage their preferences, focusing on how these features work from the user's perspective.
User Account	Validates account management features like updating information and handling password changes, ensuring these processes work as expected.
Payment	Tests the payment process including transaction handling and confirmation, based on user interactions and requirements.

History	Ensures that users can view their transaction and browsing history
	correctly, focusing on the functionality without delving into the code.

i. Example Black-box Testing in Credit Card Payment

Stripe and Credit Card Payments

Stripe is one of the well-known payment gateways that are used for making transactions through credit cards online. that offers APIs that can be used by developers to add payment functionality in their apps. There is a test mode available in the Stripe which allows one to test the payment gateway without the actual payments being processed. One of them is the test card number 4242 4242 4242 4242, which always leads to the payment approval. Any other card numbers will usually replicate a charge back or invoke other test cases, for example, lack of adequate funds, charge back.

Black-Box Testing and Stripe:

Black-box testing is a software testing process in which a tester works with the idea that she or he does not know about the internal implementation of the system. The aim is to check that the system responds to a certain or specific functional requirements in a certain way.

Applying Black-Box Testing to Stripe's Payment API:

In the context of testing the payment system using Stripe's API, here's how black-box testing could be applied:

1. **Test Case 1: Successful Payment:**

- Input: Use the card number 4242 4242 4242 4242 with valid details (e.g., expiration date, CVC).
- Expected Output: Payment should succeed, and a successful transaction response should be returned from Stripe's API.

2. **Test Case 2: Failed Payment (Invalid Card):**
   - Input: Use any card number other than 4242 4242 4242 4242.
   - Expected Output: The payment should fail, and Stripe's API should return a failure response, typically with an error code indicating an invalid card.
3. **Test Case 3: Expired Card:**
   - Input: Use an expired card with valid details.
   - Expected Output: Payment should fail, and Stripe should return an error indicating that the card is expired.
4. **Test Case 4: Invalid CVC:**
   - Input: Use a valid card number but with an invalid CVC.
   - Expected Output: Payment should fail with an error indicating an incorrect CVC.

How Black-Box Testing Helps:

In black-box testing, you are interested in how the system behaves under certain input (for example you input card details in order to check some payment features), and what result it returns (for instance whether the payment was processed or not). In the first case you do not care how Stripe internally handles the payment, it is only important to check whether the API respects the documented specifications.

Advantages in the Stripe Case:Advantages in the Stripe Case

- Real-world scenarios: Black box testing includes valid payment, invalid payment due to wrong details and many other real life simulations.

- Test environment: Stripe's test environment enables such conditions to be practiced without using actual credit card information that is appropriate for black-box testing.
- Validation of payment process: They guarantee that it processes success and failure situations effectively and make the system more sound.

Finally, employing the test card available on Stripe to emulate payment for trial equals conforming to black-box testing framework. This makes it possible for the testers to confirm the payment processing feature developed without having to come across the code or even the details of how Stripe works.

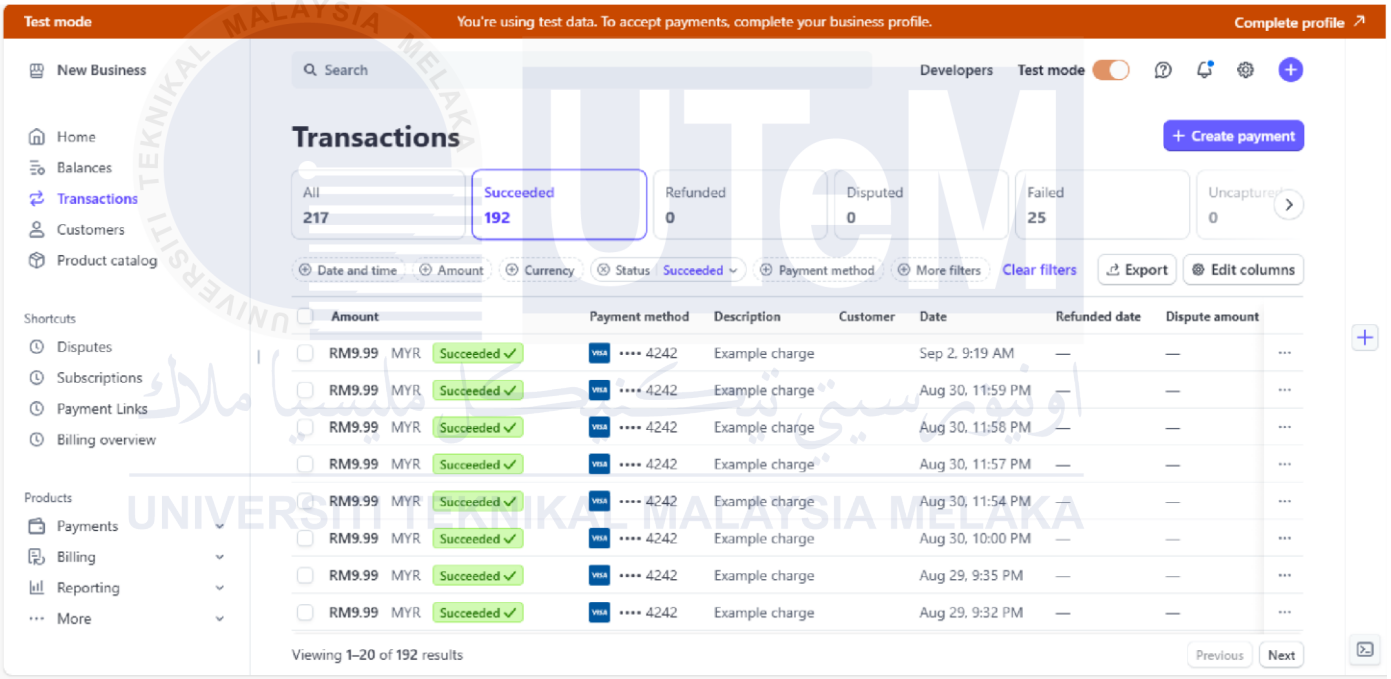


Figure 6.1 Succeeded Payment in Stripe Dashboard

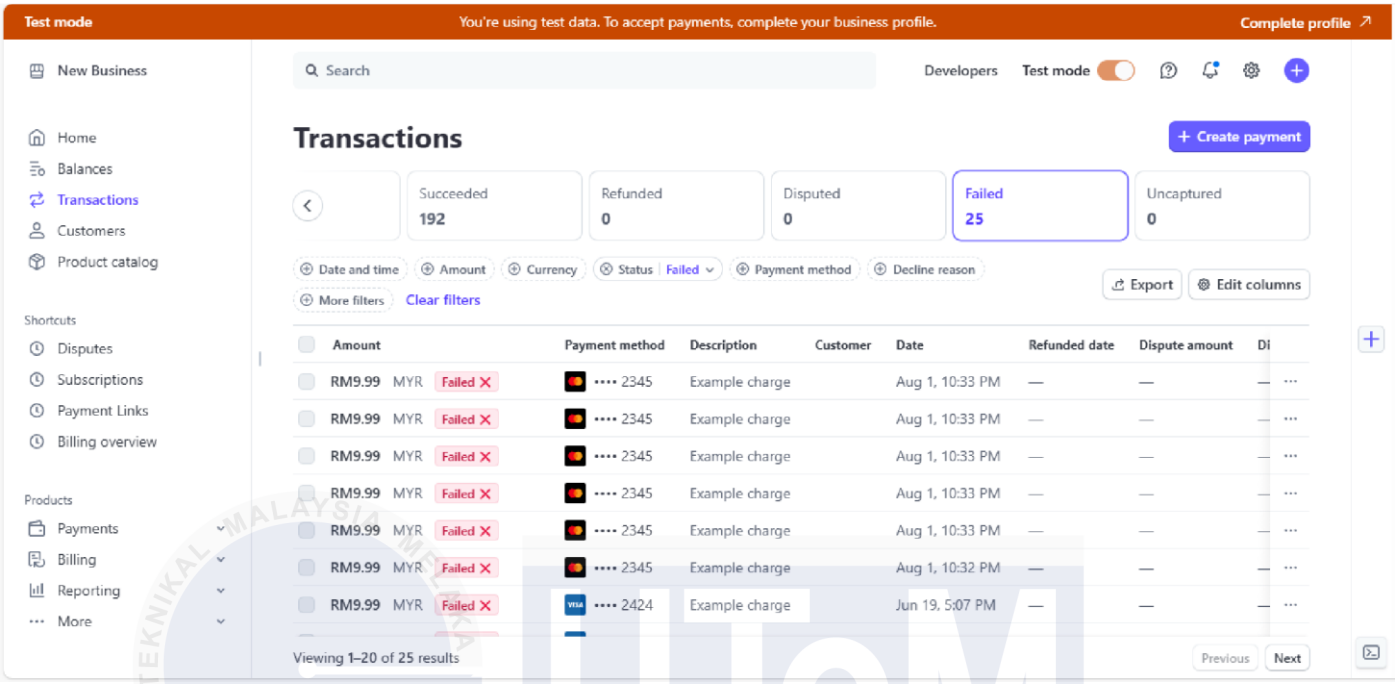


Figure 6.1 Failed Payment in Stripe Dashboard

b. White-box Testing

White-box testing, also known as clear-box or structural testing, involves examining the internal workings of a system. Testers have access to the code and design, allowing them to test the internal logic and structure. Types of tests for this white-box testing include unit testing, code coverage testing, and integration testing at the code level.

Table 6.8 White Box

Module	**Description**
User Management	Examining the internal code logic that handles user account management, such as account creation, deletion, and moderation processes.

Store Management	Testing the internal workings of how stores are managed within the system, including the code that updates, deletes, or submits store details.
Shoe Management	Analyzing the logic behind how shoes are added, updated, or transferred within the system, ensuring that inventory updates are handled correctly.
Report	Diving into the internal algorithms that generate reports based on shoe, sales, and order data, ensuring accurate data aggregation and processing.
Payment	To inspect the internal logic of payment processing and security.

6.3.2 Classes of Tests

1. **Functional Testing:**

Purpose: Ensures that each function of the software application operates in conformance with the requirement specification.

Focus: Validates the software's functionalities by testing them against the predefined requirements. It includes testing user interactions, data inputs, processing, and outputs.

Example: Checking if the login functionality correctly authenticates users when valid username and password are entered.

2. **Integration Testing:**

Purpose: Tests the interfaces between different modules or components of a software application to ensure they work together as expected.

Focus: Ensures that combined parts of the system function together without issues. It often follows unit testing and precedes system testing.

Example: Verifying that the registration process successfully creates a user record in the database.

3. **Security Testing:**

Purpose: Identifies vulnerabilities in the software, ensuring that the system is secure from unauthorized access and data breaches.

Focus: Focuses on protecting data and maintaining functionality as intended. It includes testing for SQL injection, cross-site scripting (XSS), data encryption, and user authentication.

Example: Testing whether a user's password is encrypted and stored securely in the database.

4. **Performance Testing:**

Purpose: Evaluates the system's performance under expected and stress load conditions to ensure it meets performance criteria.

Focus: Measures the system's response time, throughput, reliability, and scalability under different loads.

Example: Assessing how the report each store perform.

5. **Usability Testing:**

Purpose: Ensures that the software is user-friendly and that users can easily navigate and use the application without confusion.

Focus: Focuses on the end-user experience, checking the ease of use, intuitiveness, and overall user satisfaction.

Example: Observing how easily a new user can complete the registration process and identify any points of confusion.

6. **Stress Testing:**

Purpose: Evaluates how the software performs under extreme conditions to ensure it can handle high loads and stress without crashing or slowing down significantly.

Focus: Focuses on the software's stability and performance under heavy or peak loads. It includes testing how the system handles high volumes of data, numerous simultaneous users, and prolonged usage.

Example: Testing how a website performs when it receives a sudden surge of traffic from hundreds of users simultaneously.

6.4 Test Design

Test Design is about planning how to check if a software works correctly. It involves deciding what to test, creating test cases with steps and expected results, and preparing the necessary data. The aim is to make sure the software functions as expected, performs well, and is secure.

6.4.1 Test Description

Each module of the system will have a set of test cases identified and documented. These test cases will detail the specific conditions to be tested, the inputs to be used, and the expected results. The test cases will be designed to cover all critical functionality and scenarios, including edge cases. The test design document will also include a traceability matrix, mapping each test case to the corresponding requirement or specification.

Table 6.9 Test Description Login Module

Test Case ID	TC_01
Test Case Name	Login
Actor	Admin, Store owner and Shoe enthusiast
Description	This test case allows the tester or user to demonstrate the registration user module by following a few test cases and steps. To start the testing, tester or user needs to: 1. Click “Login” button at homepage Then proceed with the following test case.

Test Case	**Description**	**Step**	**Expected Result**

TC_01-1	Test login with valid credentials (correct username and password)	1. Insert Email: <username> 2. Insert Password: <password> 3. Click “Login” button	Users with the correct username and password should be successfully logged into the system.
TC_01-2	Test login with invalid credentials (incorrect username).	1. Insert Email: <username> 2. Insert Password: <password> 3. Click “Login” button	Users with invalid username should be denied access and shown an appropriate error message “Invalid username”.
TC_01-3	Test login with invalid credentials (incorrect password).	1. Insert Email: <username> 2. Insert Password: <password> 3. Click “Login” button	Users with invalid passwords should be denied access and shown an appropriate error message “Invalid password”.
TC_01-4	Test login with empty fields.	1. Click “Login” button	The system should prompt users to fill in both username and
			password fields if they are empty.

Table 6.10 Test Description Registration Module

Test Case ID	TC_02
Test Case Name	Registration
Actor	Store owner, Shoe enthusiast
Description	This test case allows the tester or user to demonstrate the registration user module by following a few test cases and steps. To start the testing, tester or user needs to: 1. Click “Sign Up & Save” at the navigation menu bar 2. Choose to register as store or shoe enthusiast 3. Click “Register” button Then proceed with the following test case.

Test Case	**Description**	**Step**	**Expected Result**

TC_02-1	Test registration with all required fields filled correctly.	1. Fill in all required fields. 2. Click “Register” button	Users should be able to register successfully when all required information is provided correctly and directly to the login page.
TC_02-2	Test the system’s response to duplicate registrations by entering an existing username.	1. Fill in all required fields 2.Fill in the username that has been registered 3.Click the “Register” button	Duplicate registrations should be prevented with proper validation. “Users already exist!” Prompt will appear
TC_02-3	Test login with empty fields.	1. Click the “Register” button	The system should prompt users to fill in all required fields. “Please fill in this field” required field message will appear

TC_02-4	Test the email without '@'	1. Fill in all required fields 2. Insert the email field without the symbol '@' 3. Click "Register" button	"Please include an '@' in the email address" required field message will appear
TC_02-5	Test with wrong format email	1. Fill in all required fields 2. Insert the email field until @ only 3. Click the "Register" button	"Please enter a part following @. The email incomplete" warning will be appear
TC_02-6	Test password with all alphabet or numeric	1. Fill in all required fields 2. Fill in the password field with all alphabet or numeric format. 3. Click the "Register" button	"Passwords should contain at least one uppercase letter, one lowercase letter, one number, and one special character!" Warning will appear
TC_02-7	Test registration with mismatched password and confirmation password fields.	1. Fill in all required fields 2. Fill in the password field 3. Fill the confirm password field with a different password than the password field.	"Passwords do not match!" warning will appear.

		4. Click the “Register” button	
TC_02-8	Test registration with a short password (less than 8 characters).	1. Fill in all required fields 2. Fill in the password below than 8 characters 3. Click the “Register” button	“Passwords must be at least 8 characters! ” warning will appear.

Table 6.11 Test Description Add Store by Store Owner

Test Case ID	TC_03
Test Case Name	Add Store
Actor	Store Owner
Description	This test case allows the tester or user to demonstrate the registration Hotel Halal module by following a few test cases and steps. To start the testing, tester or user needs to: 1. Login as store owner 2. Homepage of store owner display. 3. Click the “Store” page 4. Click the “Add Store” button Then proceed with the following test case.

Test Case	Description	Step	Expected Result
TC_03-01	Test add store with all required fields filled correctly.	1. Fill in all the fields correctly (e.g.: correct format) 2. Click the “Submit” button.	The store should be added successfully when all required information is provided correctly.
TC_03-2	Test add store with not all required fields filled.	1. Not fill in a the fields correctly (e.g : correct format) 2. Click the “Submit" button.	An alert message "All fills are required" should be displayed, and the registration should fail.
TC_03-3	Test to check if logo of store file was uploaded	1. Fill in all field correctly Leave the store logo field empty. 2. Click the “Submit” button. 3.	The system should prompt users with a message: “File upload error or no file uploaded.”

TC_03-04	Test to check if file upload logic	1. Fill in all the fields correctly. 2. Insert the wrong file. 3. Click the "Submit" button.	The system should prompt users to fill in the 'Store Logo' field.

TC_03-05	Test to check if the file is an actual image	1. Fill in all the fields correctly. 2. Insert the image field with a random place image. 3. Click the "Submit" button.	The system should prompt users with a message: "File is not an image."
TC_03-06	Test to check file size with an image file not exceeding the maximum allowed size which is 500Kb.	1. Fill in all fields except image 2. fields correctly Upload image file with 3. size larger than 500Kb. Click the "submit" button.	The system should prompt users with a message: "Sorry, your file is too large."

TC_03-7	Test to check that allow certain file formats only	1. Fill in all fields except 3. upload images with other formats. Click the "Submit" button.	The system should prompt users with a message: "Sorry, only JPG, JPEG, PNG & GIF files are allowed."
TC_03-8	Test to attempt to upload file	1. Fill in all fields 2. Try to attempt the image 3. Click the "Submit" button	The system should prompt users with a message: "Sorry, there was an error uploading your file."

6.4.2 Test Data

The test data to be used in this phase will be a mix of real-life and synthetic data. Real-life data will be used to simulate actual user scenarios and ensure the system behaves as expected in a live environment. Synthetic data will be created to test specific conditions, such as boundary cases or exceptional scenarios, that may not be easily replicated with real-life data.

Table 6.12 Test Data Login

Column Name	**Test Case ID**	**Test Data**	
		Username	**Password**

TD_01-1	**TC_01-1**	Aida	Aida2121!
TD_01-2	**TC_01-2**	Aidaaa	Aida2121!
TD_01-3	**TC-01_3**	Aida	Aida2!
TD_01-4	**TC-01_4**	Empty fields	Empty fields

Table 6.13 Test Data Registration Module

Column Name	**Test Case ID**	**Test Data**						
		User name	**Password**	**Confirm Password**	**Name**	**Email**	**Phone Number**	**User Type**
TD_01-1	**TC_01-1**	Aida	Aida2121!	Aida2121!	Nurul Aida	aida21@gmail.com	01123456788	Shoe enthusiast
TD_01-2	**TC_01-2**	Aida	Aida2121!	Aida2121!	Nurul Aida	aida21@gmail.com	01123456788	Shoe enthusiast

TD_01-3	**TC-01_3**	Empty fields	Empty fields	Empty fields	Empty fields	Empty fields	Empty fields	Empty fields
TD_01-4	**TC-01_4**	Aida	Aida2121!	Aida2121!	Nurul Aida	aida21gmail.com	01123456788	Shoe enthusiast
TD_01-5	**TC_01-5**	Aida	Aida2121!	Aida2121!	Nurul Aida	aida21@	01123456788	Shoe enthusiast
TD_01-6	**TC_01-6**	Aida	Aida	Aida	Nurul Aida	aida21@gmail.com	01123456788	Shoe enthusiast
TD_01-7	**TC-01_7**	Aida	Aida21	Aida21!	Nurul Aida	aida21@gmail.com	01123456788	Shoe enthusiast
TD_01-8	**TC-01_8**	Aida	Aidaaa	Aidaaa	Nurul Aida	aida21@gmail.com	01123456788	Shoe enthusiast

Table 6.14 Test Data Add Store

Test Data ID	**Test Case ID**	**Test Data**

		Store Name	Store logo	Store Branch	Store Manager	Location	Contact Info
TD_03-1	**TC_03-1**	Shoe Under Sea	shoeundersea.jpg	Alor Gajah	Hayati	Alor Gajah	0189261866
TD_03-2	**TC_03-2**	Empty fields	Empty fields	Empty fields	Empty fields	Empty fields	Empty fields
TD_03-3	**C_03-3**	Shoe Under Sea	Empty fields	Alor Gajah	Hayati	Alor Gajah	0189261866
TD_03-4	**C_03-4**	Shoe Under Sea	random.jpg	Alor Gajah	Hayati	Alor Gajah	0189261866
TD_03-5	**C_03-5**	Shoe Under Sea	random.word	Alor Gajah	Hayati	Alor Gajah	0189261866
TD_03-6	**C_03-6**	Shoe Under Sea	shoeundersea.jpg	Alor Gajah	Hayati	Alor Gajah	0189261866
TD_03-7	**C_03-7**	Shoe Under Sea	shoeundersea.pdf	Alor Gajah	Hayati	Alor Gajah	0189261866
TD_03-8	**C_03-8**	Shoe Under Sea	shoeundersea.pdf	Alor Gajah	Hayati	Alor Gajah	0189261866

6.5 Test Results and Analysis

Each test case will be executed, and the results will be recorded. The documentation will include the identification of the test case, the tester responsible for its execution, and the outcome (Success/Fail). In cases where a test case fails, a detailed analysis will be conducted to identify the cause of the failure, and the results will be documented for further investigation. The overall satisfaction of the intended users and the project team with the system's performance will also be assessed and documented, providing insights into the system's readiness for production.

Table 6.15 Test Results and Analysis Registration Module

Module		**Result**		
Test case ID	**Test Data ID**	**Description**	**Pass**	**Failed**
TC_01-1	**TD_01-1**	Valid username and password	✔	
TC_01-2	**TD_01-2**	Invalid and incorrect	✔	
		username		
TC_01-3	**TD_01-3**	Invalid and incorrect password	✔	
TC_01-4	**TD_01-4**	Empty fields	✔	

Table 6.16 Test Results and Analysis Registration Module

Module		**Result**		
Test case ID	**Test Data ID**	**Description**	**Pass**	**Failed**
TC_01-1	**TD_01-1**	Fill required fields correctly.	✔	

TC_01-2	**TD_01-2**	Fill required fields with existing username.	✔	
TC_01-3	**TD_01-3**	Empty fields	✔	
TC_01-4	**TD_01-4**	Test the email without @	✔	
TC_01-5	**TD_01-5**	Test with wrong format email	✔	
TC_01-6	**TD_01-6**	Test password with all alphabet or numeric	✔	
TC_01-7	**TD_01-7**	Test registration with mismatched	✔	
		password and confirmation password fields.		
TC_01-8	**TD_01-8**	Test registration with a short password (less than 8 characters).	✔	

Table 6.17 Test Results and Analysis Add Store By Store Owner

Module		**Result**		
Test case ID	**Test Data ID**	**Description**	**Pass**	**Failed**
TC_01-1	**TD_01-1**	Test add store with all required fields filled correctly.	✔	
TC_01-2	**TD_01-2**	Test add store with not all required fields filled.	✔	
TC_01-3	**TD_01-3**	Test to check if logo of store file was uploaded	✔	
TC_01-4	**TD_01-4**	Test to check if file upload logic	✔	
TC_01-5	**TD_01-5**	Test to check if the file is an actual image	✔	

TC_01-6	**TD_01-6**	Test to check file size with an image file not exceeding the maximum allowed size which is 500Kb.	✔	
TC_01-7	**TD_01-7**	Test to check that allow certain file formats only	✔	
TC_01-8	**TD_01-8**	Test to attempt to upload file	✔	

6.6 User Acceptance Test (UAT)

It is the last procedure that comes under the software testing cycle where real users or clients evaluate the software to confront all the test conditions they demand for an application. Whereas in other test phases, developers or other qual team performs the test, in UAT, the test is conducted by the end users who will be actually using the software in their operation. Testing is therefore centered on confirming whether the software meets all the required business needs and expectations of the users. As a result, UAT plays the critical role of making a reality check and testing the software with real-world use cases. During this phase, any problem we may have encountered is worked out with the intention of having the software ready for use. Generally, the successful outcome of UAT is the literal acceptance of the software by the client which informs others that the software is fit for live use as in a production

setting. The purpose of UAT is to verify that the software behaves as it should and indeed meets the users' expectations and requirements.

The first part of the instrument is a UAT questionnaire where the first question will be related to basic information about the participant such as name, email, gender and age etc. Collecting such variables would ensure that every respondent is accounted for, in case feedback from them is to be given. The email addresses are collected for the purposes of contact mainly for the UAT team to contact the participants, inform them of other details or even request more feedback from the respondents. The gender field is useful in determining the demography of the participants of the study; this is crucial in receiving feedback from different users' perspectives However, it is sensitive data that should be appropriately collected in regards to policies on data privacy. Last but not least, the age field reveals the participants' age distribution to avoid receiving feedback from users with identical experience and include people of various ages. This basic information is very important when categorizing the responses, when analyzing the feedback and for communication purposes during the whole UAT process.

Figure 6. 1 Age And Figure 6. 2 Gender is about the age and gender of the respondent. Among those surveyed there are 10 people, 7 of them are 23 years old, 1 man is 24 years old, 1 man is 26 years old and 1 man is 28 years old.

Age Distribution:

- **7 respondents** (70%) are 23 years old.
- **1 respondent** (10%) is 24 years old.
- **1 respondent** (10%) is 26 years old.
- **1 respondent** (10%) is 28 years old.

Gender Distribution:

- **8 respondents** (80%) are female.

- **2 respondents** (20%) are male.

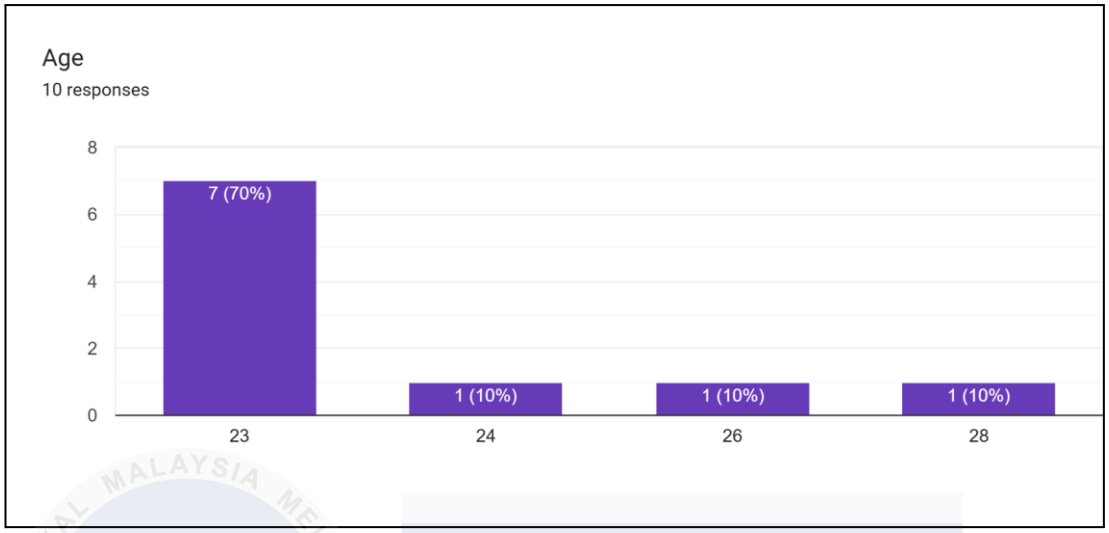


Figure 6.3 Age

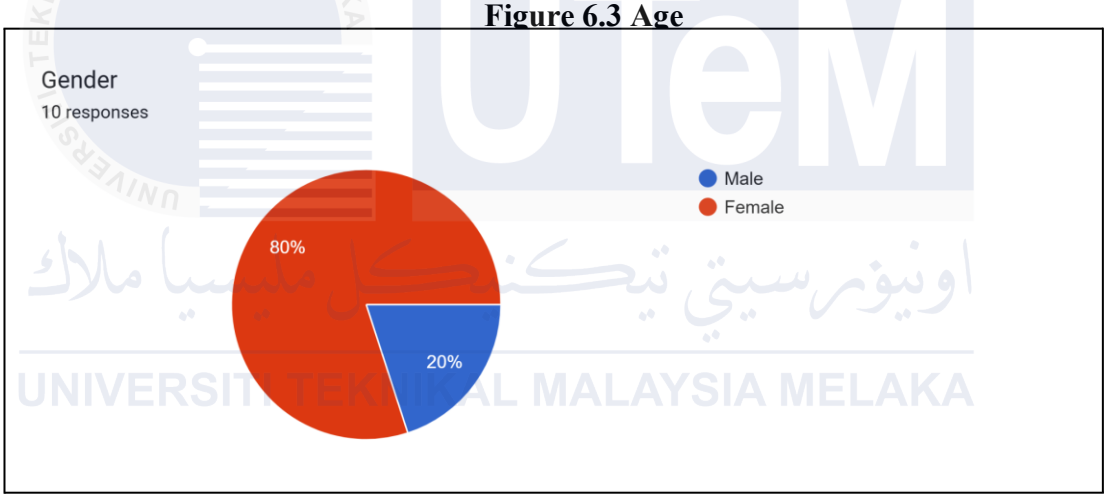


Figure 6.4 Gender

a) Shoe Enthusiast

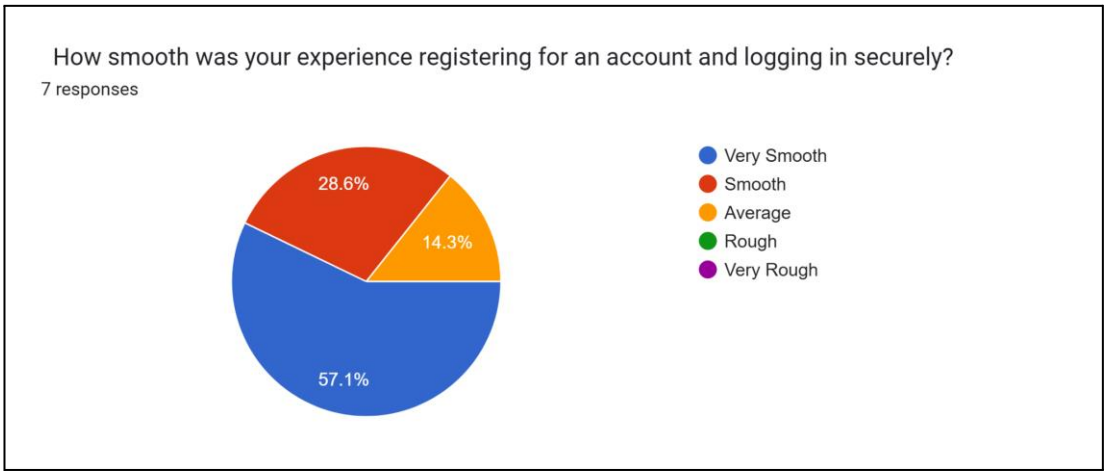


Figure 6.5 Shoe Enthusiast Question 1

- Very Smooth: 4 respondents (57.1%) felt that the experience was "Very Smooth."
- Smooth: 2 respondents (28.6%) rated the experience as "Smooth."
- Average: 1 respondent (14.3%) felt the experience was "Average."
- Rough: 0 respondents (0%) found the process to be "Rough."
- Very Rough: 0 respondents (0%) rated it as "Very Rough."

The majority of participants (85.7%) had a positive experience (either "Very Smooth" or "Smooth") with the account registration and secure login process, while a small portion (14.3%) found the experience "Average." No participants reported difficulties, as no one rated the experience as "Rough" or "Very Rough." This indicates a generally favorable user experience in terms of ease and security for account setup and login.

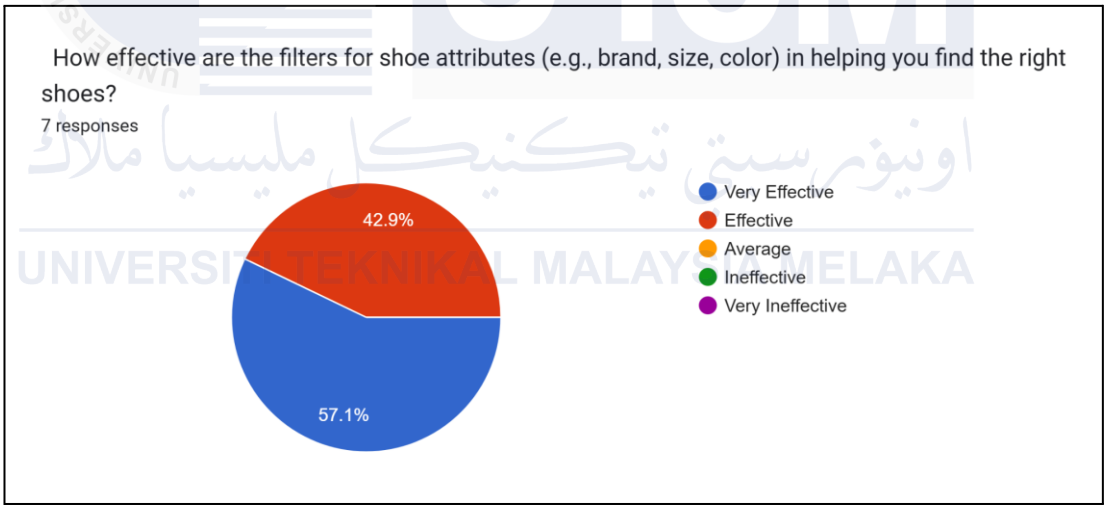


Figure 6.6 Shoe Enthusiast Question 2

- Very Effective: 4 respondents (57.1%) found the filters to be "Very Effective."
- Effective: 3 respondents (42.9%) found the filters "Effective." ● Average: 0 respondents.
- Ineffective: 0 respondents.
- Very Ineffective: 0 respondents.

All participants had a positive experience with the filters, with 100% of respondents rating the filters as either "Very Effective" or "Effective." The majority (57.1%) found the filters highly helpful in finding the right shoes, while the remaining 42.9% still found them effective. No one rated the filters as "Average" or below, indicating that the filter system is working well for most users.

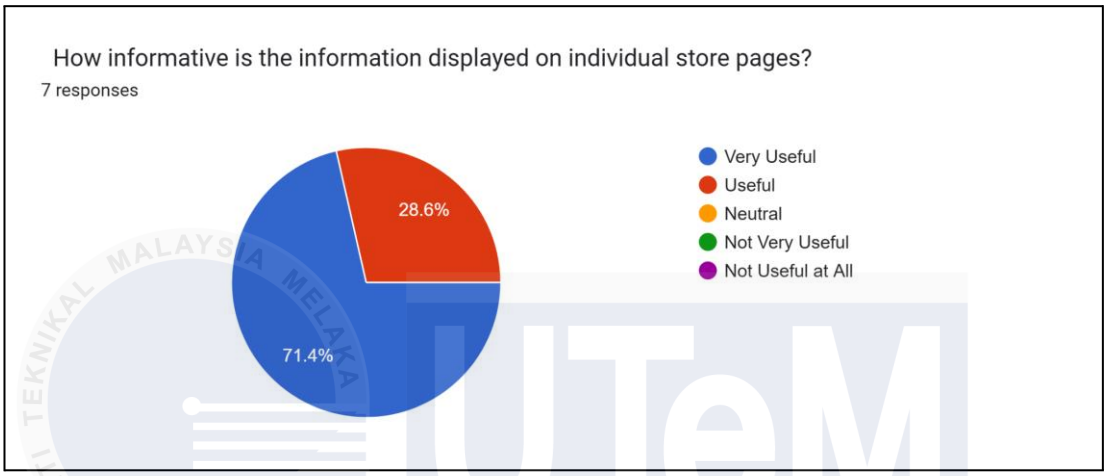


Figure 6.7 Shoe Enthusiast Question 3

- Very Useful: 5 respondents (71.4%) found the information "Very Useful."
- Useful: 2 respondents (28.6%) found it "Useful." ● Neutral: 0 respondents.
- Not Very Useful: 0 respondents.
- Not Useful at All: 0 respondents.

The feedback is overwhelmingly positive, with 100% of respondents rating the information on individual store pages as either "Very Useful" (71.4%) or "Useful" (28.6%). No participants rated the information as "Neutral" or below, indicating that the content provided on store pages is highly informative and meets users' needs.

b) Store Owner

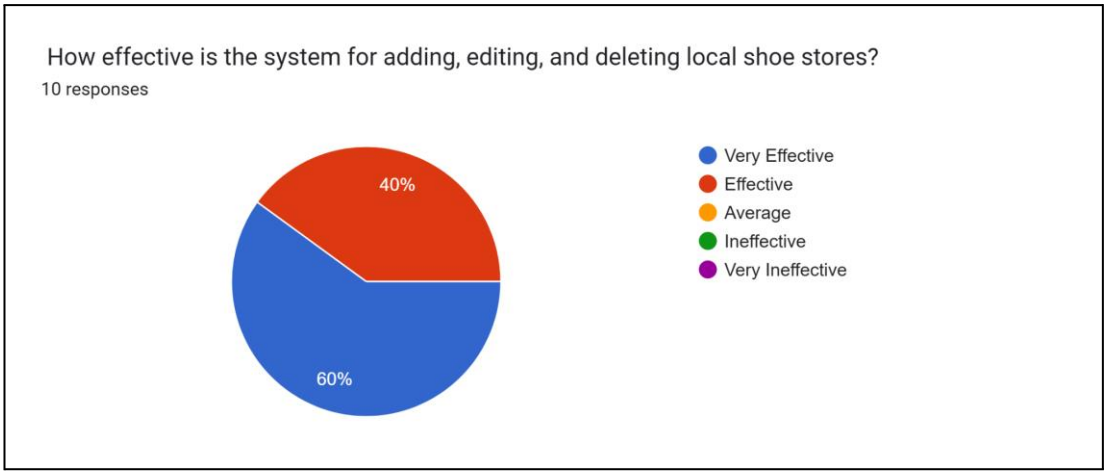


Figure 6.8 Store Owner Question 1

- Very Effective: 6 respondents (60%) found the system "Very Effective."
- Effective: 4 respondents (40%) found it "Effective." ● Average: 0 respondents.
- Ineffective: 0 respondents.
- Very Ineffective: 0 respondents.

The system for managing local shoe stores (adding, editing, and deleting) received entirely positive feedback, with 100% of respondents rating it as either "Very Effective" (60%) or "Effective" (40%). No respondents rated the system as "Average" or below, indicating that the system is functioning well and is considered efficient by all users.

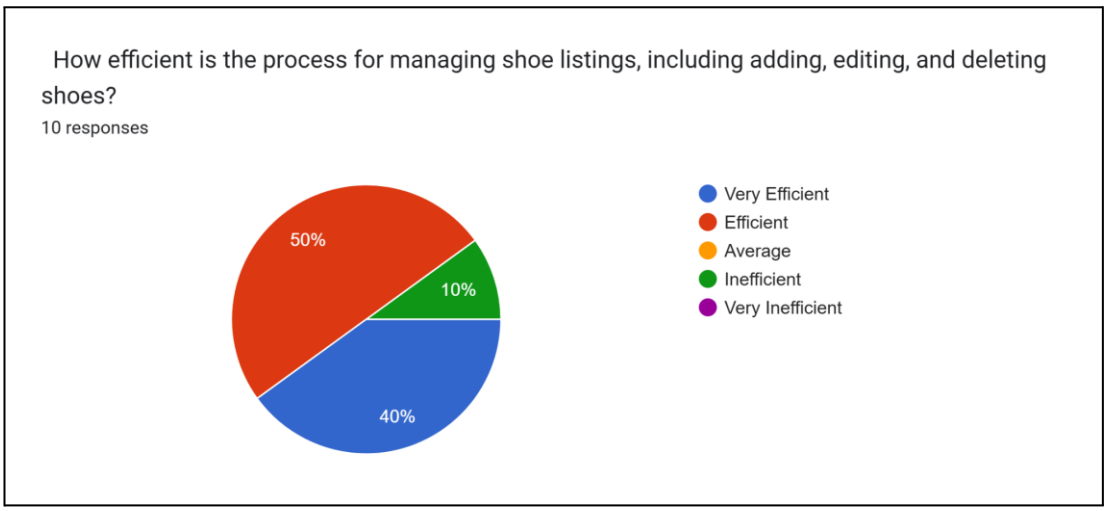


Figure 6.9 Store Owner Question 2

- Very Efficient: 4 respondents (40%) found the process "Very Efficient."
- Efficient: 5 respondents (50%) found it "Efficient." ● Average: 0 respondents.
- Inefficient: 1 respondent (10%) found it "Inefficient." ● Very Inefficient: 0 respondents.

The majority of participants (90%) found the shoe listing management process to be either "Very Efficient" (40%) or "Efficient" (50%), indicating that the system is generally well-received. However, one respondent (10%) found the process to be "Inefficient," suggesting that there may be minor room for improvement to address specific user concerns. Nonetheless, the overall feedback is highly positive, with no one rating it as "Very Inefficient."

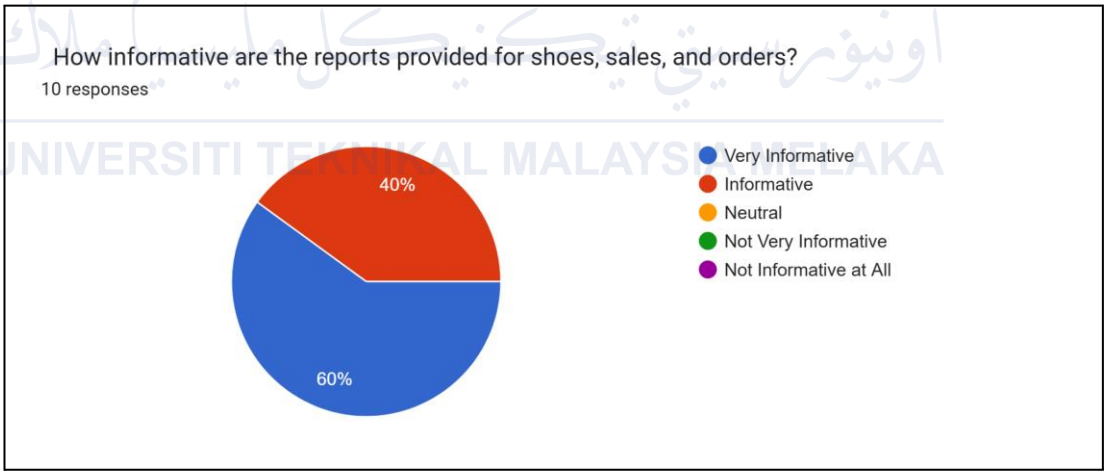


Figure 6.10 Store Owner Question 3

- Very Informative: 6 respondents (60%) found the reports "Very Informative."
- Informative: 4 respondents (40%) found them "Informative." ● Neutral: 0 respondents.
- Not Very Informative: 0 respondents.
- Not Informative at All: 0 respondents.

All participants found the reports to be either "Very Informative" (60%) or "Informative" (40%), showing that the reports are highly effective in conveying the necessary information. No respondents rated the reports as "Neutral" or below, indicating that the reporting system provides valuable and clear insights into shoes, sales, and orders for all users.

c) Admin

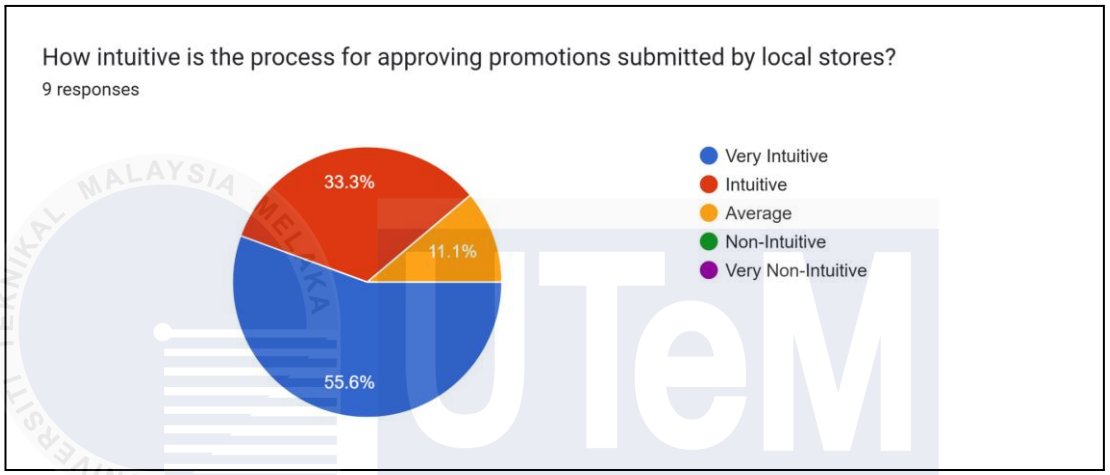


Figure 6.11 Admin Question 1

- Very Intuitive: 5 respondents (55.6%) found the process "Very Intuitive."
- Intuitive: 3 respondents (33.3%) found it "Intuitive."
- Average: 1 respondent (11.1%) found it "Average." ● Non-Intuitive: 0 respondents.
- Very Non-Intuitive: 0 respondents.

The majority of participants (88.9%) found the process for approving promotions to be either "Very Intuitive" (55.6%) or "Intuitive" (33.3%), reflecting a generally positive experience. However, 1 respondent (11.1%) rated the process as "Average," suggesting some minor room for improvement in terms of intuitiveness. No one found the process to be difficult or confusing, as there were no ratings for "Non-Intuitive" or "Very Non-Intuitive."

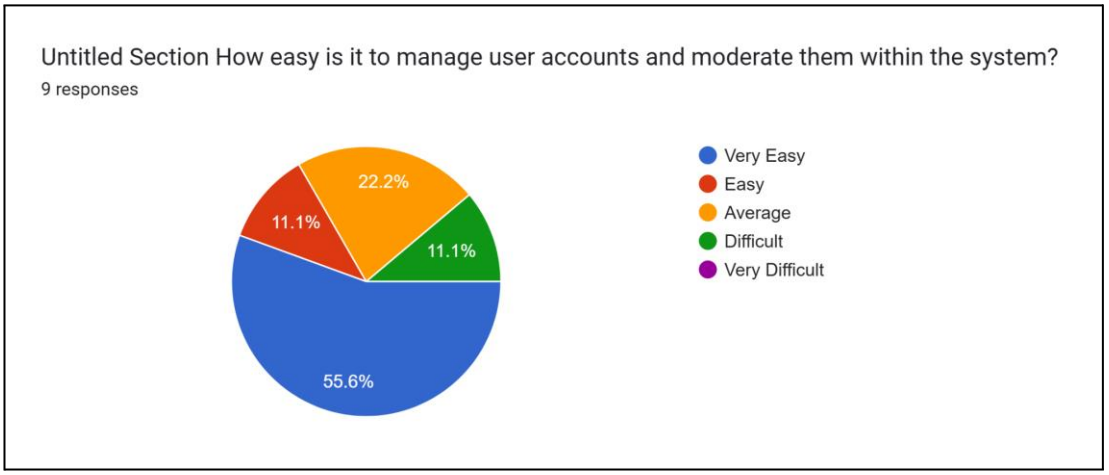


Figure 6.12 Admin Question 2

- Very Easy: 5 respondents (55.6%) found it "Very Easy."
- Easy: 1 respondent (11.1%) found it "Easy."
- Average: 2 respondents (22.2%) found it "Average."
- Difficult: 1 respondent (11.1%) found it "Difficult." ● Very Difficult: 0 respondents.

The majority of participants (66.7%) found managing and moderating user accounts to be either "Very Easy" (55.6%) or "Easy" (11.1%), indicating a generally user-friendly experience. However, 22.2% found it "Average," and 11.1% found it "Difficult," suggesting that there might be areas for improvement to address these concerns. No participants rated the process as "Very Difficult," so while the system is mostly accessible, there are some opportunities for enhancement.

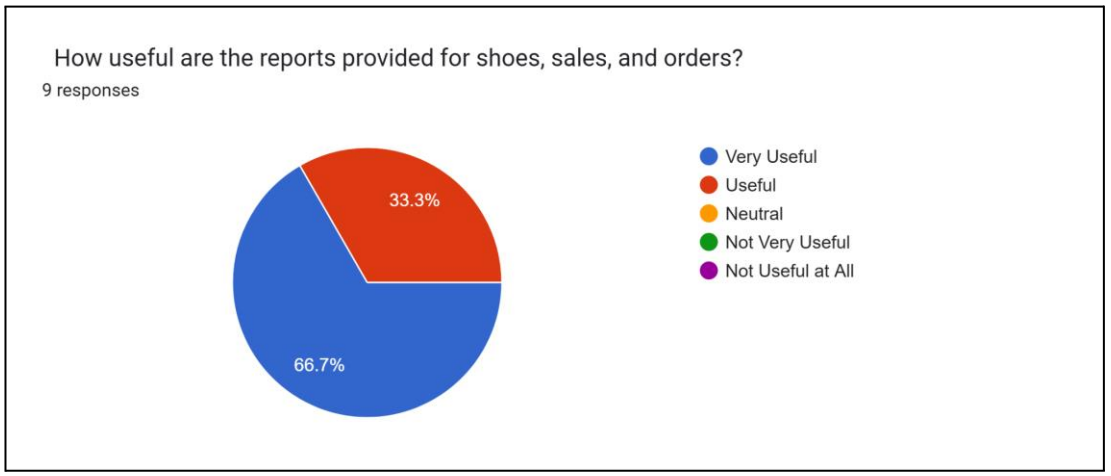


Figure 6.13 Admin Question 3

- Very Useful: 6 respondents (66.7%) found the reports "Very Useful."
- Useful: 3 respondents (33.3%) found them "Useful." ● Neutral: 0 respondents.
- Not Very Useful: 0 respondents.
- Not Useful at All: 0 respondents.

All respondents found the reports to be valuable, with 100% rating them as either "Very Useful" (66.7%) or "Useful" (33.3%). This indicates that the reports effectively meet the needs of the users and provide significant value. No feedback was given indicating that the reports were less useful, reflecting strong satisfaction with the reporting system.

6.7 Conclusion

This chapter has detailed the test plan, strategy, design, and execution results of the testing phase. The next steps involve addressing any identified issues, refining the system based on feedback, and preparing for the final deployment phase. The successful completion of this testing phase will ensure that the system is robust, secure, and ready to meet the needs of its users.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

As the FootEase project comes to a close, it is important to take review of its journey and the major accomplishments made along the way. A local directory shopping system for shoes, FootEase was created to provide a smooth experience by easily linking shoe enthusiasts with stores. This innovative platform solves the issues that shoe enthusiasts and stores in the contemporary retail industry face. This last section will summarize the system's strengths, draw attention to its shortcomings,

suggest changes, and look at commercialization options in order to make sure FootEase keeps improving and successfully serving its users' requirements.

7.2 Achievement

Most striking of all the enhancements of the “FootEase” project is the incorporation of a Rich Search capability that assists shoe lovers in searching for shoes by various features like category, brand color, among others. The shoes lovers can also handle their accounts in an efficient manner most prominently regarding the favorite list which is yet another advantage for the shoe shoppers. The main aim of the project was to bring a direct link between shoe lovers and stores within the region.

The establishment of independent store sites not only containing detailed brand shoe assortments and current specials is another success. From this improvement, shoe enthusiasts are in a better place to make informed decisions with their purchases given their eye opener.

Furthermore, the ability of the system to monitor and analyze the overall store performance as well as the annual sales have been informative to the administrators and owners of the business enterprise. This tool also assists the local businesses to optimize its functioning and development by enabling better management and business planning. In response to this problem, the “FootEase” solution offers solutions to develop modern shoes’ user experience and security in addition to helping local shoe retailers be more visible or profitable online. It is evident that the programme has generated great results in promoting Community involvement and enhancing the economy in the era of advanced technology.

7.3 Project Limitation

Despite the successes that have been achieved in the “FootEase”, there are some mistakes that have to be corrected. Deficiencies in the internet connection can

be problematic especially to users residing in areas that may be considered as the periphery of the society. This might not always be the case since local shoe businesses have to respond and take an active part for the platform to succeed. This is so since some of the old devices that most users still employ in using the internet may cause some compatibility issues. So, to maintain the users' confidence, the information and data has to be secured well, and the privacy has to be ensured well also. As the scale of the platform increases, the platform should has the ability to handle more traffic.

It is rather challenging to lure and maintain consumers' interest, especially when the competitor is already established internet services. Another reason is also that timely customer support is needed because some problems may negatively impact the overall experience of a user. Solving these problems will help towards the continuous success and growth of the "FootEase".

7.4 Suggestion and Improvement

Many recommendations may be made which may be considered important to implement change to make "FootEase" better. That is why it is necessary to improve the internet connection and, for this purpose, alternative offline features as well as PWA technologies are crucial. With low connectivity, one can still be able to search for the shoe of a particular model and also view the various specials that are being offered. More local firms have to be involved; incentives and publicity efforts including offering free trials, lower prices and advertising help the local firm to get and retain business. It is highly recommended that focusing on what has been achieved by the involved stores may go along away to encouraging people to participate and have confidence in the concept.

Another feature that has to be pushed hard is devices compatibility, that is to ensure that the platform is fully responsive and works on a variety of devices and browsers. That is why compatibility issues may be initially detected and further addressed through testing on different web browsers and older devices. Another way

is to address the issue of accessibility through providing the users with a lightweight version of the platform if they operate outdated hardware and software.

Making the user interface and user experience of the platform better and making updates to the platform more constantly, based on the users' feedback and the latest trends and trends, and introducing loyalty programs, specific recommendations, and exclusive deals will help in increasing the users' engagement and loyalty. This also supports objectives that we want to position "FootEase" as not being the same as competitors through including unique selling propositions such as incorporating supporting small businesses within the neighborhood, engaging every customer uniquely, and participating in community affairs. There is the possibility of enhancing the level of visibility and, consequently, the credibility by focusing on advantages of purchasing products and services from local businesses as well as the products that are unique and cannot be found elsewhere and working with key opinion leaders and other stakeholders.

Last but not the least; an effective management of this platform takes keen consideration towards enhancing customer support this will ensure that the consumer can effectively maneuver on the platform. This can be done by ensuring that the organization is equipped with a good support system that addresses the users' issues promptly and in a number of ways. More so, recurrent training for the customer service representatives to handle different issues would further improve the user experience. If these recommendations are to be implemented, it will help "FootEase" overcome the problems it currently faces and provide a better future and sustainable platform for both shoe lovers and shoe stores.

7.5 Potential Commercialization

Refocusing the "FootEase" project is revealing tremendous possibilities for moving to the commercial level due to its competitive advantage and the idea to use

the constantly developing sphere of the online platform. One of these is the use of the membership model is one of the approaches that can be utilized. A tiered pricing system with basic, premium and enterprise options that come with additional features such as advanced store listings, detailed analytics and marketing that enables the local store businesses to sign up onto the platform. Moreover, consumers may be offered a membership service that will be beneficial for the consumer as well as opening special deals and unique discounts for membership.

Very viable opportunities for commercialisation are also provided by advertising and promotions. Shoe businesses may further advertise their site through paying the membership fee and having their sites being listed as featured sites at the top of the home page and on the search results list. For instance, to extend users' groups and geographic areas interested in the offer, the retailer might use advertising packages within the platform and conduct promotion campaigns.

Achieving a high level of affiliation and partnership is one of the major strategies for enhancing the platform's revenue capacity. "FootEase" can expand its customer base and popularity by cooperating with prominent shoe makers and brands for developing special offers and limited edition products. Consulting with other shoe stores located close by to help in the organizing of pop-up shops and community events may help in improving users' participation, promoting companies located in the same region, and generating income through the event participation fees.

7.6 Conclusion

Conclusion 'FootEase' has done a good job to innovate the history of shoe shopping online by bridging the gap between the target audience and local stores. The performance of the platform has enhanced the number and quality of the users' interactions while making perfect sense of stores' product listings and their effectiveness. However, there are some problems which are also associated with internet connection intermittent access, compatibility of the devices, privacy of the data, user engagement and efficient customer support; the proposed enhancements seek to solve them. There are significant opportunities for the commercialization of

“FootEase” through subscription models, advertising and promotions and cooperation, which gives it the potential to be positioned as one of the leading players in the online shoe retail market. That’s why, by acting innovative and engaging the community, “FootEase” can make a longer-lasting impression on the industry.

REFERENCES

Vionic Shoes: Comfortable Stylish Shoes, Sandals, Boots & More. (n.d.). Www.vionicshoes.com. https://www.vionicshoes.com/

JD Sports Malaysia | Sneakers | Sports Fashion & Clothing. (n.d.). JD Sports. https://www.jdsports.my/

Zappos. (2022). Shoes, Sneakers, Boots, & Clothing + FREE SHIPPING | Zappos.com. Zappos.com; Zappos. https://www.zappos.com/

What Is a DBMS? (2020, October 20). LearnSQL.com. https://learnsql.com/blog/what-is-dbms/

West, C. (2021, November 17). *30+ flowchart examples (with guide, tips and templates)*. Visme.co. https://visme.co/blog/flowchart-examples/