UTHM
Universiti Tun Hussein Onn Malaysia

BIS30703:WEB DEVELOPMENT

TITLE: ETS FOOD ORDERING SYSTEM

ABSTRACT

The KTMB-ETS Food Ordering System project is designed to enhance the food service experience for passengers and restaurant operators by offering a modern, efficient, and user-friendly platform. This system is developed to address key challenges in traditional food ordering processes, such as miscommunication, time inefficiencies, and operational bottlenecks, while improving overall service quality and customer satisfaction.The project aims to provide a seamless interface that allows customers to browse menus and customize orders. Meanwhile, the system offers restaurant operators tools for real-time order management. The development process emphasizes meeting user and system requirements, ensuring functionality, convenience, and compatibility across multiple platforms.Built using web technologies such as HTML, CSS, JavaScript, and PHP, and supported by robust database management via MySQL, the system is tailored to KTMB-ETS users, offering accessibility on desktops and mobile devices. Key features include role-based access control, secure payment processing, and feedback collection mechanisms.The results demonstrate that the KTMB-ETS Food Ordering System significantly improves customer experience and operational efficiency, making it a valuable tool for modernizing food service operations. With its reliable design and potential for future enhancements such as loyalty programs, order personalization, and advanced analytics the system lays the foundation for continued growth and innovation in the food service industry.

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

The food service is one of the most dynamic, fast-moving, and ever-growing industries in the world, resulting from ever-changing consumer demands, along with technological changes. In today's digital world, all customers want to have convenience and efficiency hence, online food ordering systems are becoming popular. It changed the way people used to deal with restaurants when ordering food with such ease, having less waiting time, and creating more customer satisfaction.

The ETS Food Ordering System is required to meet the demand for a modern, user-friendly platform where customers can order food with ease and convenience, while restaurant operators have convenience in operation. It will, in the end, enable customers to view menus, place orders with modifications, and make payments online, while restaurant staff efficiently manage incoming orders, inventory, and customer preferences. This system integrates technology into food services with the aim of minimizing errors, improving the quality of services, and operational efficiency.

It also describes how ETS Food Ordering Systems solve some of the problems identified with conventional ways of ordering foods such as miscommunication in the placement of orders, time wastage in placing orders or processing payments for orders, and delays in service. As this new ordering system automates certain essential facets of ordering food, the trend now is user-friendly and hence fits within the context of modern times which technological advancement of food industries.

Last but not least, this project meets the current demand for digitization solutions for food service industries as a bridge between users and service providers with an innovative platform that will meet modern consumer expectations and support business operations effectively.

1.2 Objectives

The primary objectives of the ETS Food Ordering System are as follows:

1. To enhance user experience with user-friendly interface for customers to browse menus, place orders, and make payments.
2. To develop a web-based responsive design using HTML, CSS and PHP.
3. To test business operations that assist food establishments in managing orders, tracking inventory, and generating sales reports.

1.3 Scope of the Project

The ETS Food Ordering System is specifically designed for food establishments that wish to digitize their operations. The scope of the project includes the following features:

Table 1.1: Project Scope

Category	Scope Description
Target Users	- Users of ETS. - Restaurant or cafeteria operators within ETS premises.
System Platform	- Web-based applications accessible via desktop and mobile devices.
User Roles	- Customers: Browse menus, place orders, and make payments. - Admin/Restaurant Operators: Manage menus, monitor orders, and process payments.
Features for Customers	- View restaurant menus with images and descriptions. - Add items to the cart and customize orders - Make payments via online payment gateways - Track order status
Features for Restaurant Operators	- Add, update, or remove menu items. - View, approve, and manage incoming orders in real-time. - Manage inventory to ensure menu accuracy. - Generate sales and order reports for analysis.
Order Management	- Real-time order placement and notifications for both customers and restaurant operators.
Payment System	- Support for secure online payment gateways
Accessibility	- Designed to be user-friendly web-based and intuitive for all users, including first-time customers.

1.4 Significance of Developing the Website

The development of the ETS Food Ordering System holds substantial significance for both customers and businesses. For customers, it offers unparalleled convenience, enabling them to order their meals with just a few clicks, saving time and effort. Additionally, the system reduces the likelihood of order errors, ensuring accurate delivery of items.

For businesses, the ETS Food Ordering System serves as a powerful tool for enhancing operational efficiency and expanding market reach. By adopting a digital platform, food establishments can attract tech-savvy customers, improve order management, and gain valuable insights through sales analytics. Furthermore, the system's secure payment methods and real-time order tracking enhance customer trust and satisfaction, fostering loyalty and repeat business.

In summary, the ETS Food Ordering System represents a significant step toward modernizing the food industry, addressing the evolving needs of both customers and businesses while fostering growth and innovation.

CHAPTER 2

TECHNICAL DETAILS AND REQUIREMENTS OF THE WEBSITE

2.1. INTRODUCTION

A comprehensive understanding of the requirements is essential for the development of the KTMB-ETS Food Ordering System in order to guarantee that the system satisfies user needs and functions well in its intended scenario (Cook & Dupras, 2004). This chapter highlights the different requirements which function as the project's foundation, classifying them into four categories: software, hardware, system, and user requirements.

To guarantee that the system is easy to use and satisfies their unique needs for functionality and convenience, user requirements center on the expectations and wants of end users, such as customers and employees. System requirements specify the characteristics and functionalities that the meal ordering system must have in order to operate successfully and efficiently. Software requirements outline the required programs and tools that will make the system's operation easier, whereas hardware requirements describe the actual parts required to run the system.

Our goal is to develop a reliable and effective food ordering system that improves the overall travel experience for KTMB-ETS customers by expediting the ordering procedure by carefully examining these needs. As a crucial stage in the project, this chapter directs the design and implementation stages to meet the needs and expectations that have been defined

2.2. USER REQUIREMENT

User requirements are essential for the system, as they outline the specific needs and expectations of the end-users, including passengers and staff. These requirements focus on creating an intuitive and user-friendly interface that facilitates easy navigation, efficient order placement, and seamless payment processing. Figure 2.1 shows the user requirement for the system.

Table 2.2: User Requirements

No	User Requirements	Description
1	User friendly interface	Easy navigation for all user
2	Account management	Able to create, update and delete user informations in their account
3	Menu browsing	Can access the menu with description and price
4	Real-time order tracking	Updates on order preparation and status

5	Payment option	Support for multiple payment methods
6	Feedback mechanism	User can provide feedback on the food and services

2.3. SYSTEM REQUIREMENT

The purpose of the requirement analysis is to identify the actual function or process of the Food ordering website and to understand the system's requirements. The requirement analysis for system shows the functional, non-functional and security needs.

2.3.1 FUNCTIONAL REQUIREMENT

The functional requirements describe the functionality of the system and its component, which allow the user to input data and expect an output after the activity is performed (Malan et al., 2001). Table 2.2 Shows the functional requirement for the system.

Table 2.2: Functional Requirements

No.	Functional Requirement
1	System should provide registration page for customer
2	System should provide login page for users
3	System should allow staff to insert, update and delete product
4	System should allow staff to update order process
5	System should allow customer to add, update and delete item in the cart
6	System should allow customer to update and delete their profile management detail
7	System should allow customer to view order status
8	System should allow customer to view past orders

2.3.2 NON-FUNCTIONAL REQUIREMENT

Table 2.3 shows the non-functional requirement for the system which describe about the performance of the system (Glinz, 2007).

Table 2.3: Non-Functional Requirements

No	Non-Functional Requirements
1	The system should be available anytime, ensuring high availability
2	The system should have a user friendly interface that is easy to navigate

3	The system should be compatible with major web services
4	The system should be able to stored information in the database

2.3.3 SECURITY REQUIREMENT

Table 2.4 shows the security requirements that are available in the system.

Table 2.4: Security Requirements

No	Security Requirements
1	System should be able to authenticate and authorize user credential
2	The system should include password complexity for users
3	The system should be able to store the password after being hashed in the database
4	The system should limit login attempts after several failed login
5	The system should be able to stored information in the database
6	The system should be able to validate user input such as e-mail, phone number and more before storing it in the database
7	The system should implement role-based access control to restrict access based on user roles

2.4. HARDWARE REQUIREMENT

Hardware requirements are crucial for the successful implementation of the ETS Food Ordering System, as they define the physical components necessary to support the application's functionality and performance. Table 2.5 shows the hardware requirement for the system.

Table 2.5: Hardware Requirements

Hardware	Specification	Description
Devices	Smartphones, tablets and laptop	Compatibility with various devices that can access the web
Web server	Chrome, Microsoft Edge, Mozilla	A reliable server to host the application and database, capable of handling multiple concurrent users.
Network Infrastructure	4G / 5G, internet speed	High Speed internet connection to ensure smooth data transfer and accessibility for users

2.5. SOFTWARE REQUIREMENT

Software requirements describe the specific applications and technologies needed to build and operate the system effectively. Table 2.6 shows the software requirements for the system.

Table 2.6: Software Requirements

Software	Specification	Description
Database	MYSQL	Database Management System
Programming Language	PHP	Main language used for developing server-side logic of the system
Front-End development	HTML, CSS, Javascript	Developing the user interface and enhance user experience

CHAPTER 3

DESIGN

3.

3.1 USE CASE DIAGRAM

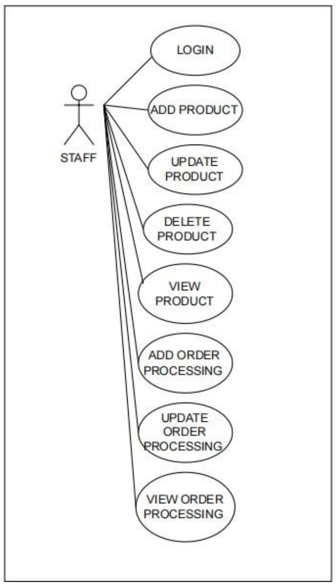


Figure 3.1: Use Case Diagram for Staff

The Figure 3.1 shows what a staff member can do in the ETS Food Ordering System. First, any staff member must log in to access the system's features. Once logged in, they have several main responsibilities that are split into two categories: product management and order management. For product management, staff members can perform four main tasks: they can add new products to the menu, update existing product details (like prices or descriptions), delete products that are no longer available, and view all products in the system.

For order management, staff members have three main functions: they can add new orders into the system, update the status of existing orders (for example, marking them as "in preparation" or "ready for pickup"), and view all orders to keep track of what needs to be prepared. All these functions are connected directly to the staff actor, showing that staff members have direct access to these features after logging in. This design ensures that staff members have all the necessary tools to manage both the menu items and customer orders efficiently in one integrated system.

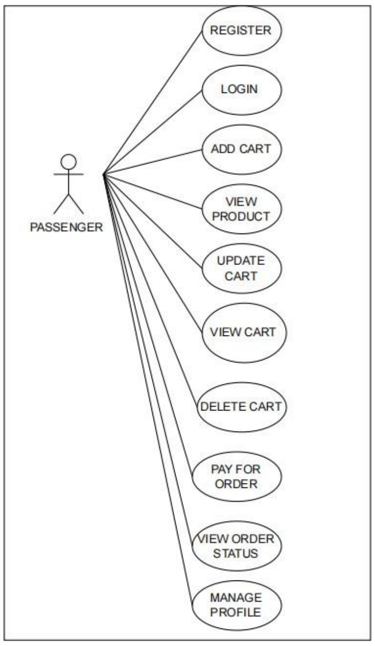


Figure 3.2: Use Case Diagram for Passenger

The Figure 3.2 illustrates all the actions that customers (or passengers) can perform in the ETS Food Ordering System. To begin using the system, new customers need to register for an account, and returning customers can simply log in to access the system's features. Once logged in, customers have several options related to shopping and ordering food. They can view products available in the menu and add items they want to purchase to their cart. When it comes to managing their shopping cart, customers have complete control. They can view what's currently in their cart, update quantities of items, or delete items they no longer want to purchase.

For the ordering process, customers can proceed to pay for their order once they're satisfied with their cart contents. After placing an order, they can keep track of its progress by viewing their order status. Additionally, the system allows customers to manage their profile, where they can update their personal information such as contact details or delivery preferences. This design makes the ordering process straightforward for customers, following a logical flow from browsing products to completing their purchase, while also giving them tools to manage their account and track their orders.

3.2 CLASS DIAGRAM

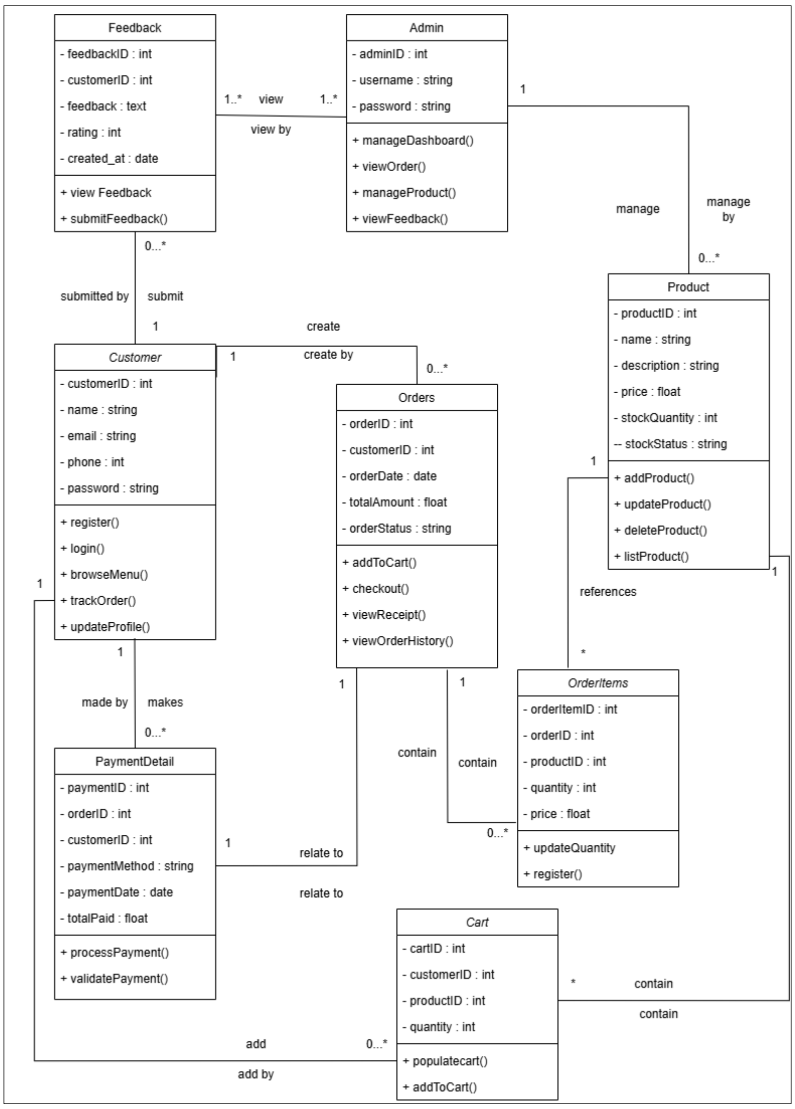


Figure 3.2: Class Diagram

The ETS Food Ordering System's relationships are primarily centered around customer interactions. Customers make payments through PaymentDetails, which are directly related to specific Orders. When customers create Orders, these orders contain multiple OrderItems, which in turn reference specific Products from the menu. Customers can also submit Feedback about their experience and add items to their Cart before purchase, with the Cart containing references to Products they wish to order.

The administrative side is managed through the Admin class, which has two key relationship functions: viewing customer Feedback and managing Product details in the system. The Cart acts as a temporary storage system between Customers and Products before an order is finalized, while PaymentDetails maintains the financial records of each transaction by relating customer payments to their specific orders. This structure ensures a smooth flow from product selection to order completion, while maintaining proper records of all transactions and customer interactions.

3.3 DATABASE DIAGRAM

3.3.1 DATA DICTIONARY

Table 3.1: Customer Data Dictionary

Field Name	Key	Data Type	Constraints
customerID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
name		VARCHAR (100)	NOT NULL
email	Unique Key	VARCHAR (100)	UNIQUE, NOT NULL
phone		VARCHAR (15)	NOT NULL
password		VARCHAR (255)	NOT NULL

The Customer table stores information about customers who use the platform. It includes their unique ID, name, email, password, and phone number. The email field is unique to ensure no duplicate accounts, and the table serves as a central point for customer-related data.

Table 3.2: Admin Data Dictionary

Field Name	Key	Data Type	Constraints
adminID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
username	Unique Key	VARCHAR(50)	UNIQUE, NOT NULL
password		VARCHAR(255)	NOT NULL

The Admin table holds the credentials for administrators of the system. Each admin has a unique ID. Username and password, allowing secure access to administrative interfaces.

Table 3.3: Product Data Dictionary

Field Name	Key	Data Type	Constraints
productID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
name		VARCHAR(100)	NOT NULL
description		TEXT	
price		DECIMAL(10,2)	NOT NULL
stockQuantity		INT	NOT NULL

stockStatus		VARCHAR(100)	NOT NULL

The Product table contains details about the products available for purchase. It includes fields for the product's name, description, price, stock quantity, and stock status, which help manage inventory and display product information to customers.

Table 3.4: Orders Data Dictionary

Field Name	Key	Data Type	Constraints
orderID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
customerID	Foreign Key	INT	NOT NULL, REFERENCES Customer(customerID)
orderDate		DATETIME	DEFAULT CURRENT_TIMESTAMP
totalAmount		DECIMAL(10,2)	NOT NULL
orderStatus		VARCHAR(100)	NOT NULL

The Orders table tracks customer orders. Each order is linked to a customer and includes the order date, total amount, and status. It forms the core of the order management system.

Table 3.5: Payment Detail Data Dictionary

Field Name	Key	Data Type	Constraints
paymentID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
orderID	Foreign Key	INT	NOT NULL, REFERENCES Orders(orderID)
customerID	Foreign Key	INT	NOT NULL, REFERENCES Customer(customerID)
paymentMethod		VARCHAR(100)	NOT NULL
paymentDate		DATETIME	DEFAULT CURRENT_TIMESTAMP
totalPaid		DECIMAL(10,2)	NOT NULL

The PaymentDetail table stores payment-related information for orders. It records the payment ID, order ID, customer ID, payment method, payment date, and total paid. This table ensures that payments are properly tracked and linked to their respective orders and customers.

Table 3.6: Feedback Data Dictionary

Field Name	Key	Data Type	Constraints
feedbackID	Primary Key	INT	AUTO_INCREMENT, NOT NULL

customerID	Foreign Key	INT	NOT NULL, REFERENCES Customer(customerID)
feedback		TEXT	NOT NULL
rating		INT	NOT NULL
created_at		TIMESTAMP	Default: CURRENT_TIMESTAMP, Not Null

The Feedback table collects customer feedback about the platform or products. It includes fields for the customer's name, email, contact number, feedback content, and rating. This table helps improve customer experience.

Table 3.7: Cart Data Dictionary

Field Name	Key	Data Type	Constraints
cartID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
customerID	Foreign Key	INT	NOT NULL, REFERENCES Customer(customerID), ON DELETE CASCADE
productID	Foreign Key	INT	NOT NULL, REFERENCES Product(productID), ON DELETE CASCADE
quantity		INT	NOT NULL

The Cart table manages customers' shopping carts. It associates customers with the products they have added to their cart and tracks the quantity of each product. This table enables customers to save and review items before completing a purchase.

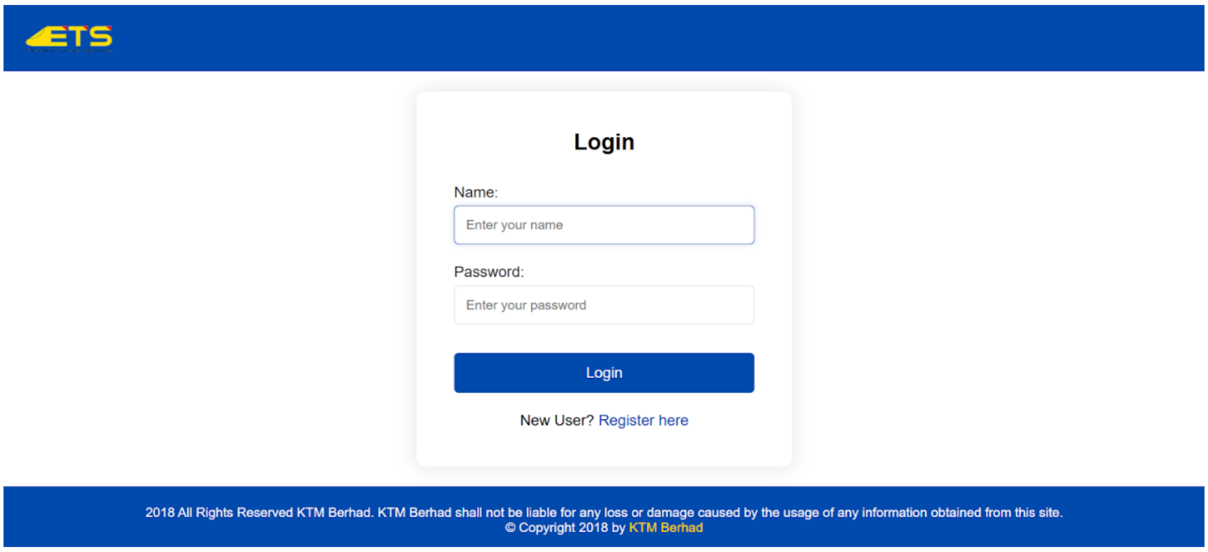
Table 3.8: Order Items Data Dictionary

Field Name	Key	Data Type	Constraints
orderItemID	Primary Key	INT	AUTO_INCREMENT, NOT NULL
orderID	Foreign Key	INT	NOT NULL, REFERENCES Orders(orderID) ON DELETE CASCADE
productID	Foreign Key	INT	NOT NULL, REFERENCES Product(productID), ON DELETE CASCADE
quantity		INT	NOT NULL
price		DECIMAL(10,2)	NOT NULL

The OrderItems table records the specific products included in each order. It links orders to products, tracks the quantity of each product ordered, and records the price at the time of purchase. This table facilitates detailed tracking of order contents.

3.4 INTERFACE DESIGN

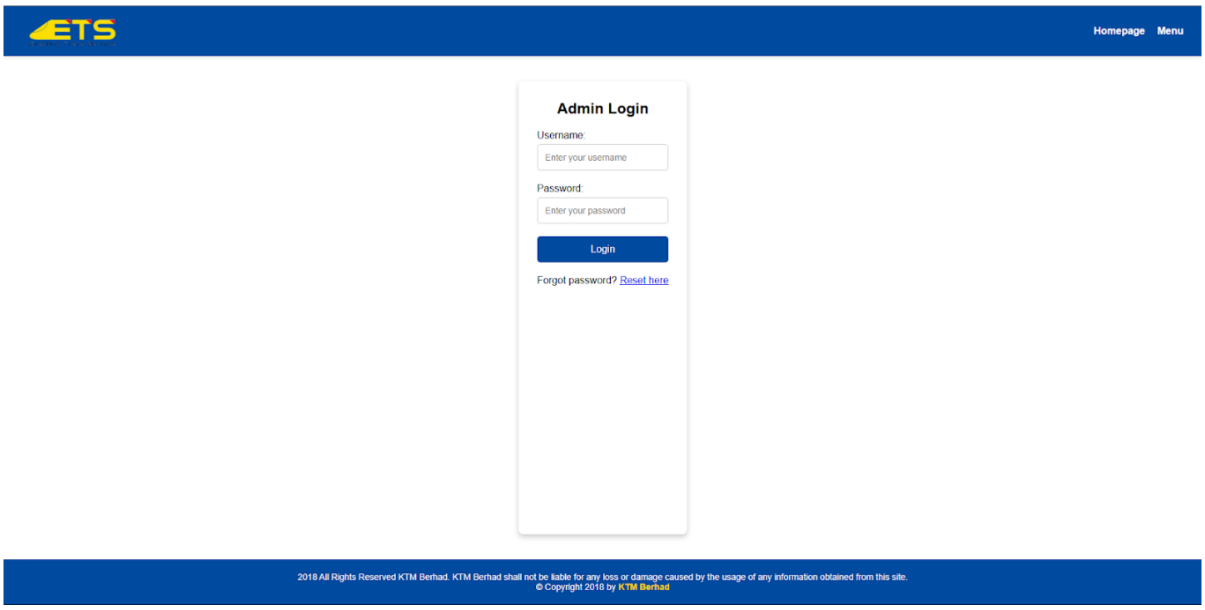
3.4.1 AUTHENTICATION INTERFACE



2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.4: Customer Login Page

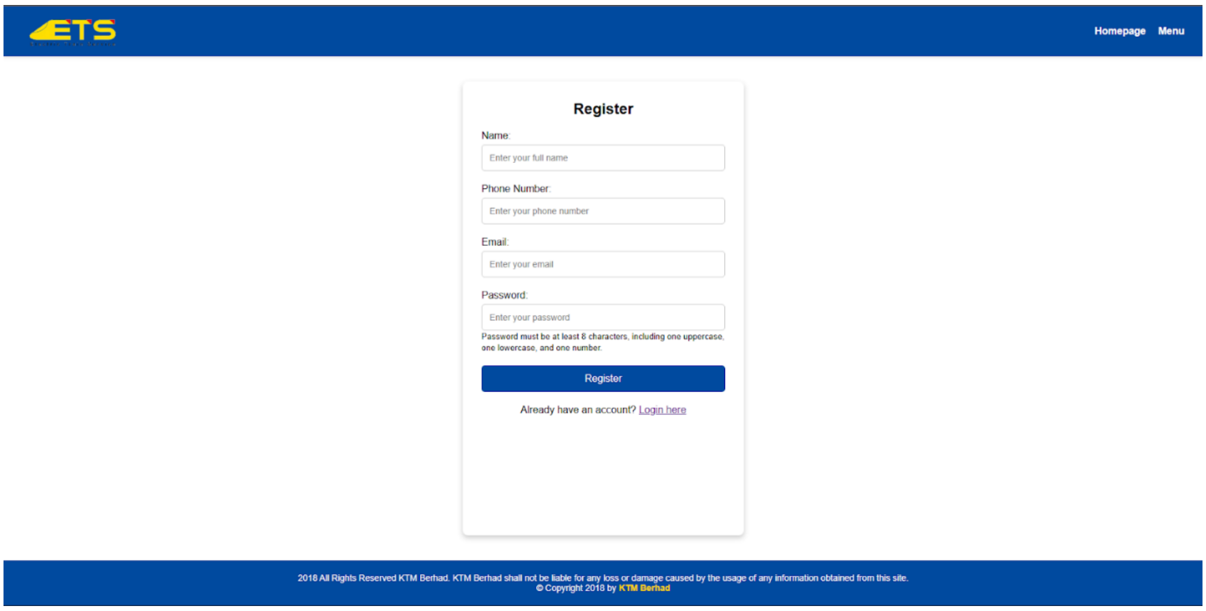
The login page serves as the main entry point for users to access their accounts. It features a simple and secure form where users enter their name and password. New users who don't have an account yet can easily find a link to the registration page to sign up.



2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.5: Admin Login Page

The staff login page is a secure gateway for administrators to access the system's management features. It's designed to be straightforward, requiring only a username and password. This simple but secure approach ensures that only authorized staff members can access the administrative functions of the system.



Register

Name:
Enter your full name

Phone Number:
Enter your phone number

Email:
Enter your email

Password:
Enter your password

Password must be at least 8 characters, including one uppercase, one lowercase, and one number.

Register

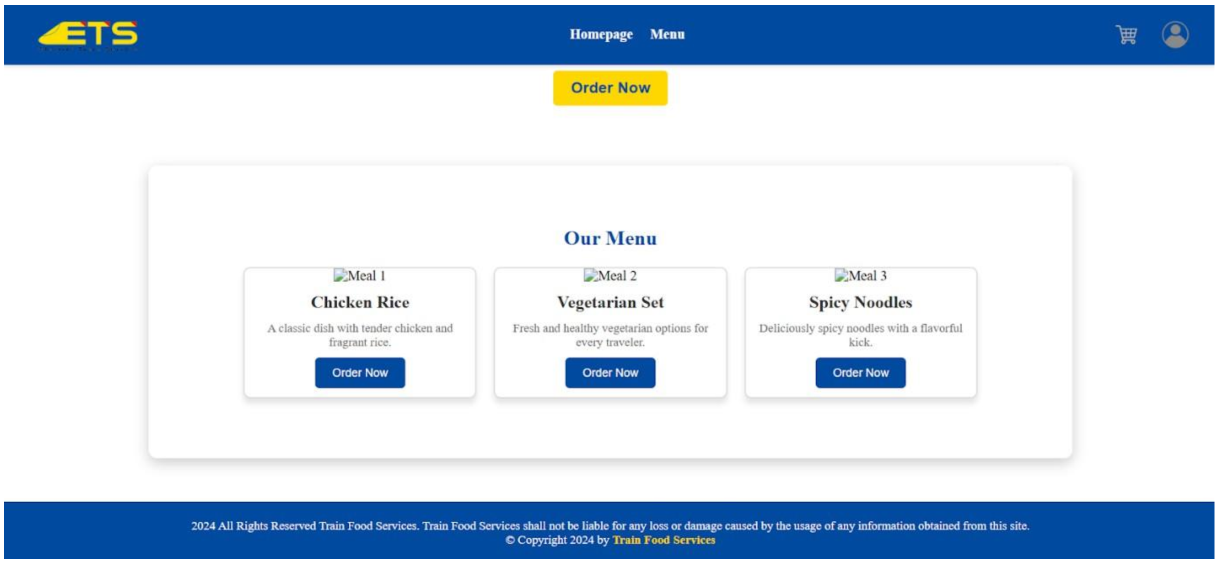
Already have an account? Login here

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.6: Registration Page

The registration page welcomes new users with a straightforward form to create their account. Users need to provide basic information including their name, email address, password, and phone number. The system checks this information to make sure no duplicate accounts are created, ensuring each user has a unique profile in the system.

3.4.2 CUSTOMER INTERFACE



Our Menu

Meal 1
Chicken Rice
A classic dish with tender chicken and fragrant rice.
Order Now

Meal 2
Vegetarian Set
Fresh and healthy vegetarian options for every traveler.
Order Now

Meal 3
Spicy Noodles
Deliciously spicy noodles with a flavorful kick.
Order Now

2024 All Rights Reserved Train Food Services. Train Food Services shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2024 by Train Food Services

Figure 3.7: Home Page

The home page is designed to be welcoming and informative, showcasing the best the platform has to offer. It displays featured food items and current promotions prominently. Users can easily find the main menu and access their profile from here. A prominent "Order Now" button takes customers directly to the menu page to start their food ordering journey.

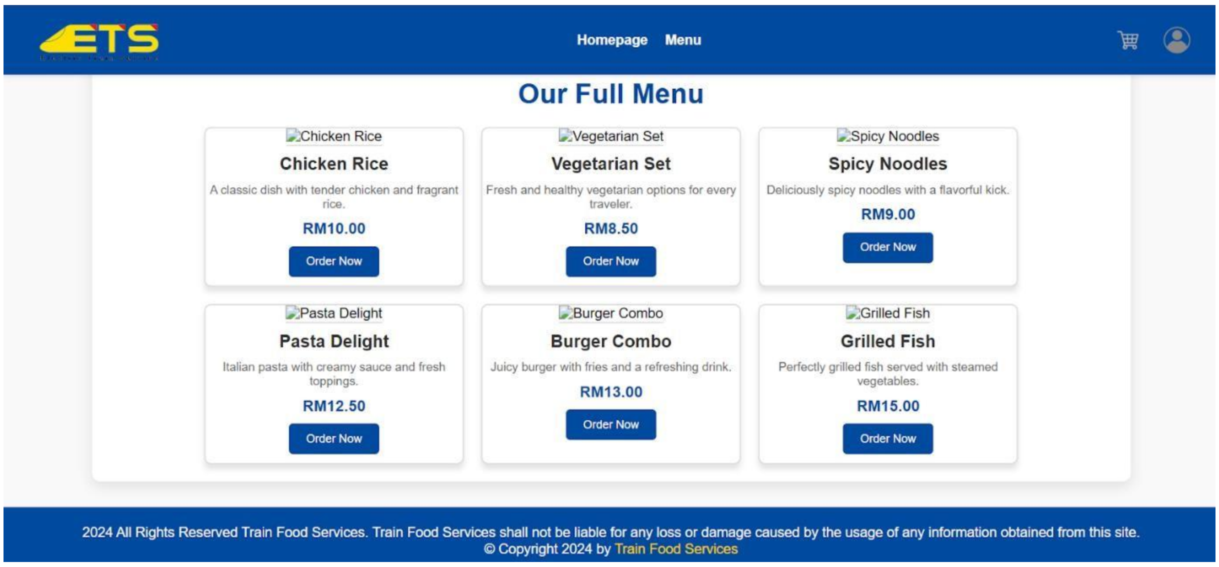

ETS
Homepage
Menu
Our Full Menu
Chicken Rice
Chicken Rice
A classic dish with tender chicken and fragrant rice.
RM10.00
Order Now
Vegetarian Set
Vegetarian Set
Fresh and healthy vegetarian options for every traveler.
RM8.50
Order Now
Spicy Noodles
Spicy Noodles
Deliciously spicy noodles with a flavorful kick.
RM9.00
Order Now
Pasta Delight
Pasta Delight
Italian pasta with creamy sauce and fresh toppings.
RM12.50
Order Now
Burger Combo
Burger Combo
Juicy burger with fries and a refreshing drink.
RM13.00
Order Now
Grilled Fish
Grilled Fish
Perfectly grilled fish served with steamed vegetables.
RM15.00
Order Now
2024 All Rights Reserved Train Food Services. Train Food Services shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2024 by Train Food Services

Figure 3.8: Menu Page

The menu page is organized to make food ordering simple and enjoyable. All food items are clearly displayed with their names, descriptions, appetizing images, and prices. Products are grouped into categories to help customers find what they're looking for quickly. Each item can be easily added to the shopping cart.

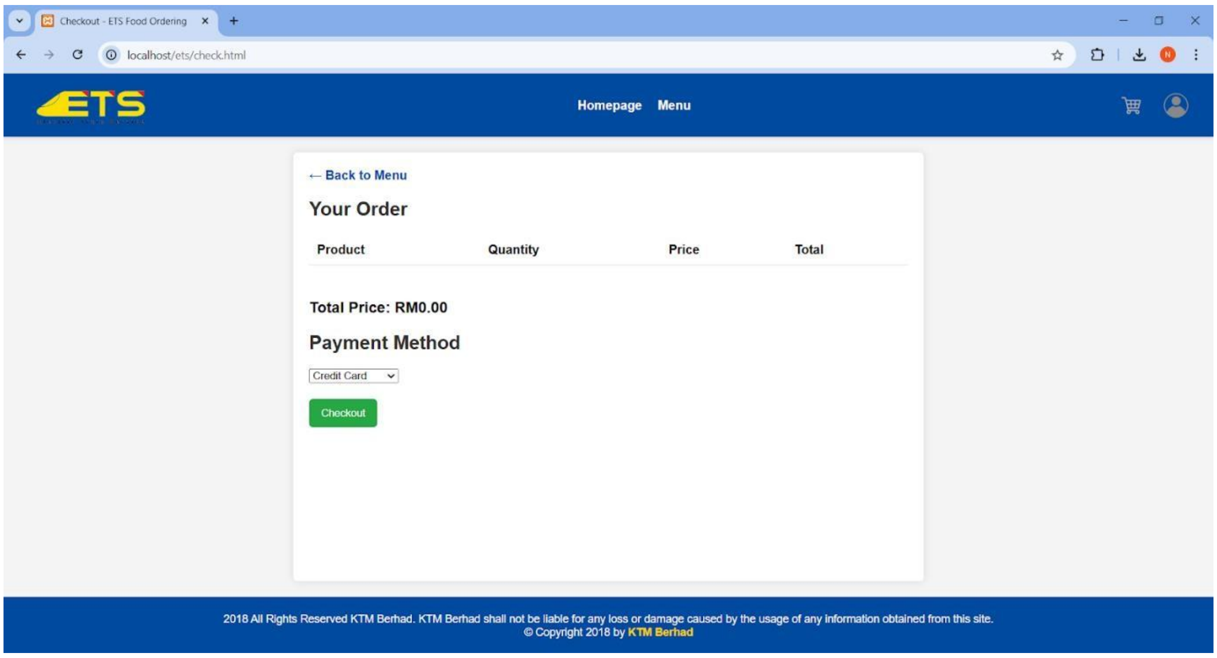
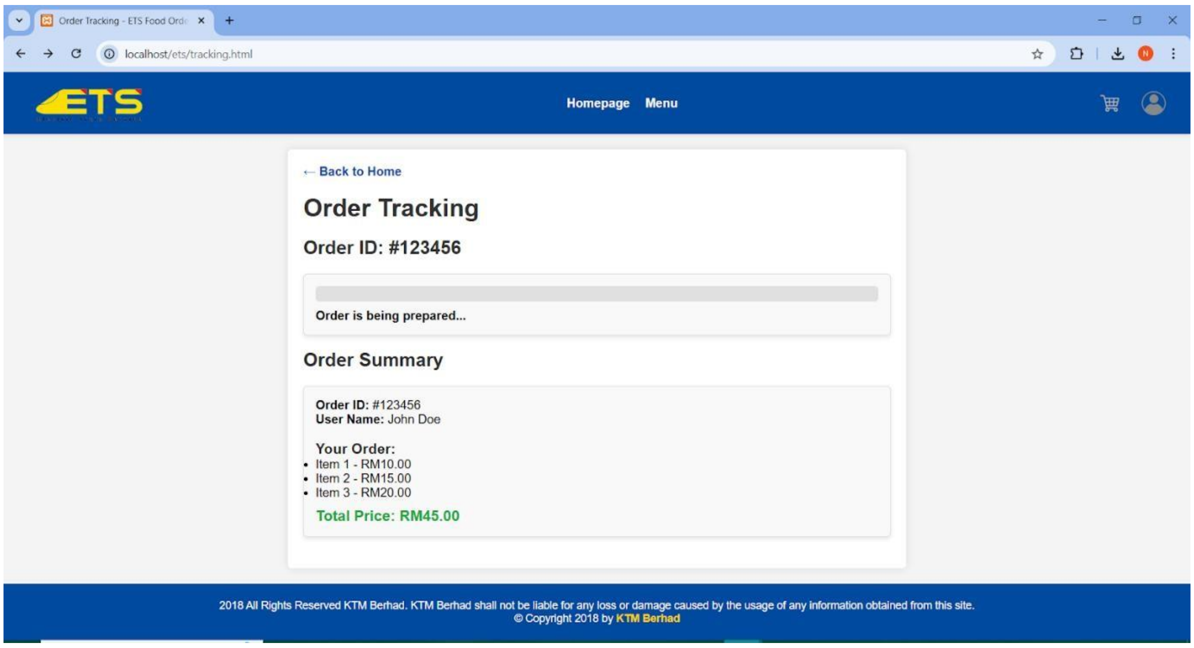

Checkout - ETS Food Ordering
localhost/ets/check.html
ETS
Homepage
Menu
← Back to Menu
Your Order
Product
Quantity
Price
Total
Total Price: RM0.00
Payment Method
Credit Card
Checkout
2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.9: Checkout Page

The checkout page provides a clear summary of the customer's order before payment. It shows a detailed table listing all selected items, their quantities, individual prices, and the total cost. Customers can choose their preferred payment method and enter their delivery address. The page makes it easy to review everything before finalizing the purchase.



Order Tracking - ETS Food Orde
localhost/ets/tracking.html
Homepage Menu

← Back to Home

Order Tracking

Order ID: #123456

Order is being prepared...

Order Summary

Order ID: #123456
User Name: John Doe

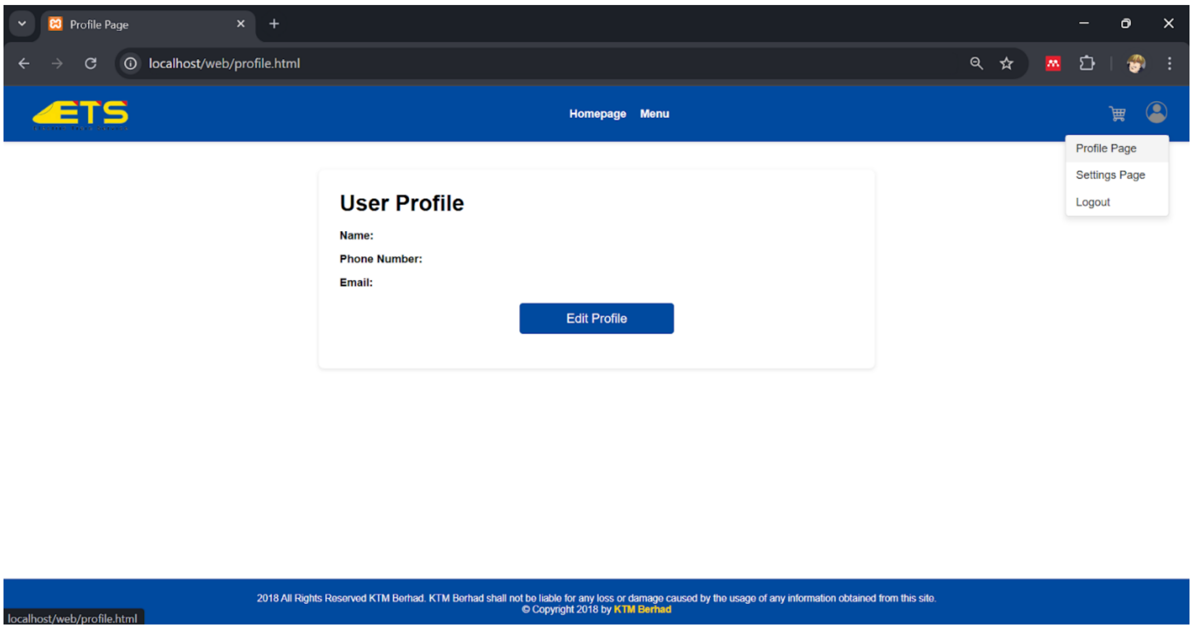
Your Order:
- Item 1 - RM10.00
- Item 2 - RM15.00
- Item 3 - RM20.00

Total Price: RM45.00

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.10: Order Tracking Page

This page helps customers stay informed about their order's progress. It displays essential information including the order ID and a real-time status update. Customers can see a complete order summary and track their food from kitchen preparation to delivery.



Profile Page
localhost/web/profile.html
Homepage Menu

Profile Page
Settings Page
Logout

User Profile

Name:
Phone Number:
Email:

Edit Profile

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

localhost/web/profile.html

Figure 3.11: Profile Page

The profile page gives customers easy access to their personal information. It displays their name, phone number, and email address. Users can update any of this contact information as needed, ensuring their details stay current for smooth order delivery.

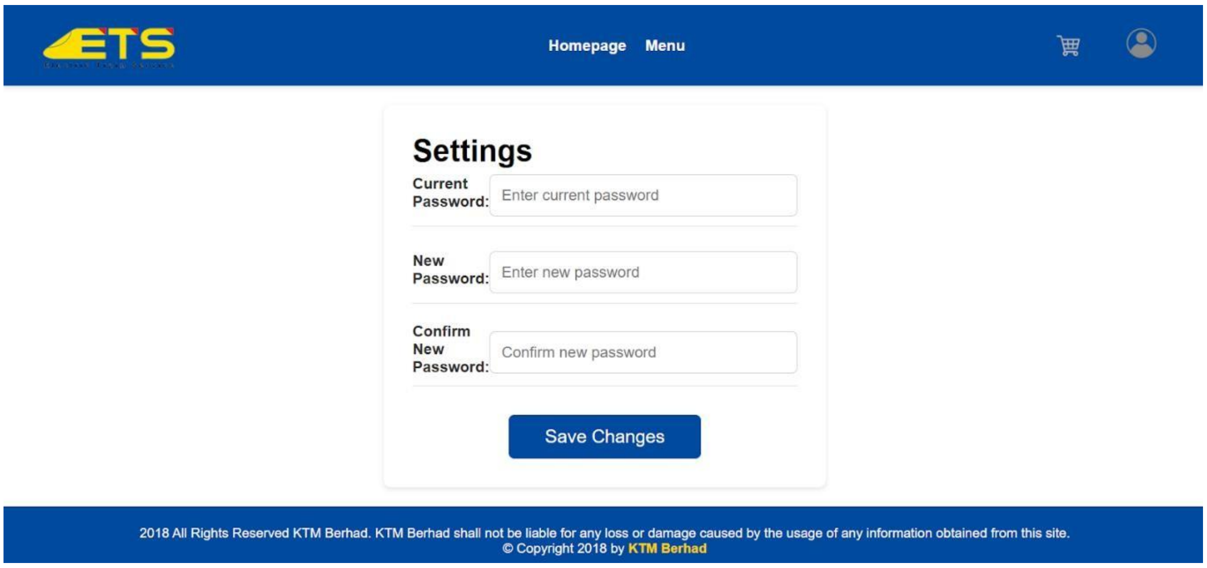
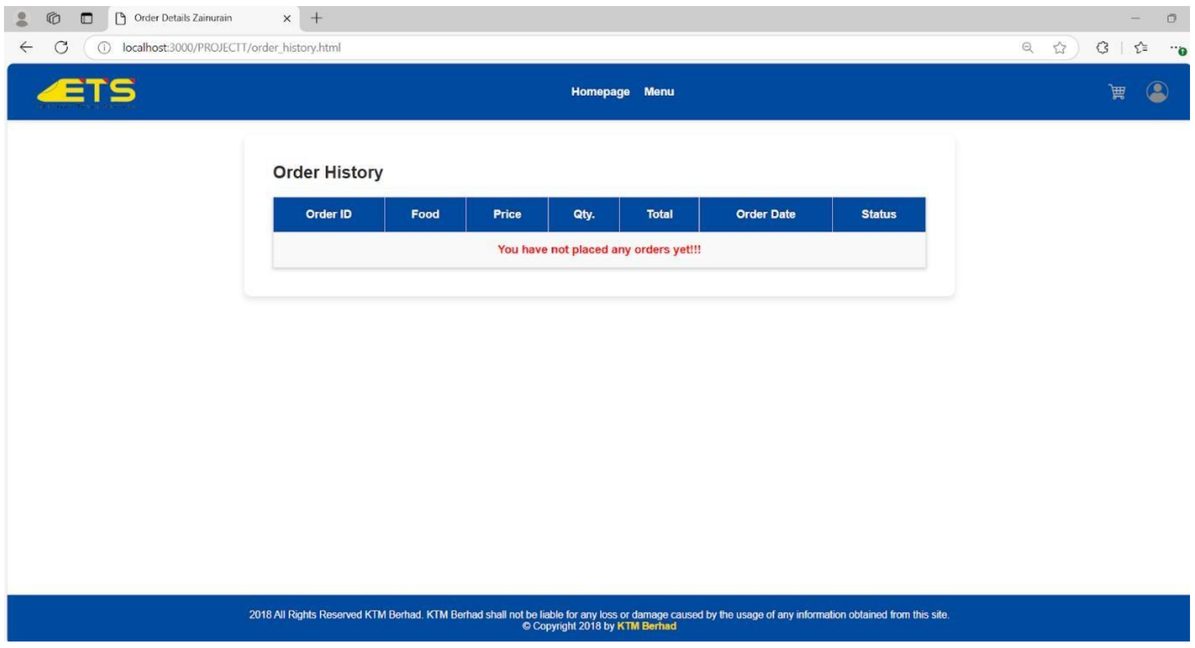

ETS
Homepage
Menu
Settings
Current Password:
Enter current password
New Password:
Enter new password
Confirm New Password:
Confirm new password
Save Changes
2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.12: Setting Page

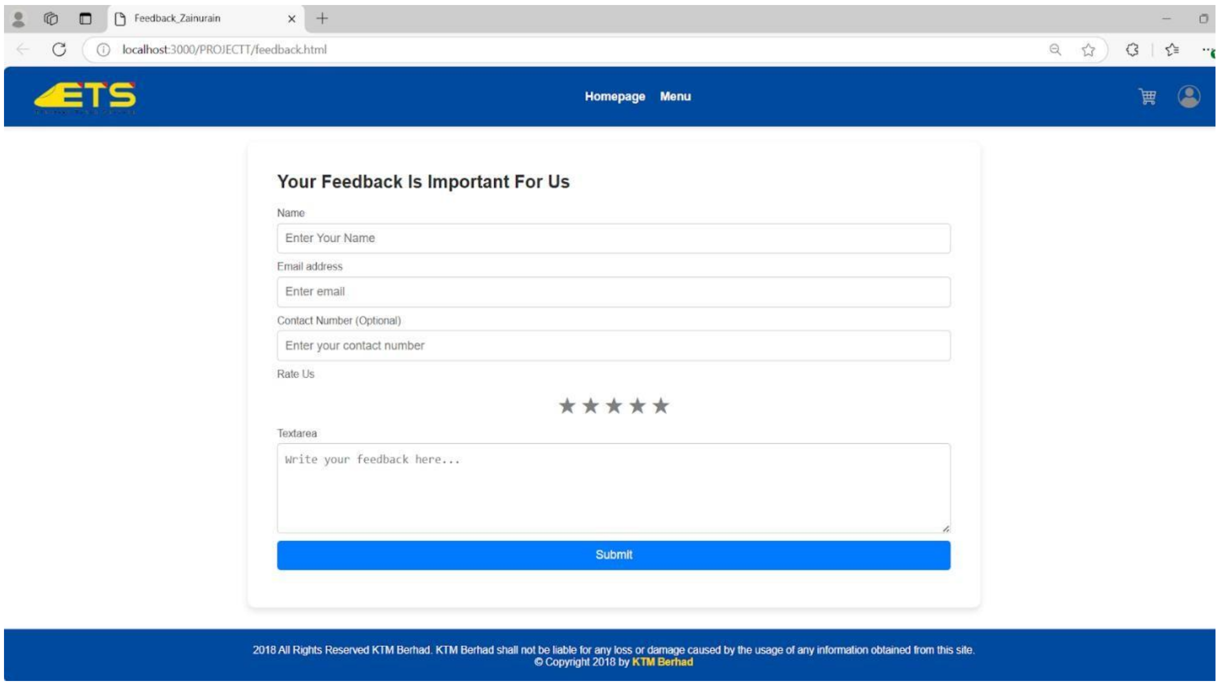
In the settings page, users can manage their account security by changing their password. The page includes fields for entering the current password, choosing a new password, and confirming the new password to prevent typing errors.


Order Details Zainurain
localhost:3000/PROJECT1/order_history.html
ETS
Homepage
Menu
Order History
2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Order ID	Food	Price	Qty.	Total	Order Date	Status
You have not placed any orders yet!!!						

Figure 3.13: Order History Page

The order history page maintains a record of all past orders. It displays a comprehensive table showing important details for each order: the order ID, food items ordered, prices, quantities, total amount, order date, and delivery status. This helps customers track their spending and reorder their favorite items.



Your Feedback Is Important For Us

Name

Enter Your Name

Email address

Enter email

Contact Number (Optional)

Enter your contact number

Rate Us

★★★★★

Textarea

Write your feedback here...

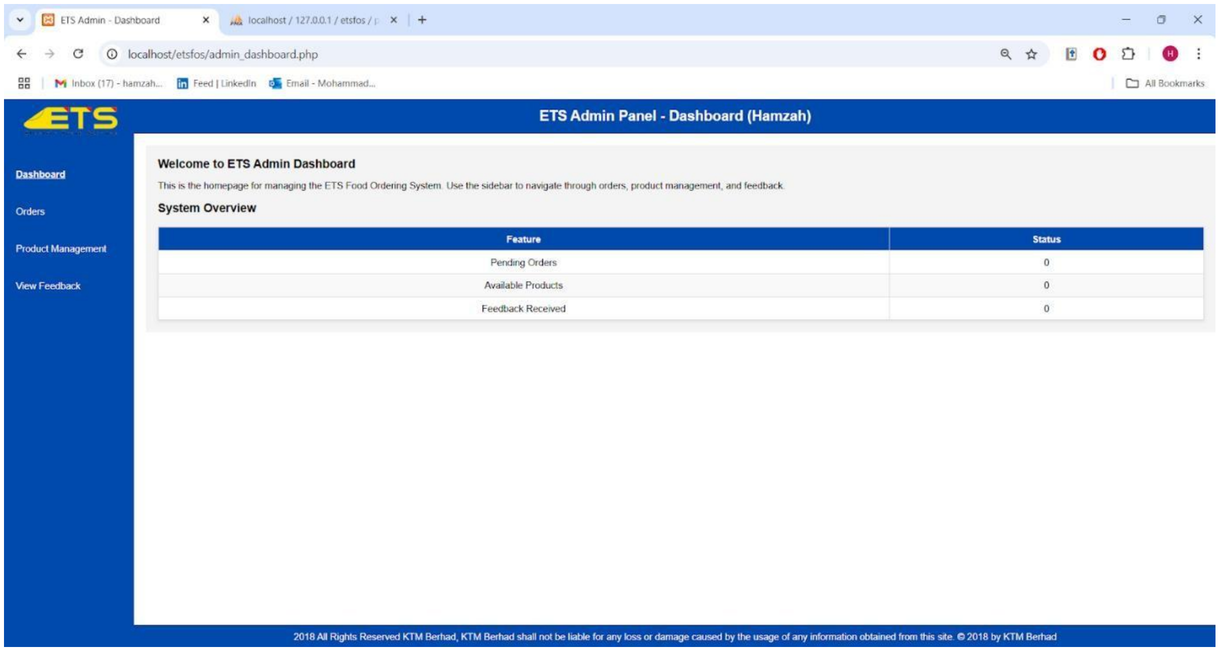
Submit

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.14: Feedback Page

The feedback page encourages customer interaction and service improvement. Users can rate their experience and provide detailed feedback through a text area. This space allows customers to share complaints, suggestions, or request support, helping the platform maintain and improve service quality.

3.4.3 ADMIN INTERFACE



ETS Admin Panel - Dashboard (Hamzah)

Welcome to ETS Admin Dashboard

This is the homepage for managing the ETS Food Ordering System. Use the sidebar to navigate through orders, product management, and feedback.

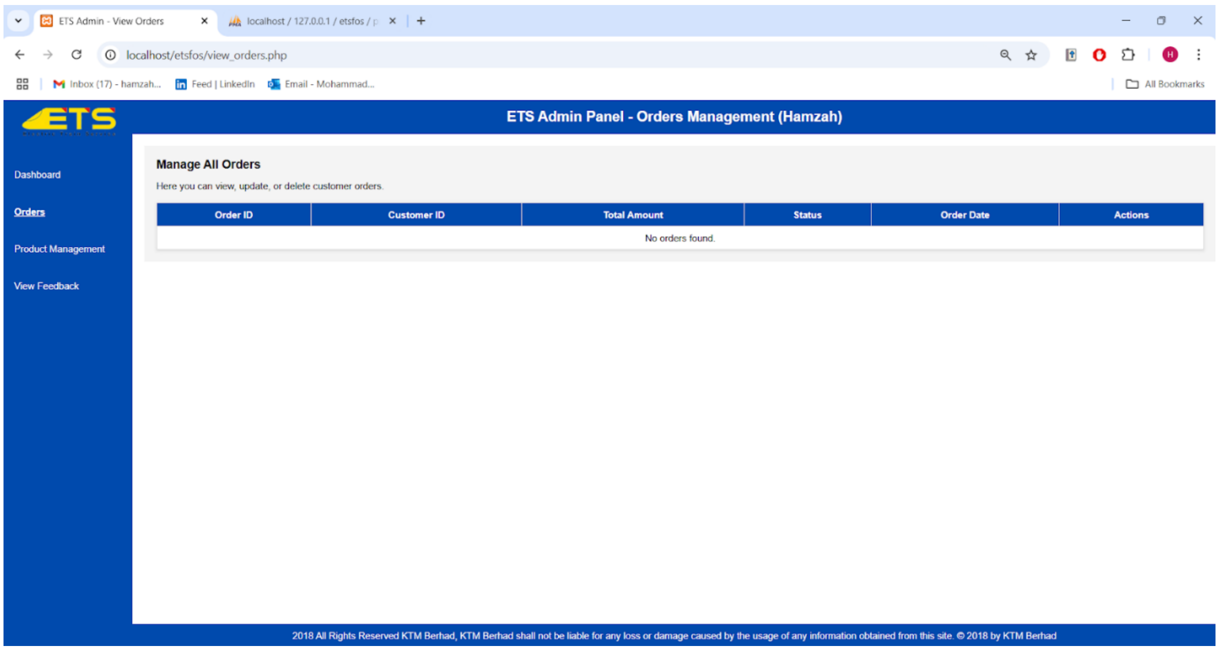
System Overview

Feature	Status
Pending Orders	0
Available Products	0
Feedback Received	0

2018 All Rights Reserved KTM Berhad, KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site. © 2018 by KTM Berhad

Figure 3.15: Dashboard Page

The dashboard serves as the command center for administrators, providing a quick snapshot of everything happening in the system. Staff can see important numbers like how many orders are waiting to be processed, the total number of available food items, and how many customer feedback messages have been received. The page includes easy navigation buttons to access different sections like orders, product management, and customer feedback. The system overview is presented in a clear table format, making it easy to track daily operations.



ETS Admin Panel - Orders Management (Hamzah)

Manage All Orders

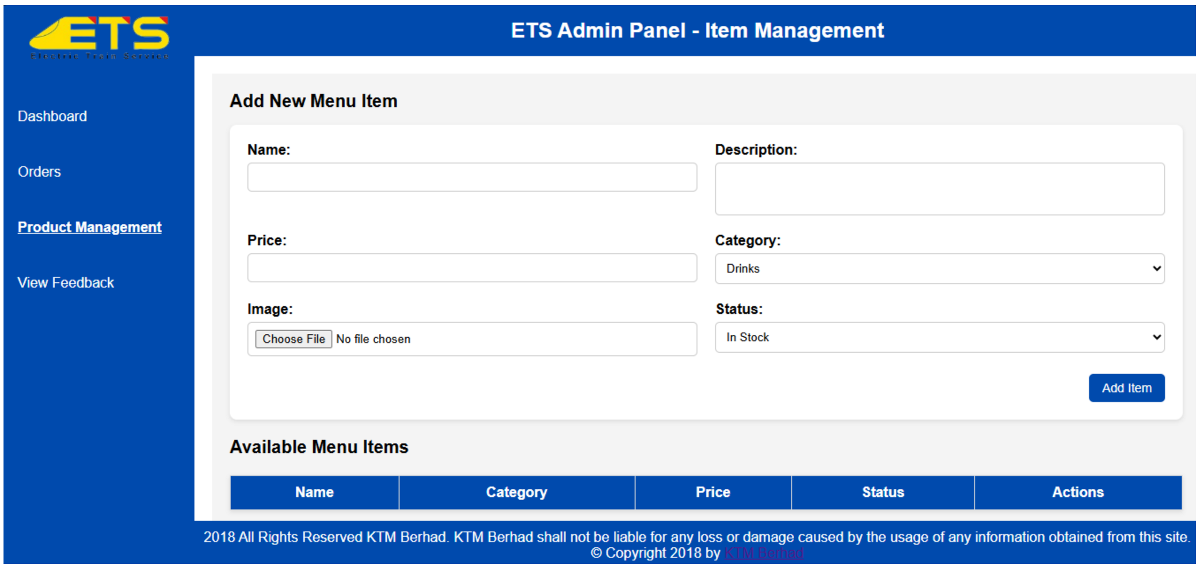
Here you can view, update, or delete customer orders.

Order ID	Customer ID	Total Amount	Status	Order Date	Actions
No orders found.					

2018 All Rights Reserved KTM Berhad, KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site. © 2018 by KTM Berhad

Figure 3.16: View Order Page

The order management page helps staff keep track of all customer orders in one place. It displays a comprehensive table showing essential information for each order, including the order ID, customer ID, total amount, and current status. This organized layout makes it easy for staff to monitor orders and ensure they're being processed efficiently. Staff can quickly see which orders need attention and manage them accordingly.



ETS Admin Panel - Item Management

Add New Menu Item

Name:	Description:
Price:	Category: Drinks
Image: No file chosen	Status: In Stock

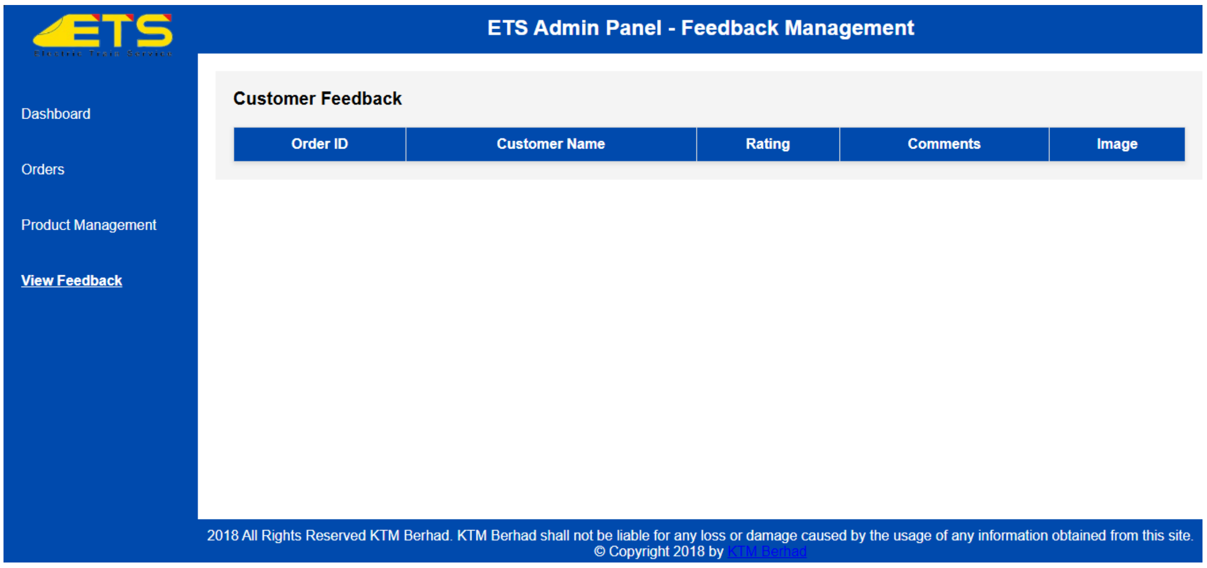
Available Menu Items

Name	Category	Price	Status	Actions

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.17: Item Management Page

This page gives staff complete control over the menu items available to customers. It includes a form where staff can add new items by entering details like the item's name, price, image, description, category, and availability status. Below the form, there's a table showing all current menu items, making it easy to see and manage the entire menu. Each item in the table can be updated or deleted using simple action buttons, helping staff keep the menu current and accurate.


ETS Admin Panel - Feedback Management

Dashboard
Orders
Product Management
View Feedback

Customer Feedback

Order ID	Customer Name	Rating	Comments	Image

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 3.18: View Feedback Page

The feedback page helps staff understand customer experiences and improve services. It displays all customer feedback in an organized table, showing important details like the order ID, customer name, rating, and any comments or images shared by customers. This information helps staff identify areas for improvement and address any customer concerns promptly. The clear layout makes it easy to track and respond to customer feedback effectively.

3.5 WEB-TREE DIAGRAM

The web tree diagram shows how the ETS Food Ordering System is organized, with two main branches: one for regular Users and another for Staff/Admin, each having their own specific features and access paths.

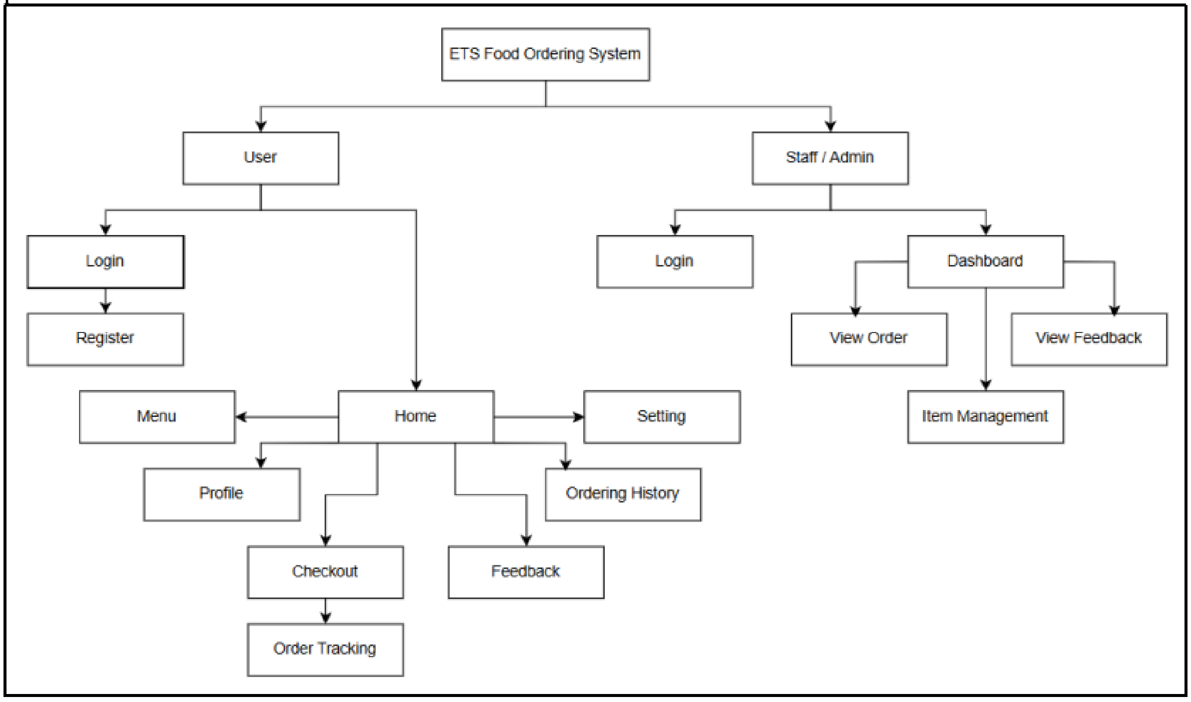


Figure 3.18: Web-Tree Diagram

On the User side, customers start their journey either by logging in to their existing account or registering as a new user. Once logged in, they reach the Home page, which acts as a central hub connecting to various features. From the Home page, users can access the Menu to browse food items, view their Profile for personal information, proceed to Checkout to complete their orders, and track their orders through the Order Tracking feature. Users can also access their Settings to manage account preferences, view their Ordering History to see past purchases, and provide Feedback about their experience.

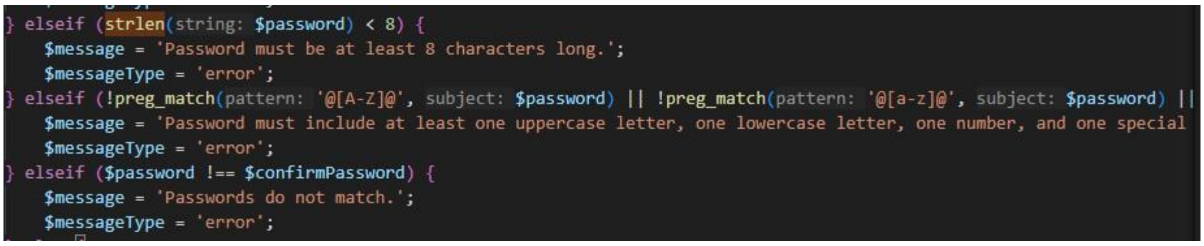
On the Staff/Admin side, the structure is simplified for management purposes. Staff members begin by logging in, which takes them to their Dashboard. From the Dashboard, they can access several key management features: View Order to monitor and process customer orders, Item Management to update the menu and food items, and View Feedback to read and respond to customer comments and suggestions.

This organization ensures that both customers and staff have easy access to the features they need most. The structure is clear and simple, with the Home page serving as the main navigation point for users, while the Dashboard serves the same purpose for staff members. Each branch of the tree is designed to make navigation simple and efficient for its intended users.

CHAPTER 4

IMPLEMENTATION

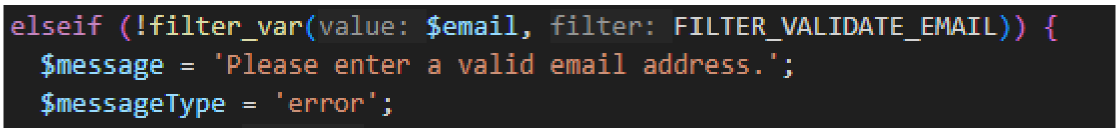
4.1. REGISTER PAGE



```
} elseif (strlen(string: $password) < 8) {
    $message = 'Password must be at least 8 characters long.';
    $messageType = 'error';
} elseif (!preg_match(pattern: '@[A-Z]@', subject: $password) || !preg_match(pattern: '@[a-z]@', subject: $password) ||
    $message = 'Password must include at least one uppercase letter, one lowercase letter, one number, and one special
    $messageType = 'error';
} elseif ($password !== $confirmPassword) {
    $message = 'Passwords do not match.';
    $messageType = 'error';
```

Figure 4.1: Password Policy

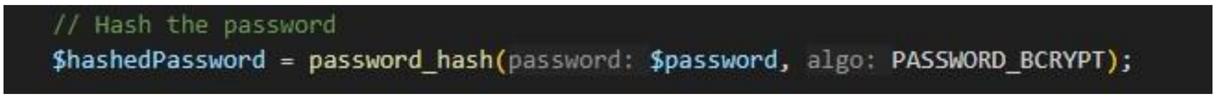
This code implements strict password validation requirements. The strlen($password) function ensures that the password is at least 8 characters long, which meets security criteria for preventing brute-force assaults. The following preg_match() calls check the password's complexity, requiring at least one uppercase letter, one lowercase letter, one digit, and one special character. These factors ensure that passwords are difficult to guess, reducing the risk of using weak or predictable passwords.



```
elseif (!filter_var(value: $email, filter: FILTER_VALIDATE_EMAIL)) {
  $message = 'Please enter a valid email address.';
  $messageType = 'error';
```

Figure 4.2: Email Address Validation

The filter_var() function, together with the FILTER_VALIDATE_EMAIL filter, ensures that users give genuine email addresses. This effectively verifies the format of the email string, rejecting erroneous inputs that could cause problems such as unsuccessful communication or SQL injection vulnerabilities.

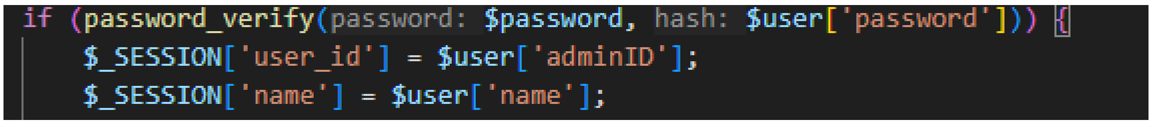


```
// Hash the password
$hashedPassword = password_hash(password: $password, algo: PASSWORD_BCRYPT);
```

Figure 4.3: Hash Password

Once authenticated, the password is securely hashed using the password_hash() function and the PASSWORD_BCRYPT method. Hashing encrypts the password, rendering it inaccessible even if the database is compromised. Using BCRYPT ensures strong encryption, and the produced hash is automatically salted, which improves security by preventing dictionary attacks.

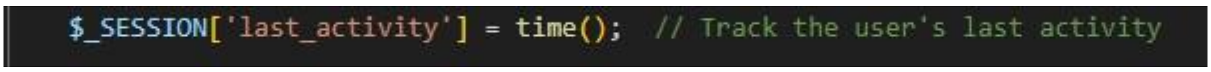
4.2. LOGIN PAGE



```
if (password_verify(password: $password, hash: $user['password'])) {
    $_SESSION['user_id'] = $user['adminID'];
    $_SESSION['name'] = $user['name'];
```

Figure 4.4: Login User Credential Validation

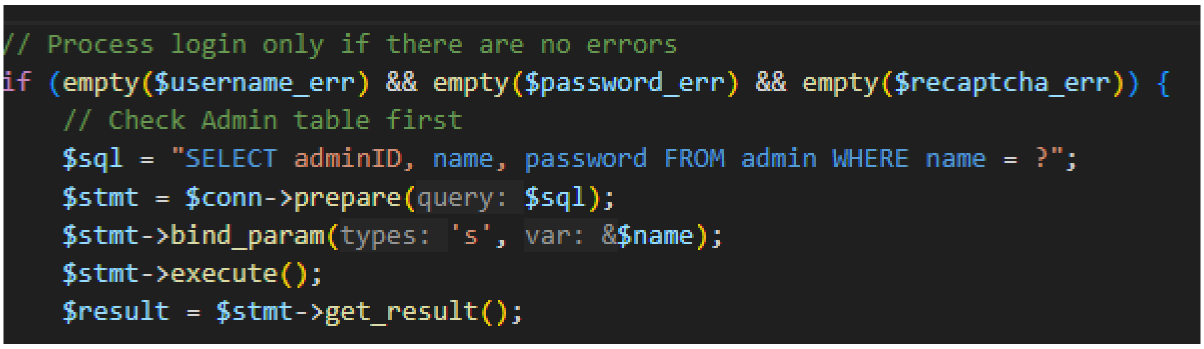
This function validates a user's login credentials by comparing the submitted password to the hashed password stored in the database. If the password is correct, a new session is started (or a current one is extended), and user-specific data is kept in it.
The password_verify() function secures password authentication, lowering the danger of disclosing sensitive user data. When the verification is successful, the following session variables are set. $_SESSION['user_id']: Saves the user's unique identification, such as adminID, for future use in session-based access control. $_SESSION['name']: Stores the user's name, which can be used to show information or personalise user experiences. This method guarantees that the user remains logged in during their session, allowing for seamless navigation within the application's restricted regions.



```
    $_SESSION['last_activity'] = time();  // Track the user's last activity
```

Figure 4.5: Timestamp of Activity

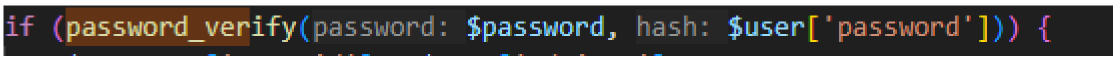
Every time the session data is reviewed,this timestamp confirms that the session is still active. If this timestamp exceeds a specified session limit (for example, 30 minutes).



```
// Process login only if there are no errors
if (empty($username_err) && empty($password_err) && empty($recaptcha_err)) {
    // Check Admin table first
    $sql = "SELECT adminID, name, password FROM admin WHERE name = ?";
    $stmt = $conn->prepare(query: $sql);
    $stmt->bind_param(types: 's', var: &$name);
    $stmt->execute();
    $result = $stmt->get_result();
```

Figure 4.6: Input Sanitization

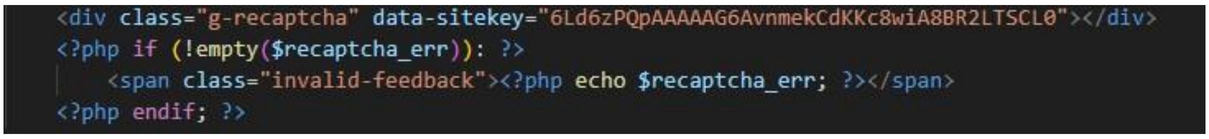
To prevent SQL Injection attacks, we employ prepared statements that bind parameters rather than directly adding user input into the query. This is a highly effective defence because it isolates the SQL code and the user data.The input ($name here) is sanitised with bind_param, which means that any special characters in the user's input are regarded as data rather than executable code.



```
if (password_verify(password: $password, hash: $user['password'])) {
```

Figure 4.7: Password Hashing

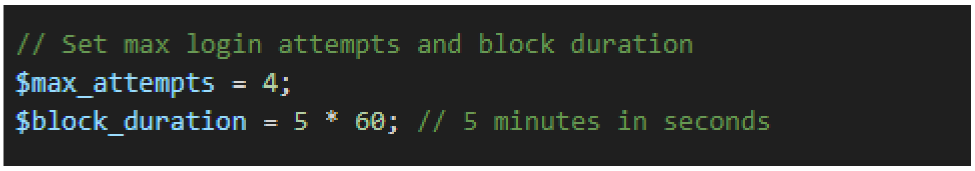
This approach ensures that passwords are safely kept in the database. We do not keep the password in plain text, but rather in a hashed format using PHP's password_hash function. When users log in, we compare the hashed password to their input.



```
<div class="g-recaptcha" data-sitekey="6Ld6zPQpAAAAAG6AvnmekCdKKc8wiA8BR2LTSCL0"></div>
<?php if (!empty($recaptcha_err)): ?>
    <span class="invalid-feedback"><?php echo $recaptcha_err; ?></span>
<?php endif; ?>
```

Figure 4.8: Google Captcha Form

This code loads the reCAPTCHA widget into the form. The data-sitekey parameter associates the reCAPTCHA widget with your Google account. The form includes the user's response (whether they properly identified photos or completed a task).

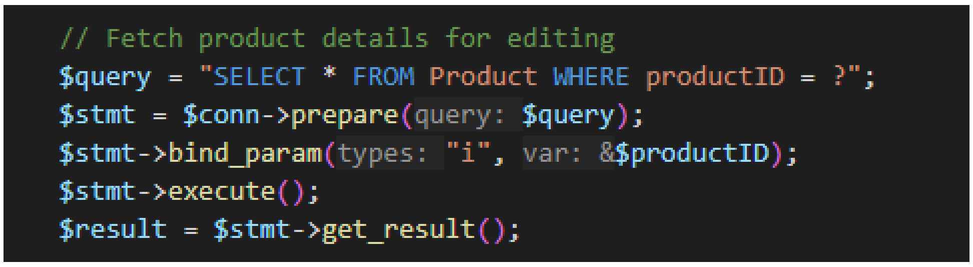


```
// Set max login attempts and block duration
$max_attempts = 4;
$block_duration = 5 * 60; // 5 minutes in seconds
```

Figure 4.9: Limit Login Attempts

By implementing a block time mechanism into the login process, we safeguard users from attackers who may guess usernames or passwords. If an attacker continues to enter wrong data, we temporarily stop them.If a user exceeds the maximum number of attempts (determined by $max_attempts), the system saves the block duration in the session ($_SESSION['blocked_until']). After this time has passed, login is permitted again.This significantly minimises the probability of brute-force login attempts items, or delete items they no longer want to purchase.

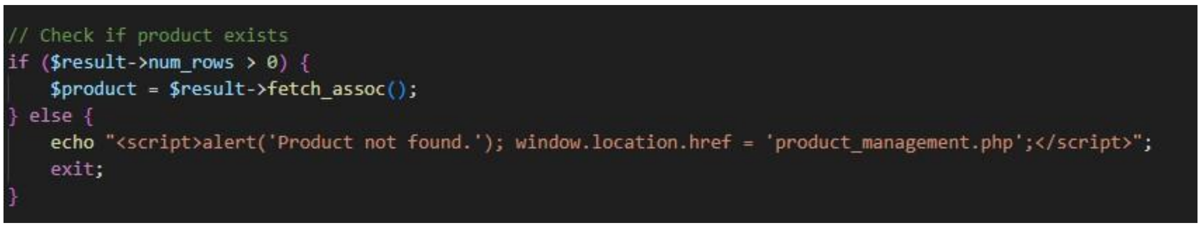
4.3. PRODUCT MANAGEMENT



```
// Fetch product details for editing
$query = "SELECT * FROM Product WHERE productID = ?";
$stmt = $conn->prepare(query: $query);
$stmt->bind_param(types: "i", var: &$productID);
$stmt->execute();
$result = $stmt->get_result();
```

Figure 4.10: Fetch Product Details

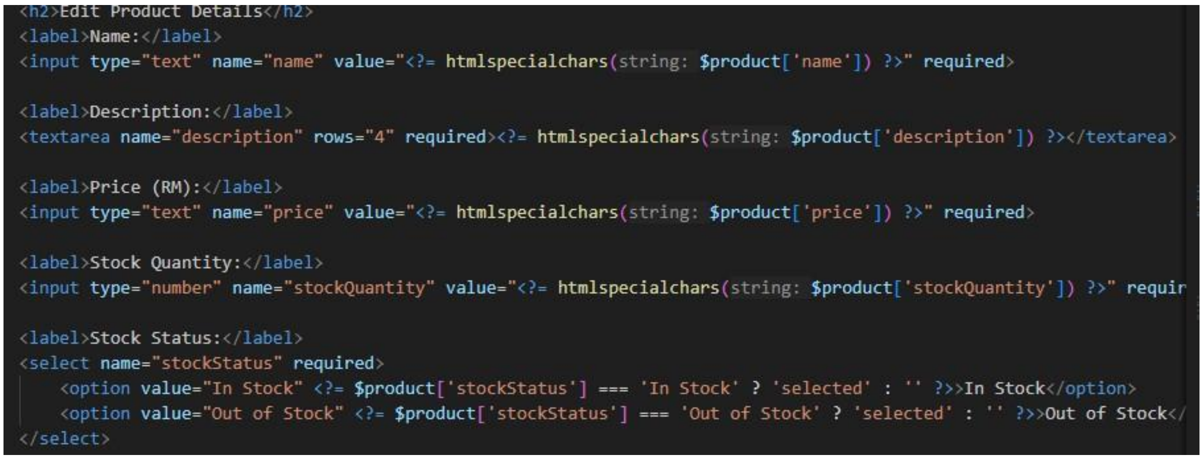
The statements facilitate the safe and efficient execution of queries. The placeholder (?) is replaced by the bound value ($productID), thus even if an attacker attempts to inject malicious SQL code into the productID, it will not be executed.Binding the parameter (bind_param) enables secure input handling by specifying the expected type of the parameter and ensuring that only integers are permitted.The execute() method executes the query, while get_result() fetches the corresponding records from the database.This procedure improves security to prevent SQL injection, efficiency by reusing the same prepared query, and readability by making the query dynamic and adaptable.



```
// Check if product exists
if ($result->num_rows > 0) {
    $product = $result->fetch_assoc();
} else {
    echo "<script>alert('Product not found.'); window.location.href = 'product_management.php';</script>";
    exit;
}
```

Figure 4.11: Checking Product Existence

This part of code determines whether a product exists in the database by seeing if any rows were returned after the query was executed. If the query returns one or more rows (showing that a product with the 'productID' exists), the 'fetch_assoc()' method retrieves the product's details as an associative array and stores the data in the '$product' variable. However, if no matching product is discovered, the code displays a JavaScript alert with the message "Product not found" and sends the user back to the 'product_management.php' page to avoid further processing or inappropriate actions. The 'exit' statement ensures that no extra code is performed following the redirection, so preserving the flow and preventing problems.

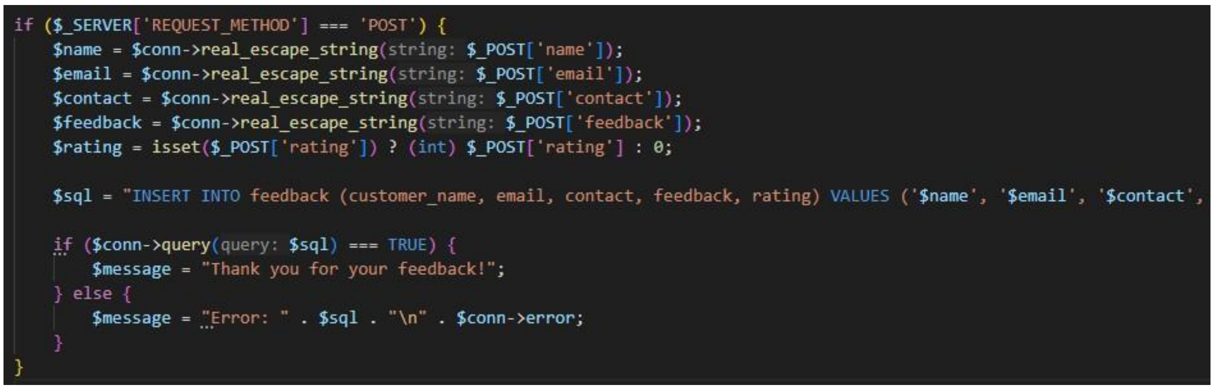


```
<h2>Edit Product Details</h2>
<label>Name:</label>
<input type="text" name="name" value="<?= htmlspecialchars(string: $product['name']) ?>" required>

<label>Description:</label>
<textarea name="description" rows="4" required><?= htmlspecialchars(string: $product['description']) ?></textarea>

<label>Price (RM):</label>
<input type="text" name="price" value="<?= htmlspecialchars(string: $product['price']) ?>" required>

<label>Stock Quantity:</label>
<input type="number" name="stockQuantity" value="<?= htmlspecialchars(string: $product['stockQuantity']) ?>" requir

<label>Stock Status:</label>
<select name="stockStatus" required>
    <option value="In Stock" <?= $product['stockStatus'] === 'In Stock' ? 'selected' : '' ?>>In Stock</option>
    <option value="Out of Stock" <?= $product['stockStatus'] === 'Out of Stock' ? 'selected' : '' ?>>Out of Stock</
</select>
```

Figure 4.12: Product Editing Form

This piece of the code displays the product editing form, which allows the administrator to amend the product information. The 'h2' tag indicates the form's header, which reads "Edit Product Details." Each product attribute, including name, description, price, stock quantity, and stock status, is presented within the appropriate HTML form components. For example, the product's name is displayed in a text input form ('input type="text">'), pre-filled with the current product's name value (escaped using 'htmlspecialchars' to prevent XSS attacks). Similarly, the description appears in a textarea field, while the price and stock quantity fields accept text and number inputs, respectively. The stock state is handled via a select dropdown, which selects the appropriate choice based on the product's current stock status. The'mandatory' element ensures that the user cannot submit the form without entering the needed values. This form structure allows the administrator to change and amend product information immediately.

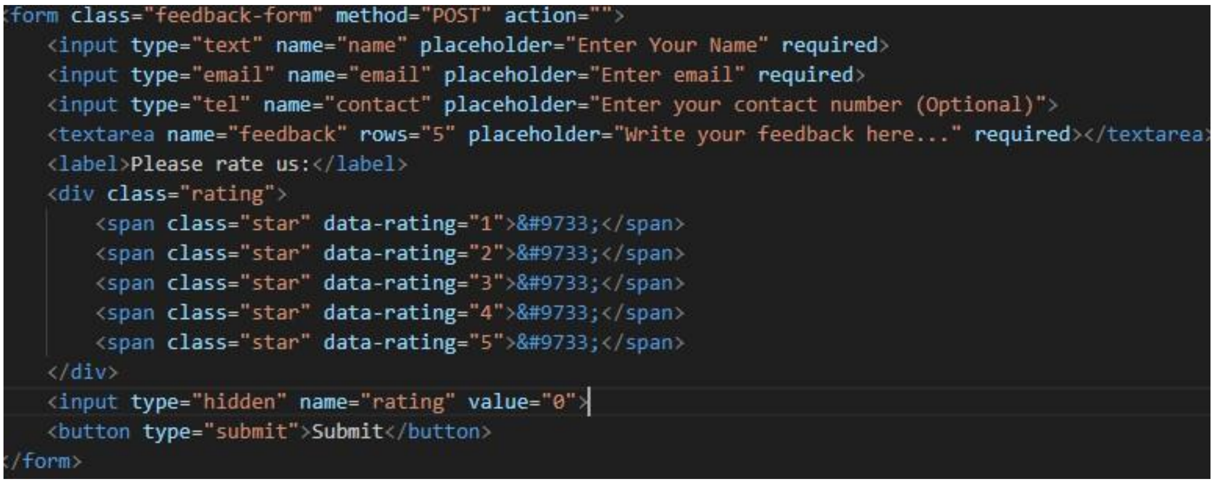
4.4. FEEDBACK PAGE



```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $name = $conn->real_escape_string(string: $_POST['name']);
    $email = $conn->real_escape_string(string: $_POST['email']);
    $contact = $conn->real_escape_string(string: $_POST['contact']);
    $feedback = $conn->real_escape_string(string: $_POST['feedback']);
    $rating = isset($_POST['rating']) ? (int) $_POST['rating'] : 0;

    $sql = "INSERT INTO feedback (customer_name, email, contact, feedback, rating) VALUES ('$name', '$email', '$contact',

    if ($conn->query(query: $sql) === TRUE) {
        $message = "Thank you for your feedback!";
    } else {
        $message = "Error: " . $sql . "\n" . $conn->error;
    }
}
```

Figure 4.13:PHP Script for Feedback Form Handling and Database Insertion

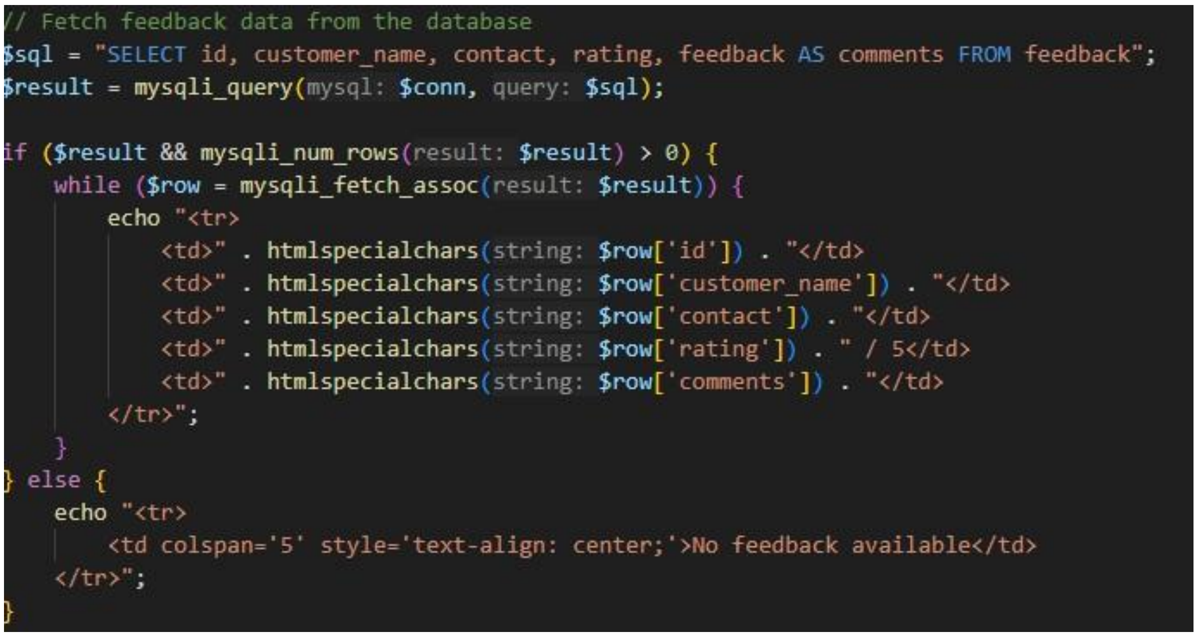
The image above displays a PHP script that handles a form submission when the request method is "POST." It extracts data entered the form, such as the customer's name, email, phone number, feedback, and rating. To prevent SQL injection attacks, the input is sanitized with real_escape_string() before being inserted into the database table named feedback. The script then creates a SQL INSERT query to save the sanitized data in the database. If the query completes successfully, a success message is presented; otherwise, an error message with the failed SQL query and error details is displayed.



```
<form class="feedback-form" method="POST" action="">
    <input type="text" name="name" placeholder="Enter Your Name" required>
    <input type="email" name="email" placeholder="Enter email" required>
    <input type="tel" name="contact" placeholder="Enter your contact number (Optional)">
    <textarea name="feedback" rows="5" placeholder="Write your feedback here..." required></textarea
    <label>Please rate us:</label>
    <div class="rating">
        <span class="star" data-rating="1">&#9733;</span>
        <span class="star" data-rating="2">&#9733;</span>
        <span class="star" data-rating="3">&#9733;</span>
        <span class="star" data-rating="4">&#9733;</span>
        <span class="star" data-rating="5">&#9733;</span>
    </div>
    <input type="hidden" name="rating" value="0">
    <button type="submit">Submit</button>
</form>
```

Figure 4.14: Feedback Form

The second part of the code displays an HTML form layout meant to collect user feedback. It has input areas for the user's name, email address, phone number (optional), and a textarea for comments. There's also a rating system with hidden input and clickable star icons. Unicode characters are used to represent the stars, which are then allocated data attributes with rating values ranging from 1 to 5. When a user clicks on a star, the rating value is kept in a hidden input field and submitted along with the form. Finally, the form includes a "Submit" button, which launches the PHP script. The design includes a straightforward rating mechanism and a simple interface for users to offer input.



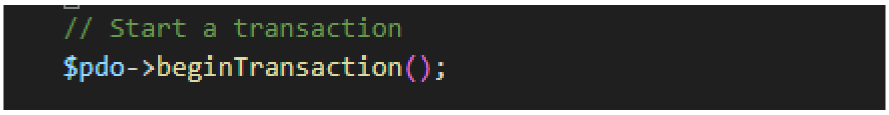
```
// Fetch feedback data from the database
$sql = "SELECT id, customer_name, contact, rating, feedback AS comments FROM feedback";
$result = mysqli_query(mysql: $conn, query: $sql);

if ($result && mysqli_num_rows(result: $result) > 0) {
    while ($row = mysqli_fetch_assoc(result: $result)) {
        echo "<tr>
            <td>" . htmlspecialchars(string: $row['id']) . "</td>
            <td>" . htmlspecialchars(string: $row['customer_name']) . "</td>
            <td>" . htmlspecialchars(string: $row['contact']) . "</td>
            <td>" . htmlspecialchars(string: $row['rating']) . " / 5</td>
            <td>" . htmlspecialchars(string: $row['comments']) . "</td>
        </tr>";
    }
} else {
    echo "<tr>
        <td colspan='5' style='text-align: center;'>No feedback available</td>
    </tr>";
}
```

Figure 4.15: Input Sanitization

The function'mysqli_num_rows($result)' determines the number of rows in the query result, ensuring that there is data to display. Within the 'if' statement, a 'while' loop iterates through the fetched rows. 'mysqli_fetch_assoc($result)' returns each row as an associative array. The loop processes each row individually, dynamically constructing an HTML table row ('') for each feedback entry.

Before displaying the data, the 'htmlspecialchars' function sanitizes it by transforming special characters to HTML-safe equivalents. This is a crucial security feature for preventing Cross-Site Scripting (XSS) attacks and ensuring that any data presented on the webpage does not execute as malicious code. If there are no rows to display, an alternative message appears, telling the admin that no feedback is available. This offers a consistent user experience even when the database lacks feedback entries.

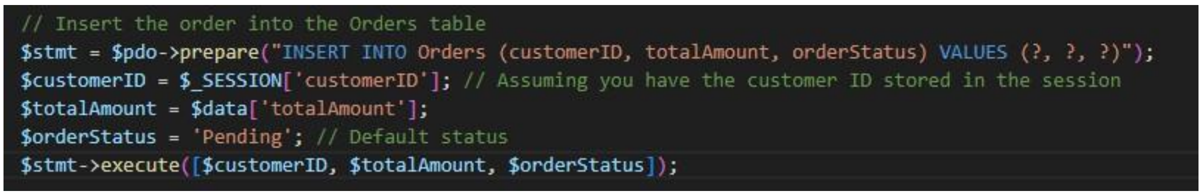
4.5. CHECKOUT AND ORDER TRACKING PAGE



```
// Start a transaction
$pdo->beginTransaction();
```

Figure 4.16: Transaction payment

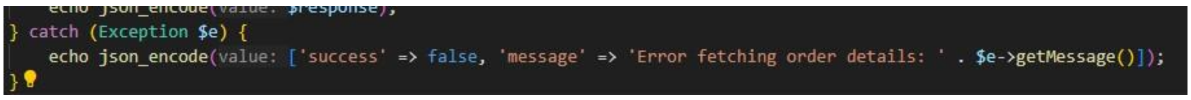
This shows how transactions protect data. Transactions verify order, item, and payment details are inserted correctly. To avoid saving incomplete or incorrect data, the transaction rollback if a database constraint is violated.



```
// Insert the order into the Orders table
$stmt = $pdo->prepare("INSERT INTO Orders (customerID, totalAmount, orderStatus) VALUES (?, ?, ?)");
$customerID = $_SESSION['customerID']; // Assuming you have the customer ID stored in the session
$totalAmount = $data['totalAmount'];
$orderStatus = 'Pending'; // Default status
$stmt->execute([$customerID, $totalAmount, $orderStatus]);
```

Figure 4.17: SQL queries

The statements above ensure that user-provided data is safely tied to SQL queries. This strategy reduces the possibility of SQL injection attacks, which are a common weakness in online applications.



```
    echo json_encode(value: $response);
} catch (Exception $e) {
    echo json_encode(value: ['success' => false, 'message' => 'Error fetching order details: ' . $e->getMessage()]);
}
```

Figure 4.18: Display Error Fetching Details

The code maintains robustness with a try-catch block, which manages possible exceptions during data retrieval. If an error occurs, such as a database connectivity failure or an invalid action, it is gracefully handled, and a generic but informative message is returned to the client. This strategy protects important system details from exposure, improving the application's overall security and stability.

CHAPTER 5

RESULTS AND DISCUSSION

5.
5.1. USER MANUAL

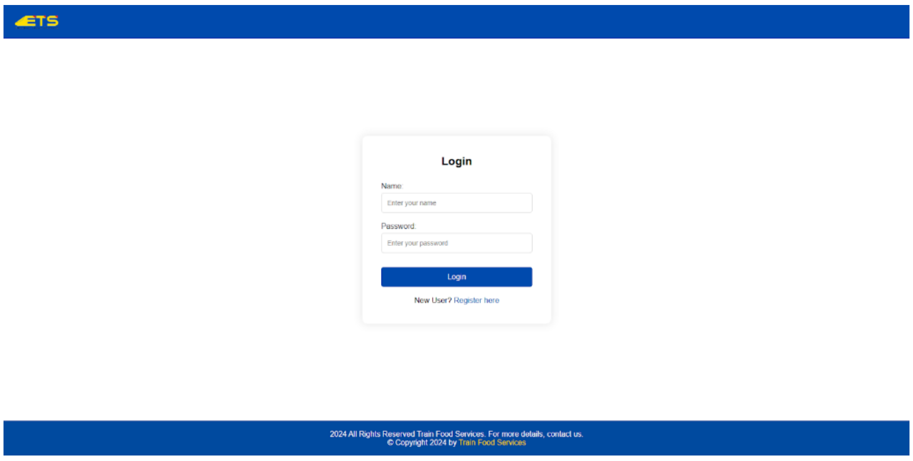
Customer Manual

<u>Booking Order</u>

Step 1: To access the system, the customer must log in by entering their registered email and password. If the customer is new, they need to register by providing the following details:

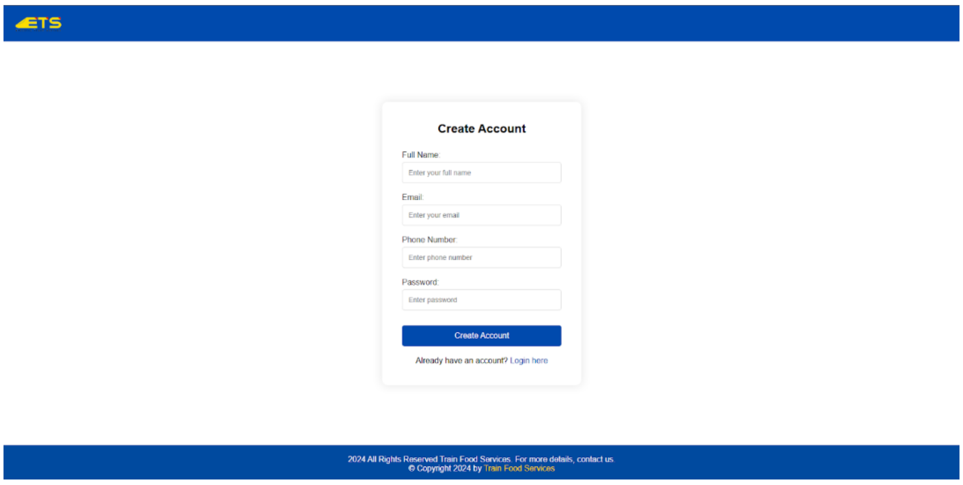
- Name
- Phone number
- Email
- Password

This registration process creates a user account for future access.



2024 All Rights Reserved Train Food Services. For more details, contact us.
© Copyright 2024 by Train Food Services

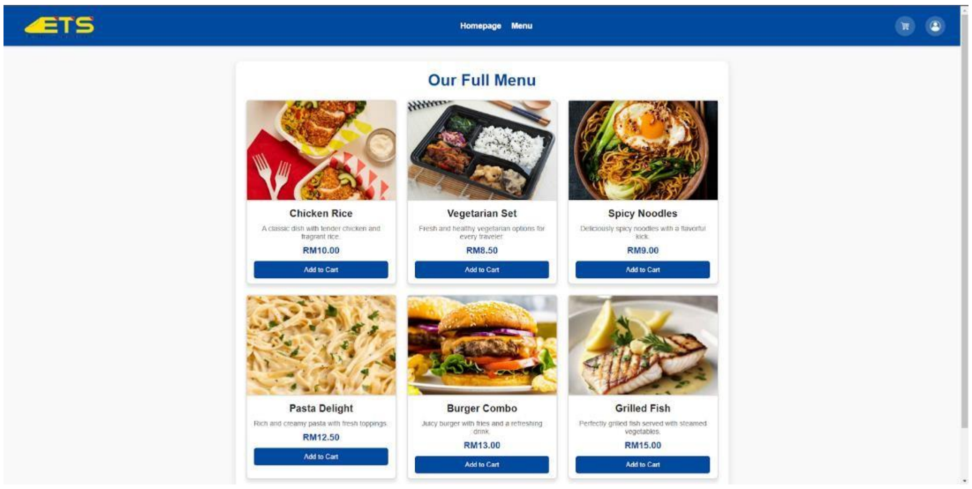
Figure 5.1: Login Page



2024 All Rights Reserved Train Food Services. For more details, contact us.
© Copyright 2024 by Train Food Services

Figure 5.2: Register Page

Step 2: After logging in, customers can browse the menu. By clicking on the Add to Cart button, they can select items they wish to order.



Chicken Rice
A classic dish with tender chicken and fragrant rice.
RM10.00

Vegetarian Set
Fresh and healthy vegetarian options for every traveler.
RM8.50

Spicy Noodles
Deliciously spicy noodles with a flavorful kick.
RM9.00

Pasta Delight
Rich and creamy pasta with fresh toppings.
RM12.50

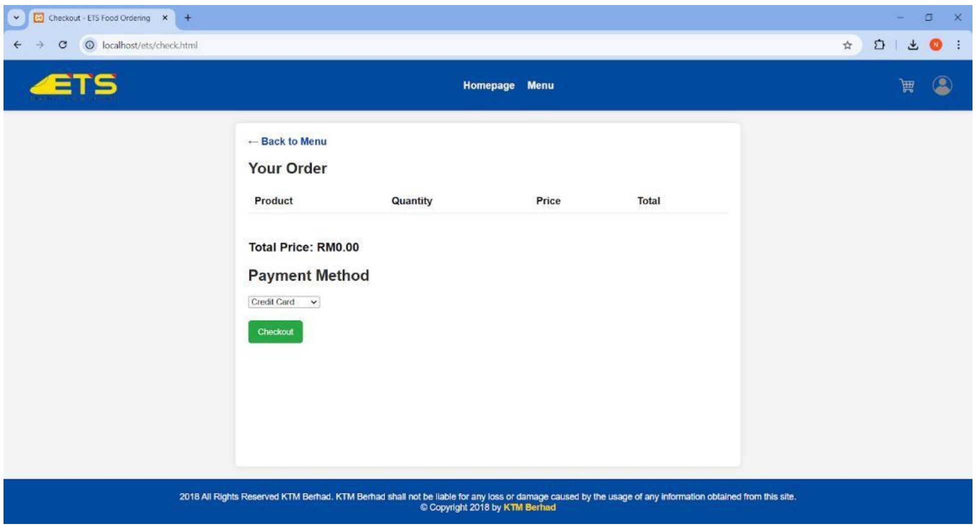
Burger Combo
Juicy burger with fries and a refreshing drink.
RM13.00

Grilled Fish
Perfectly grilled fish served with steamed vegetables.
RM15.00

Figure 5.3: Book Order Page

Step 3: Once the desired menu items are added to the cart, customers can proceed to the checkout process. They need to:

1. Select a preferred payment method.
2. Click on the **Checkout** button to confirm the order.



← Back to Menu

Your Order

Product	Quantity	Price	Total

Total Price: RM0.00

Payment Method

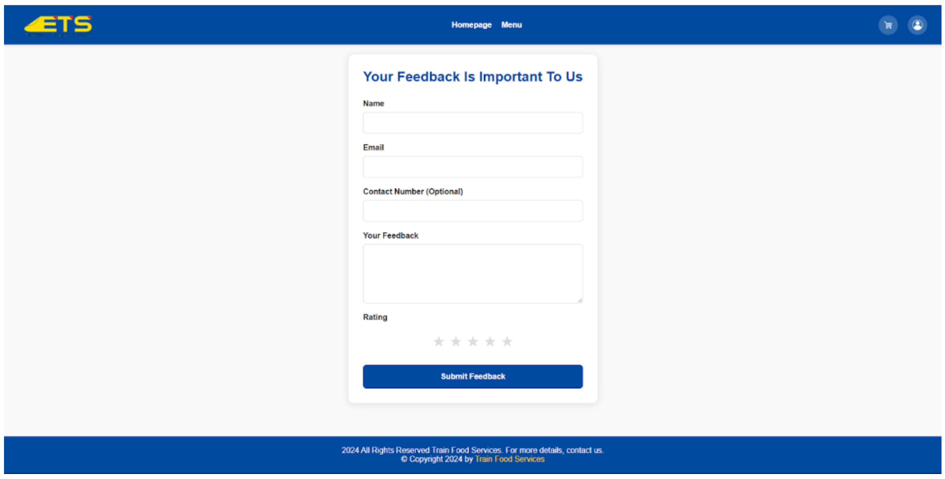
Credit Card

Checkout

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 5.4: Checkout Page

Step 4: After receiving the food, customers are encouraged to share their experience by filling out a feedback form.



Your Feedback Is Important To Us

Name

Email

Contact Number (Optional)

Your Feedback

Rating

★ ★ ★ ★ ★

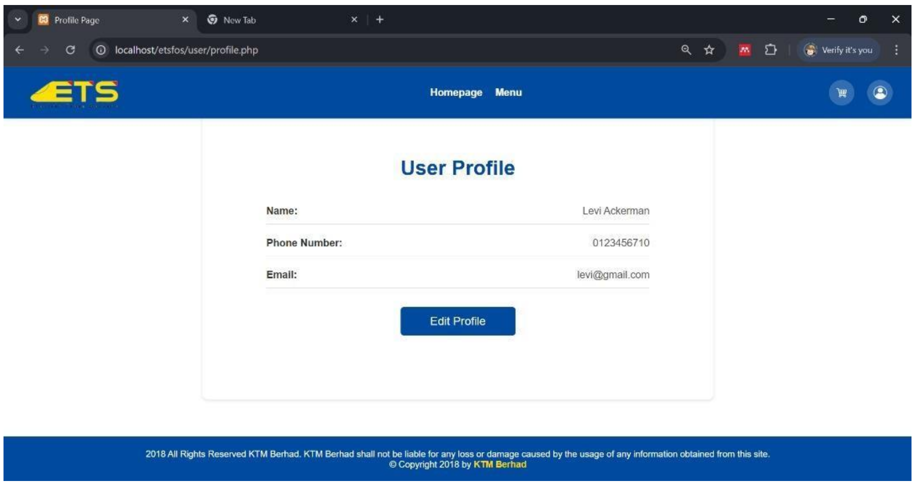
Submit Feedback

2024 All Rights Reserved Train Food Services. For more details, contact us.
© Copyright 2024 by Train Food Services

Figure 5.5: Feedback Page for User

<u>Manage Profile</u>

Step 1: To change or update the user profile, the user can navigate to the page by clicking on the user profile and choose to edit the profile.



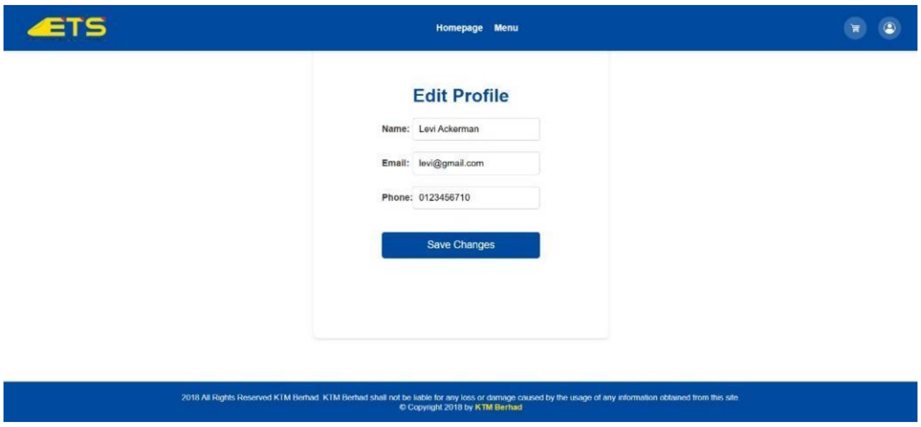
User Profile

Name:	Levi Ackerman
Phone Number:	0123456710
Email:	levi@gmail.com

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 5.6: Profile Page for user

Step 2: Fill in the new information that user wants to fill in, then click save changes button to save the updated information.



Edit Profile

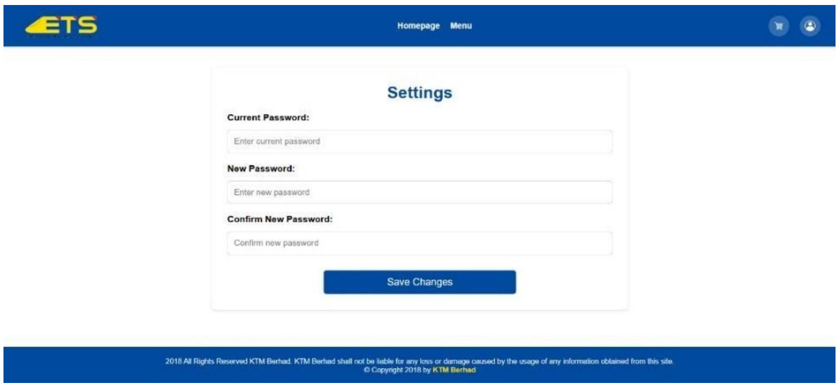
Name:	Levi Ackerman
Email:	levi@gmail.com
Phone:	0123456710

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

Figure 5.7: Profile Page for user

<u>Manage New Password</u>

Step 1: The user can navigate the settings feature on the user profile. By navigating that feature, s can update their new password. By clicking on the save changes button, the new password will be saved to the database.



Settings

Current Password:	Enter current password
New Password:	Enter new password
Confirm New Password:	Confirm new password

2018 All Rights Reserved KTM Berhad. KTM Berhad shall not be liable for any loss or damage caused by the usage of any information obtained from this site.
© Copyright 2018 by KTM Berhad

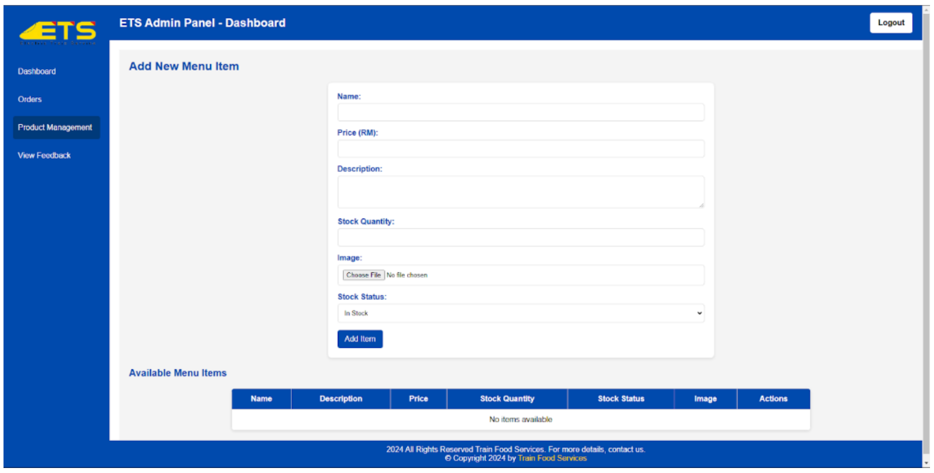
Figure 5.8: Update Password Page for User

Admin Manual

<u>Manage Product</u>

Step 1: The user can log in to the system by entering their name and password.

Step 2: After logging in, admins can navigate to the Product Management section. Here, they can fill in the necessary information to add a new product to the menu and click the add item button to save the menu .



ETS Admin Panel - Dashboard

Add New Menu Item

Name:

Price (RM):

Description:

Stock Quantity:

Image: No file chosen

Stock Status: In Stock

Available Menu Items

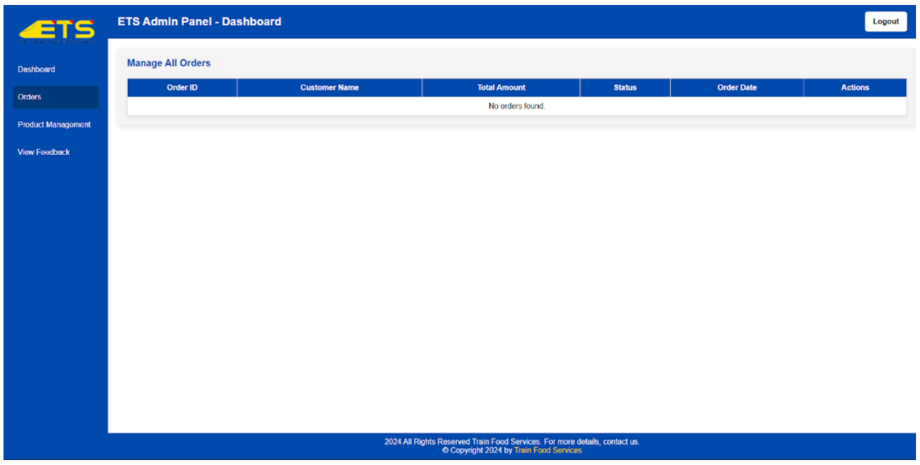
Name	Description	Price	Stock Quantity	Stock Status	Image	Actions
No items available						

2024 All Rights Reserved Train Food Services. For more details, contact us.
© Copyright 2024 by Train Food Services

Figure 5.9: Product Management Page

<u>View Order</u>

Step 1: To view customer feedback, admins can click on the View Feedback section. This displays feedback submitted by customers, allowing admins to assess customer satisfaction.



ETS Admin Panel - Dashboard

Manage All Orders

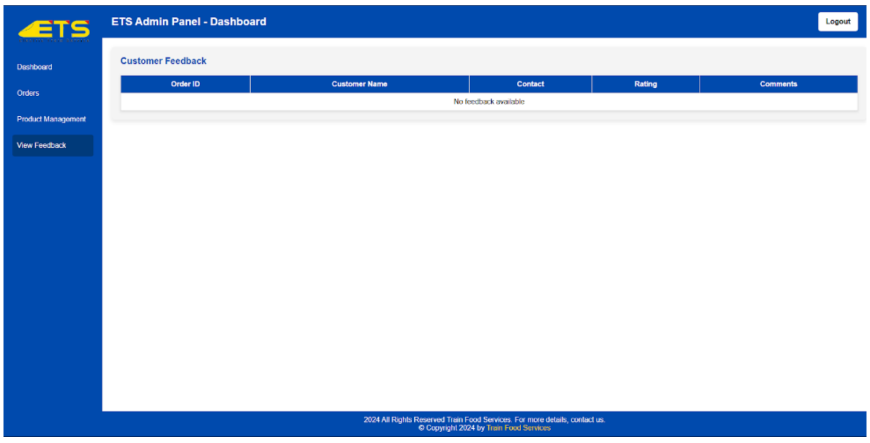
Order ID	Customer Name	Total Amount	Status	Order Date	Actions
No orders found.					

2024 All Rights Reserved Train Food Services. For more details, contact us.
© Copyright 2024 by Train Food Services

Figure 5.10: View OrderPage for Admin

View Feedback

Step 1: Click on the View Feedback box to see a list of feedback from customers.



Customer Feedback

Order ID	Customer Name	Contact	Rating	Comments
No feedback available				

2024 All Rights Reserved Train Food Services. For more details, contact us.
© Copyright 2024 by Train Food Services

Figure 5.11: View Feedback Page for Admin

5.2. FLOWCHART

User Flowchart

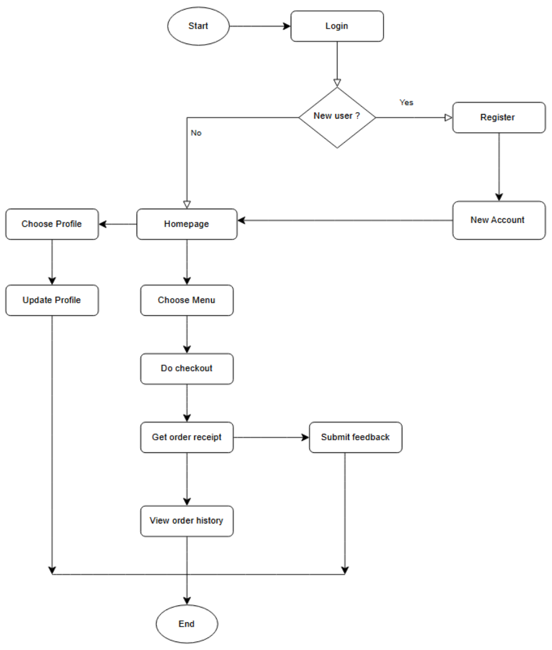


Figure 5.12: User Flowchart

The user flowchart visually represents the steps customers follow to interact with the system, from logging in to placing an order and providing feedback.

Admin Flowchart

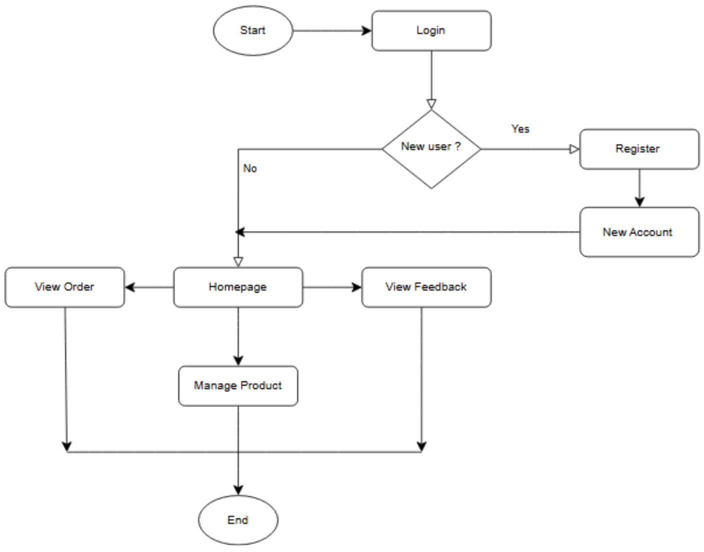


Figure 5.13: AdminFlowchart

The admin flowchart illustrates the process admins follow to manage products, view orders, and access customer feedback, ensuring smooth operations of the system.

5.3. DISCUSSION

The system is designed to provide a smooth and user-friendly experience for both customers and administrators. For customers, the process of booking an order is simple. They can log in, select their desired menu items, choose a payment method, and complete their order by checking out. After receiving their food, customers are encouraged to provide feedback, which is a valuable feature for improving service quality. This process ensures that customers have an easy and pleasant experience while using the platform. However, adding features such as order tracking or a support chatbot could further enhance the user experience.

On the administrative side, the system allows for efficient product management and order tracking. Administrators can log in to add or update products and view customer orders with ease. This streamlined functionality helps save time and improves workflow. That said, there is room for improvement, such as including advanced tools like sales analytics, automated stock alerts, and a dashboard for better oversight. Adding these features would make administrative tasks even more efficient and effective.

The system's design, as illustrated in the flowcharts, ensures a logical and easy-to-follow workflow for both users and administrators. The use of a relational database helps manage customer data, orders, and product information efficiently. Despite these strengths, challenges were encountered during development, such as integrating secure payment methods and optimizing the interface for all devices. These challenges were addressed through careful planning, testing, and improvements.

Looking forward, the system has growth potential. Future updates could include additional customer features like loyalty rewards, notifications, and a more personalized user experience. For administrators, tools like automated reporting, better search functionality, and advanced analytics could be added. Overall, the system successfully achieves its goal of simplifying order booking for customers and streamlining management tasks for administrators. With ongoing improvements, it can become a robust and comprehensive solution.