

# **Project Report**

**“Unimo Family – Friendly”**

**Restaurant Management System**

**Group – 15**

GROUP - 15

## DECLARATION

We, the undersigned, declare that this project report titled “Unimo Family – Friendly Restaurant Management System” is our original work and has not been submitted for any other degree or professional qualifications.

Team Members:

M.P.F.Farjana

.....

M.M.Safeeya

.....

B.M.D.Basanayake

.....

B.G.A.I.Bathalawaththa

.....

G.G.J.S.P.K.Gawarathenna

.....

M.Dhushanthani

.....

## **ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our project supervisor, for their invaluable guidance and support throughout the project. We also thank our friends, family, and team members for their encouragement and assistance. Special thanks to Unimo Family Friendly Restaurant for providing us the opportunity and resources to undertake this project.

GROUP - 15

# TABLE OF CONTENTS

Declaration .....	II
Acknowledgement .....	III
Table of Contents .....	IV
Table of Figures .....	VII
List of Tables .....	X
<b>Chapter 1</b> .....	<b>1</b>
1.1 Introduction .....	1
1.1.1 Problem Summary .....	1
1.2 Background and Motivation .....	1
1.2.1 Background .....	1
1.2.2 Motivation .....	2
1.3 Aim, Objectives, and Scope .....	2
1.3.1 Aim .....	2
1.3.2 Objectives .....	2
1.3.3 Scope .....	2
<b>Chapter 2</b> .....	<b>3</b>
2.1 Propose Solutions .....	3
<b>Chapter 3</b> .....	<b>4</b>
3.1 Resource Requirements .....	4
<b>Chapter 4</b> .....	<b>5</b>
4.1 Technology Adapted .....	5

<b>Chapter 5</b>	<b>6</b>
5.1 Design	6
5.1.1 Wireframes	6
5.1.1.1 Wireframe of User Login System	6
5.1.1.2 Wireframe of Home Page	7
5.1.1.3 Wireframe of Reservation Page	8
5.1.1.4 Wireframe of Orders Page	9
5.1.1.5 Wireframe of Payments Page	10
5.1.1.6 Wireframe of Menu Page	11
5.1.1.7 Wireframe of Reports Page	12
5.1.1.8 Wireframe of Admin Login System	13
5.1.1.9 Wireframe of Admin Page	14
5.1.2 Diagrams	15
5.1.2.1 ER Diagram	15
5.1.2.2 Context Diagram	16
5.1.2.3 Data Flow Diagrams	17
5.1.2.3.1 DFD Diagram Level 0	17
5.1.2.3.1 DFD Diagram Level 1	18
5.1.2.4 Class Diagram	19
5.1.2.5 Use Case Diagram	20
<b>Chapter 6</b>	<b>21</b>
6.1 Pages	21
6.1.1 User Login Page	21
6.1.2 Home Page	22
6.1.3 Reservation Page	23

6.1.4 Orders Page .....	25
6.1.5 Menu Page .....	27
6.1.6 Payments Page .....	29
6.1.7 Reports Page .....	30
6.1.8 Admin Login Page .....	32
6.1.9 Admin Page.....	33
<b>Chapter 7 .....</b>	<b>34</b>
7.1 Conclusion and Further Works .....	34
7.1.1 Conclusion .....	34
7.1.2 Further Works .....	34
<b>Chapter 8 .....</b>	<b>36</b>
8.1 Summary .....	36
<b>Chapter 9 .....</b>	<b>38</b>
9.1 References .....	38
<b>Chapter 10 .....</b>	<b>40</b>
10.1 Appendixes.....	40
10.1.1 Appendix A: Sample Code Snippets.....	40
10.1.2 Appendix B: User Manuals.....	46
10.1.3 Appendix C: Project Plan Timeline .....	47

## TABLE OF FIGURES

<b>Figure 1</b> .....	<b>6</b>
1.1 Wireframe of User Login System .....	6
<b>Figure 2</b> .....	<b>7</b>
2.1 Wireframe of Home Page.....	7
<b>Figure 3</b> .....	<b>8</b>
3.1 Wireframe of Reservation Page .....	8
<b>Figure 4</b> .....	<b>9</b>
4.1 Wireframe of Orders Page.....	9
<b>Figure 5</b> .....	<b>10</b>
5.1 Wireframe of Payments Page.....	10
<b>Figure 6</b> .....	<b>11</b>
6.1 Wireframe of Menu Page .....	11
<b>Figure 7</b> .....	<b>12</b>
7.1 Wireframe of Reports Page .....	12
<b>Figure 8</b> .....	<b>13</b>
8.1 Wireframe of Admin Login System.....	13
<b>Figure 9</b> .....	<b>14</b>
9.1 Wireframe of Admin Page .....	14
<b>Figure 10</b> .....	<b>15</b>
10.1 ER Diagram.....	15
<b>Figure 11</b> .....	<b>16</b>
11.1 Context Diagram .....	16
<b>Figure 12</b> .....	<b>17</b>
12.1 DFD Level 0 Diagram.....	17

<b>Figure 13</b> .....	<b>18</b>
13.1 DFD Level 1 Diagram .....	18
<b>Figure 14</b> .....	<b>19</b>
14.1 Class Diagram .....	19
<b>Figure 15</b> .....	<b>20</b>
15.1 Use Case Diagram.....	20
<b>Figure 16</b> .....	<b>21</b>
16.1 User Login System .....	21
<b>Figure 17</b> .....	<b>22</b>
17.1 Home Page .....	22
<b>Figure 18</b> .....	<b>24</b>
18.1 Reservation Page .....	24
<b>Figure 19</b> .....	<b>26</b>
19.1 Orders Page .....	26
<b>Figure 20</b> .....	<b>27</b>
21.1 Menu Page.....	27
<b>Figure 21</b> .....	<b>29</b>
20.1 Payments Page.....	29
<b>Figure 22</b> .....	<b>30</b>
22.1 Reports Page.....	30
<b>Figure 23</b> .....	<b>32</b>
23.1 Admin Login Page .....	32
<b>Figure 24</b> .....	<b>33</b>
24.1 Admin Page .....	33



<b>Figure 25</b> .....	<b>40</b>
25.1 Login Interface Code.....	40
<b>Figure 26</b> .....	<b>41</b>
26.1 Menu Page Code .....	41
<b>Figure 27</b> .....	<b>42</b>
27.1 Reservation Page Code.....	42
<b>Figure 28</b> .....	<b>43</b>
28.1 Orders Page Code.....	43
<b>Figure 29</b> .....	<b>44</b>
29.1 Admin Page Code.....	44
<b>Figure 30</b> .....	<b>45</b>
30.1 Delete Menu Code.....	45
<b>Figure 31</b> .....	<b>45</b>
31.1 Update Menu Code.....	45
<b>Figure 32</b> .....	<b>46</b>
32.1 Print Bill Code.....	46

## LIST OF TABLES

<b>Table 1</b> .....	<b>4</b>
1.1 Resource Requirements.....	4
<b>Table 2</b> .....	<b>5</b>
1.2 Technology Adapted .....	5
<b>Table 3</b> .....	<b>47</b>
1.3 Project Plan Timeline .....	47

# **CHAPTER 1**

## **1.1 INTRODUCTION**

Client Name and Address:

Unimo Family – Friendly Restaurant,

No 125,

Kandy Road,

Madawala

### **1.1.1 PROBLEM SUMMARY**

Unimo Family – Friendly Restaurant’s way of handling orders, inventory, reservations, and customer information is done by manually, leading to errors and inefficiencies. There is a need for an automated system to streamline operations, enhance customer satisfaction, and improve overall restaurant management system.

## **1.2 BACKGROUND AND MOTIVATION**

### **1.2.1 BACKGROUND**

Unimo Family – Friendly Restaurant faces challenges with its current manual processes, such as frequent order errors, mismatched inventory counts, and difficulties in managing customer reservations. These issues negatively impact customer satisfaction and operational efficiency.

## **1.2.2 MOTIVATION**

To address these challenges, Unimo Family – Friendly Restaurant seeks to implement an automated system. The new system aims to reduce errors, optimize resource usage, and improve customer satisfaction, thereby enhancing the overall efficiency of the restaurant's operations.

## **1.3 AIM, OBJECTIVES, AND SCOPE**

### **1.3.1 AIM**

To develop a robust restaurant management system for Unimo Family – Friendly Restaurant that automates processes, enhances efficiency, and improves customer satisfaction.

### **1.3.2 OBJECTIVES**

- Order Management: Ensure quick and easy handling of orders.
- Inventory Management: Track inventory accurately and optimize resource usage.
- Reservation System: Manage table bookings effectively.
- Customer Management: Maintain detailed records of customers and their preferences.
- Reporting and Analytics: Generate reports to monitor restaurant performance and support data-driven decision-making.
- User Interface Development: Create an intuitive interface for staff to interact with the system easily.

### **1.3.3 SCOPE**

The new system will manage orders, inventory, reservations, customer information, and provide basic reports on restaurant performance.

## CHAPTER – 2

### 2.1 PROPOSED SOLUTION

We propose developing a web-based management system using HTML, CSS, JavaScript, and PHP. The system will include modules for order management, inventory management, reservation handling, customer management, and reporting.

Key Features:

- **Order Management:** Easy order taking and processing, order history viewing, and status updates.
- **Inventory Management:** Monitoring inventory levels, automated reordering, and waste reduction.
- **Reservation System:** Efficient management and modification of reservations.
- **Customer Management:** Tracking customer information and preferences for better service.
- **Reporting and Analytics:** Generating sales, inventory, and customer preference reports to aid in decision-making.

## CHAPTER – 3

### 3.1 RESOURCE REQUIREMENTS

- **Hardware**
  - **Server:** A computer to store information and run the system.
  - **Client Computers:** Computers for staff to access the system
- **Software**
  - **Programming Languages:** HTML, CSS, JavaScript, PHP
  - **Development Tools:** Integrated Development Environment (IDE) – Visual Studio Code
  - **Database Management System (DBMS):** MySQL

Hardware	Server
	Client Computers
Software	HTML
	CSS
	JavaScript
	PHP
	MySQL
	IDE (Visual Studio)

Table 1.1 – Resource Requirements

## CHAPTER – 4

### 4.1 TECHNOLOGY ADOPTED

The project utilizes the following technologies:

- **HTML and CSS:** For front-end design and layout.
- **JavaScript:** For client-side scripting and interactivity.
- **PHP:** For server-side scripting and database interactions.
- **MySQL:** For database management.

Technology	Purpose
HTML	Structure and content of web pages
CSS	Styling and layout of web pages
JavaScript	Client-side scripting and interactivity
PHP	Server-side scripting and back-end development
MySQL	Database management

Table 1.2 – Technology Adapted

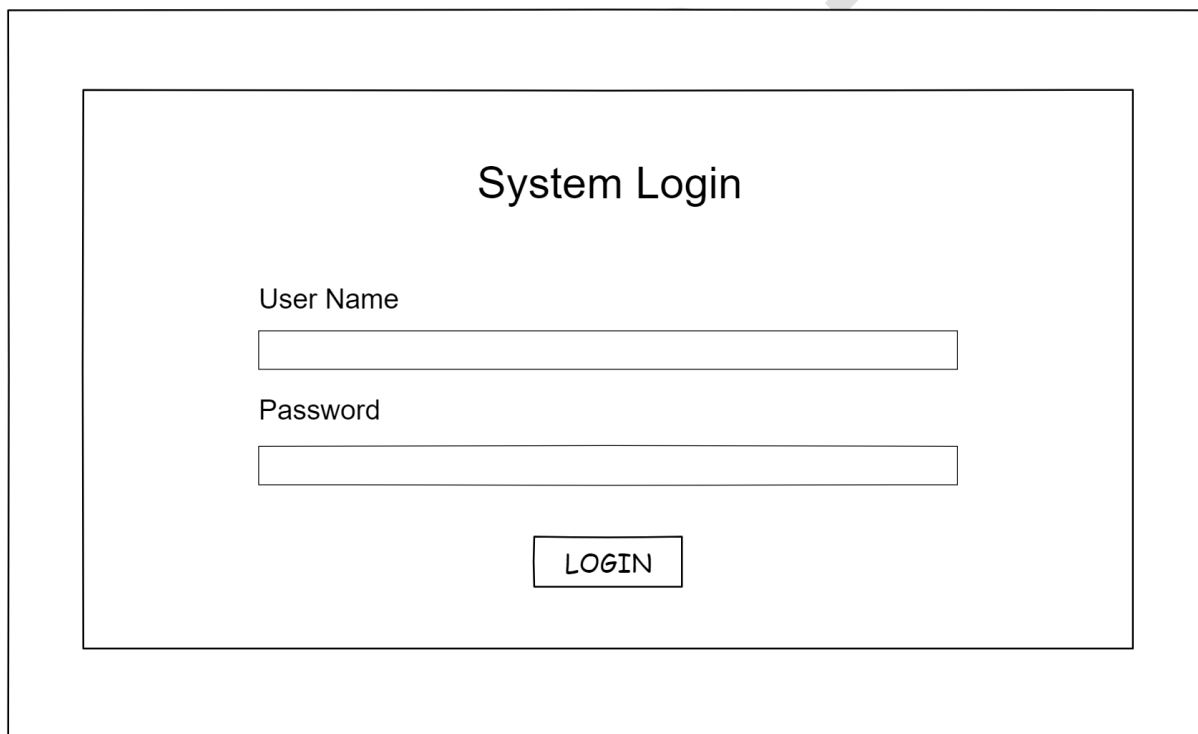
## CHAPTER 5

### 5.1 DESIGN

#### 5.1.1 WIREFRAMES

##### 5.1.1.1 WIREFRAME OF USER LOGIN SYSTEM

The user login system wireframe includes input fields for username and password, and a submit button. This allows users to securely log in to the system to access personalized features and services.



The wireframe illustrates a user login interface. It features a central rectangular box containing the title "System Login". Below the title, there are two input fields: one for "User Name" and one for "Password". A "LOGIN" button is positioned below the password field. The entire login form is enclosed within a larger rectangular frame.

Figure 1.1 - Wireframe of User Login System



### 5.1.1.2 WIREFRAME OF HOME PAGE

The home page wireframe shows the layout and design elements including navigation bar, promotional sections, and featured items. This provides users with a welcoming and informative introduction to the restaurant's offerings.

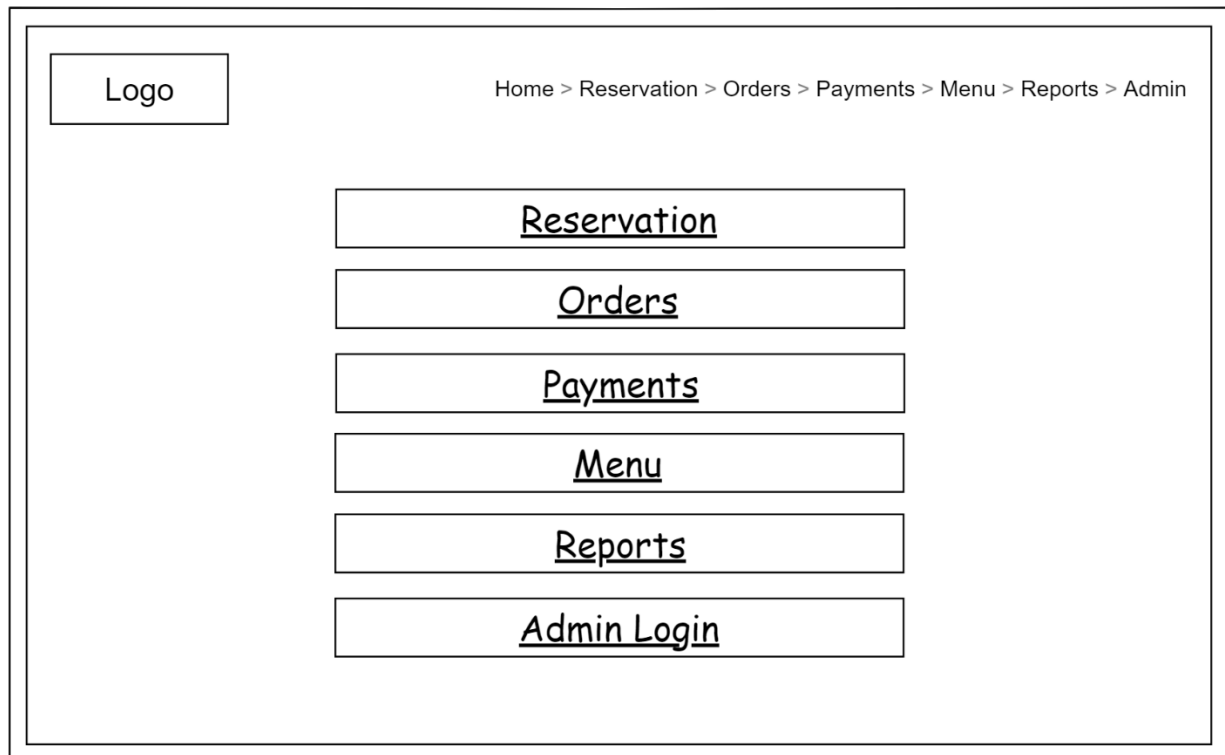


Figure 2.1 - Wireframe of Home Page

### 5.1.1.3 WIREFRAME OF RESERVATION PAGE

The reservation page wireframe outlines the form for booking tables, including date, time, number of guests, and special requests. This ensures an organized and efficient way for customers to reserve tables.

Logo

Home > Reservation > Orders > Payments > Menu > Reports > Admin

First Name

Last Name

Number of Guests

Email

Phone Number

Date

Time

Table Number

Special Occasion

Submit

Table Number	Table Status
01	Available
02	Occupied

### Reservation List

Reservation ID	First Name	Last Name	Number of Guests	Email	Phone Number	Date	Time	Table Number	Special Occasion	Delete
										Delete
										Delete

Figure 3.1 - Wireframe of Reservation Page

#### 5.1.1.4 WIREFRAME OF ORDERS PAGE

The orders page wireframe provides a detailed view of the menu items available for order, along with options for customization. Users can add items to their cart and proceed to checkout. This facilitates an efficient ordering process.

Logo

Home > Reservation > Orders > Payments > Menu > Reports > Admin

Reservation ID

Table Number

Item Ordered

Item Amount

Quantity

CALCULATE

Total Amount

SUBMIT

Reservation ID	Table Number	Number of Guests

Orders List

Reservation ID	Order ID	Table Number	Item Ordered	Item Amount	Quantity	Total Amount	Delete
							Delete
							Delete
							Delete
							Delete
							Delete
							Delete

Figure 4.1 - Wireframe of Orders Page

### 5.1.1.5 WIREFRAME OF PAYMENTS PAGE

The payments page wireframe outlines the layout for processing payments, including fields for entering payment details, selecting payment methods, and a submit button. This ensures a smooth and secure payment process for users.

Logo

Home > Reservation > Orders > Payments > Menu > Reports > Admin

Reservation ID

Table ID

Table Number

Grant Total Amount

CALCULATE

Payment Method

Cash

Debit Card

Credit Card

SUBMIT

PRINT BILL

R/ID	O/ID	T/Num	Item/O	Amount	Qun	Total/A

### Payments List

Reservation ID	Table Number	Grant Total Amount	Payment Method	Delete
				Delete
				Delete

Figure 5.1 - Wireframe of Payments Page

### 5.1.1.6 WIREFRAME OF MENU PAGE

The menu page wireframe displays the layout for listing food and beverage items categorized by sections like starters, mains, desserts, and drinks. This helps customers easily navigate and choose from the available menu.

Logo

Home > Reservation > Orders > Payments > Menu > Reports > Admin

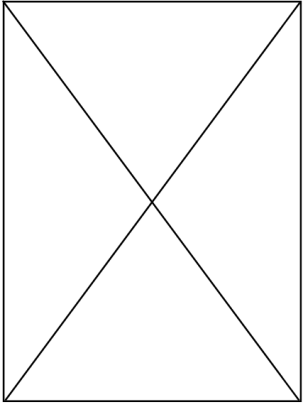
Item ID

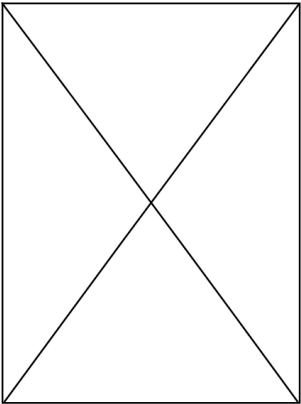
Item Category

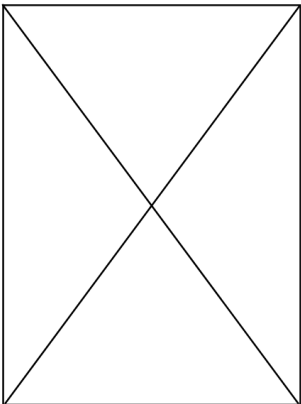
Item Name

Item Amount

SUBMIT







**Menu List**

Item ID	Item Category	Item Name	Item Amount	Delete	Edit
				<input type="button" value="Delete"/>	<input type="button" value="Edit"/>
				<input type="button" value="Delete"/>	<input type="button" value="Edit"/>

Figure 6.1 - Wireframe of Menu Page

### 5.1.1.7 WIREFRAME OF REPORTS PAGE

The reports page wireframe shows the layout for generating and viewing various reports on sales, inventory, and customer data. This allows administrators to monitor and analyze restaurant performance effectively.

Reports

Generate Reports

Reservation List

Reservation ID	First Name	Last Name	Number of Guests	Email	Phone Number	Date	Time	Table Number	Special Occasion

Orders List

Reservation ID	Order ID	Table Number	Item Ordered	Item Amount	Quantity	Total Amount

Payments List

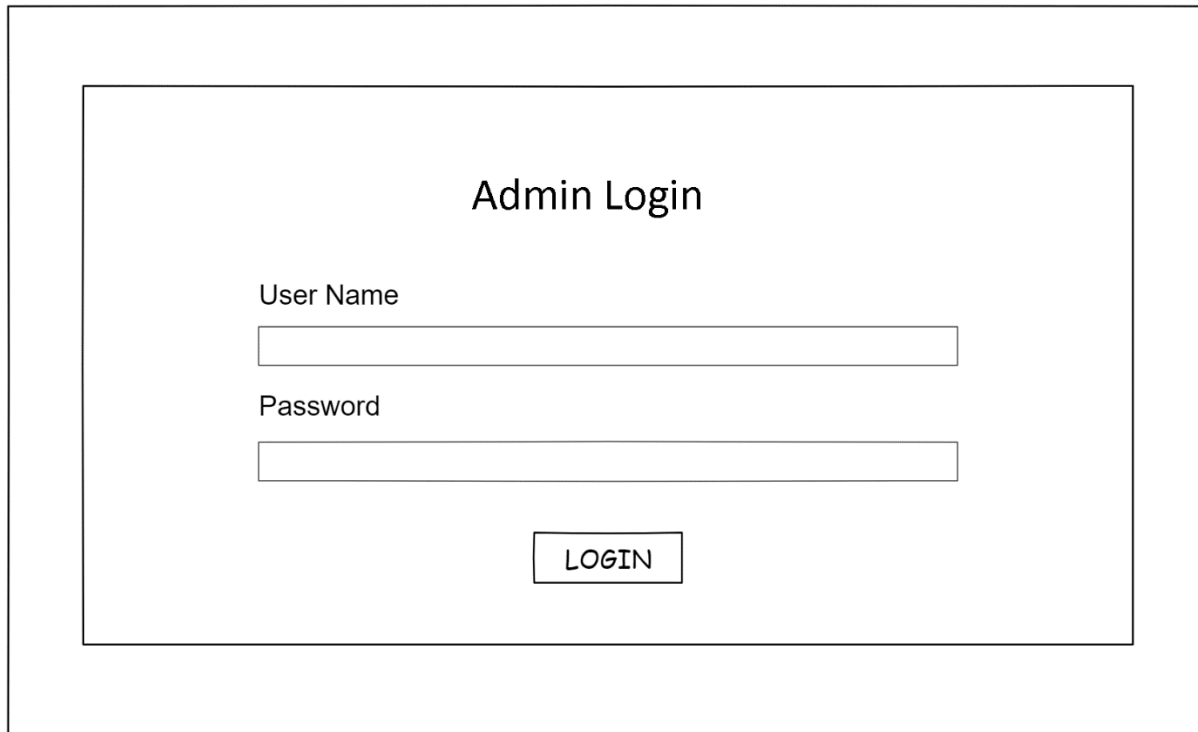
Reservation ID	Table Number	Grant Total Amount	Payment Method

Download Report

Figure 7.1 - Wireframe of Reports Page

### 5.1.1.8 WIREFRAME OF ADMIN LOGIN SYSTEM

The admin login system wireframe includes input fields for admin username and password, along with a submit button. This provides a secure way for administrators to access.



The wireframe shows a rectangular container with a double border. Inside, the title "Admin Login" is centered at the top. Below the title, the label "User Name" is positioned to the left of a horizontal input field. Below the input field, the label "Password" is positioned to the left of another horizontal input field. At the bottom center, there is a rectangular button labeled "LOGIN".

Figure 8.1 - Wireframe of Admin Login System

### 5.1.1.9 WIREFRAME OF ADMIN PAGE

The admin page wireframe includes sections for managing employees. Administrator can add new employees, edit existing employee details, and delete employee records. This centralizes administrative functions, making it easier to maintain the system.

Logo

Home > Reservation > Orders > Payments > Menu > Reports > Admin

## Add Employees

Employee ID

Full Name

Date of Birth

mm/dd/yy

Address

Phone Number

Email

Job Title

Hours of Work

Salary

Submit

## Employee List

Employee ID	Full Name	Date of Birth	Address	Phone Number	Email	Job Title	Hours of Work	Salary	Delete	Edit
									Delete	Edit
									Delete	Edit

Figure 9.1 - Wireframe of Admin Page



## 5.1.2 DIAGRAMS

### 5.1.2.1 ER DIAGRAM

The ER diagram illustrates the database schema, showing entities such as Users, Reservations, Orders, Menu Items, and their relationships.

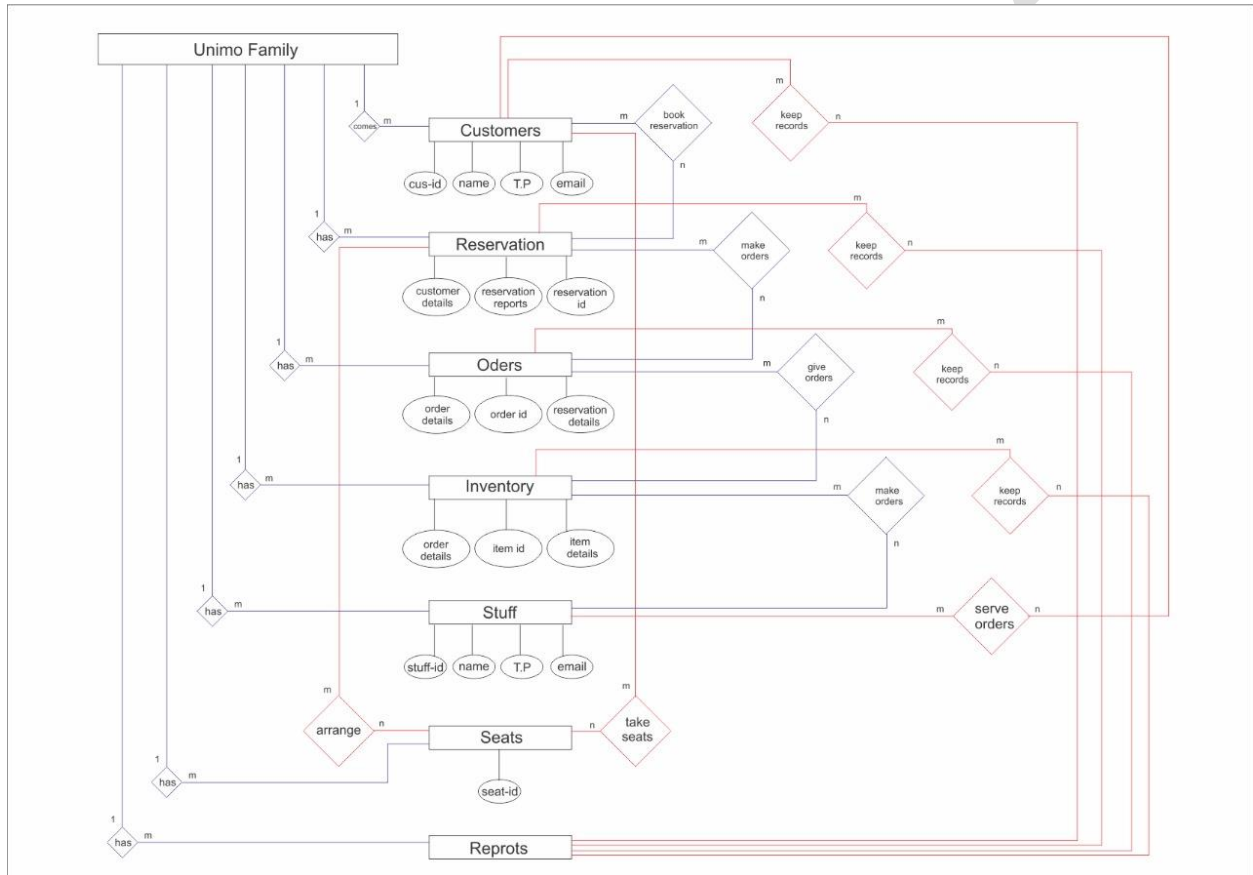


Figure 10.1 - ER Diagram

### 5.1.2.2 CONTEXT DIAGRAM

The context diagram provides a high-level overview of the system, showing how it interacts with external entities like customers and admin staff.

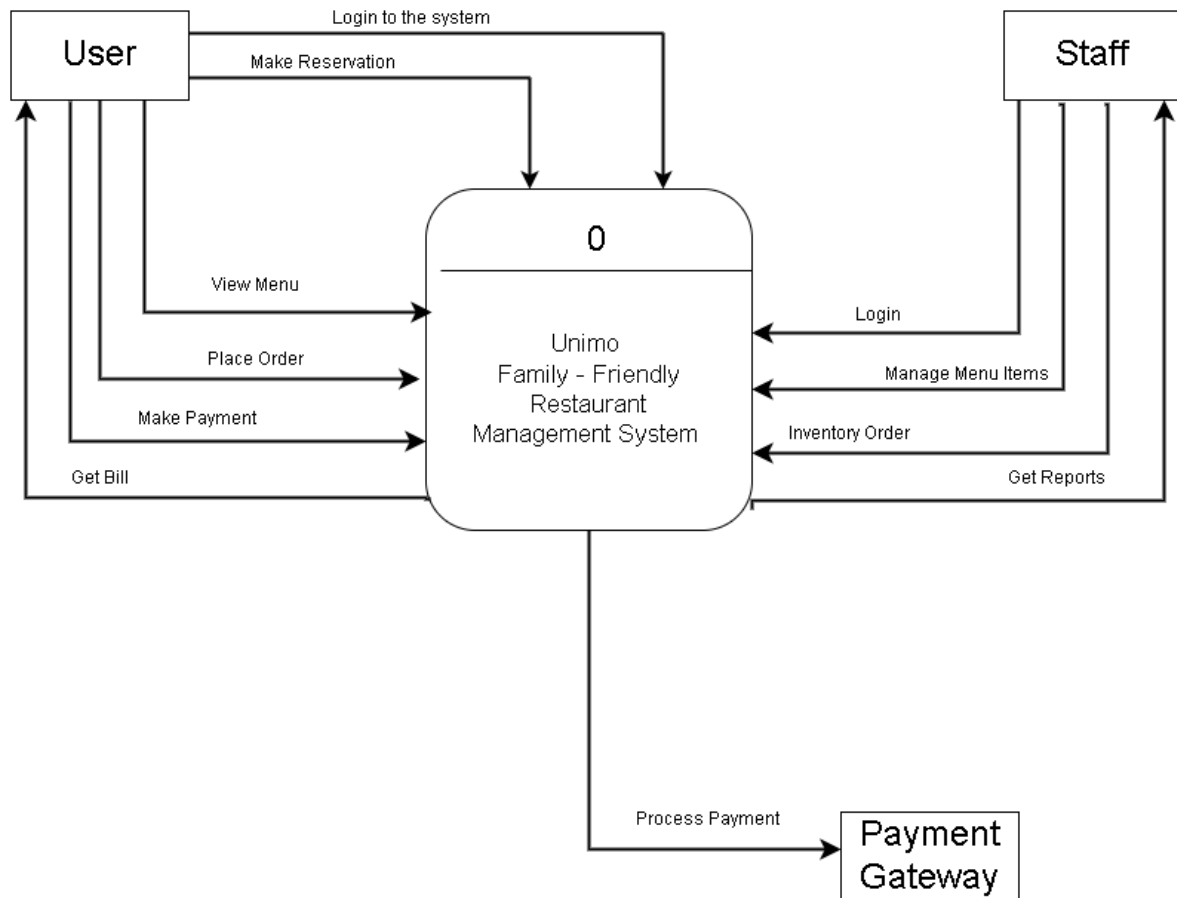


Figure 11.1 - Wireframe of Context Diagram

### 5.1.2.3 DATA FLOW DIAGRAMS

#### 5.1.2.3.1 DATA FLOW DIAGRAM - LEVEL 0

The Level 0 DFD shows the main processes of the system, including user registration, reservation handling, order processing, and report generation.

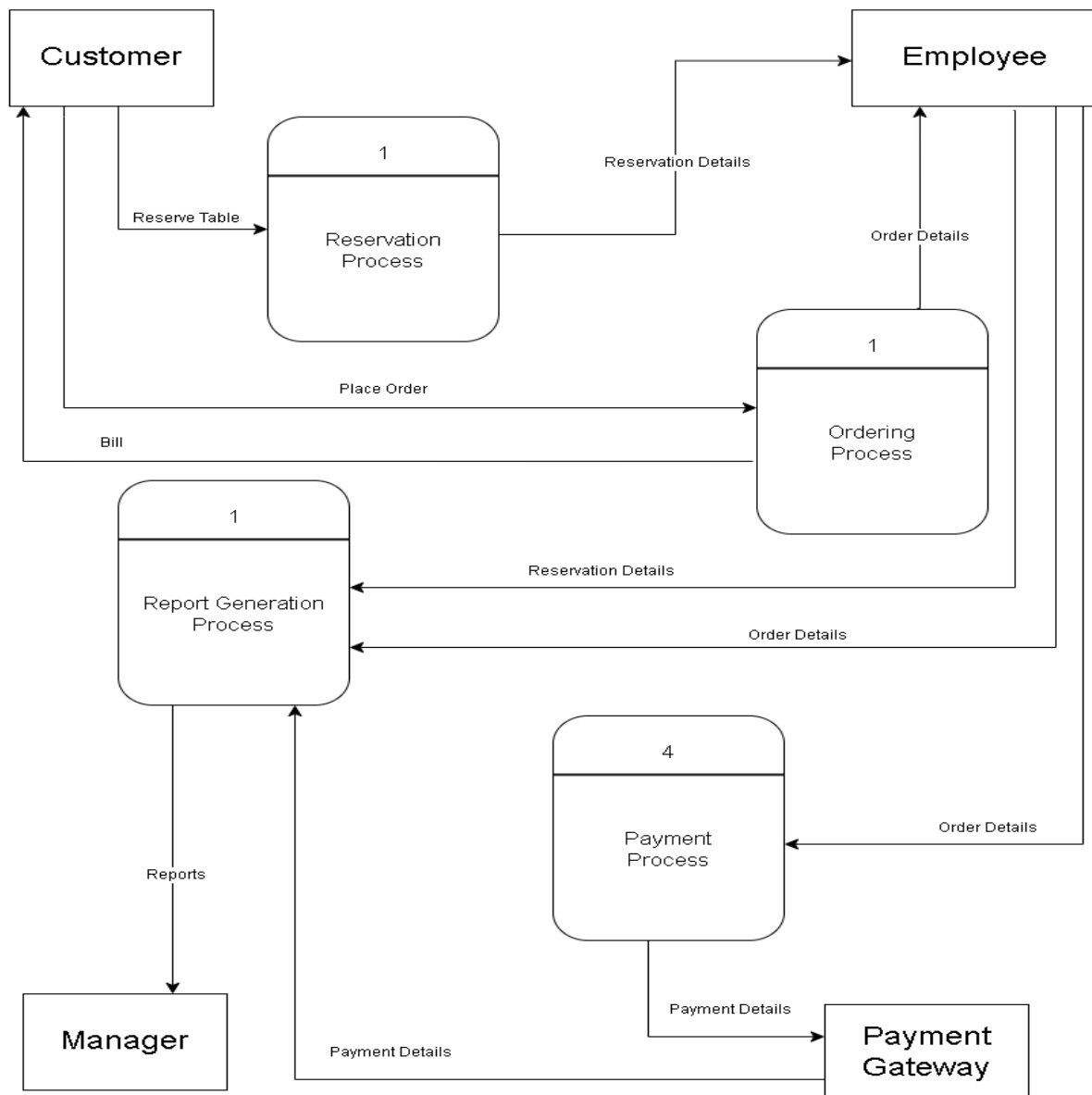


Figure 12.1 - DFD Level 0

### 5.1.2.3.2 DATA FLOW DIAGRAM - LEVEL 1

The Level 1 DFD provides a detailed view of the processes identified in Level 0, breaking down each process into sub-processes.

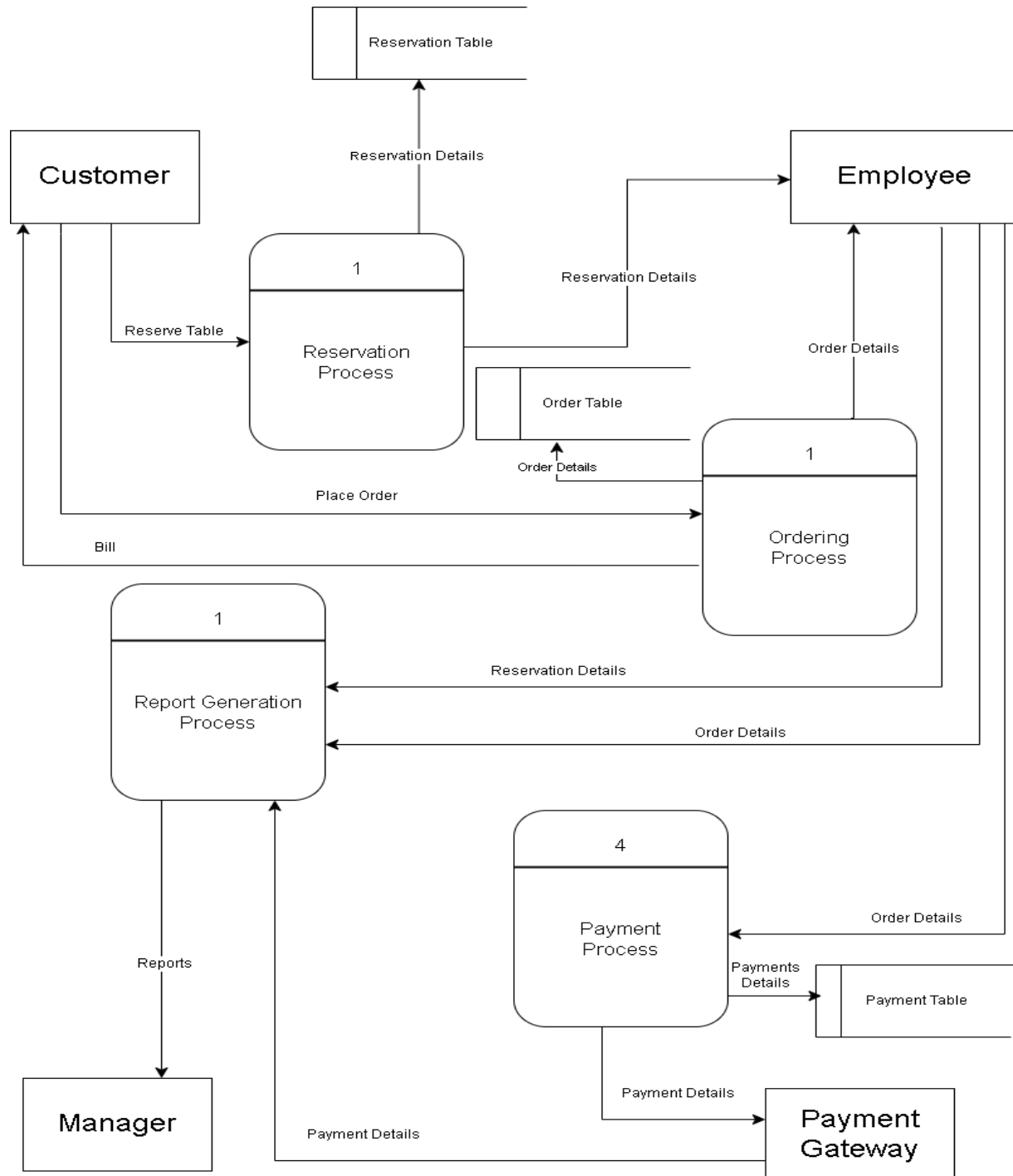


Figure 13.1 - DFD Level 1

#### 5.1.2.4 CLASS DIAGRAM

The class diagram details the object-oriented structure of the system, including classes such as User, Reservation, Order, Menu Item, and their attributes and methods.

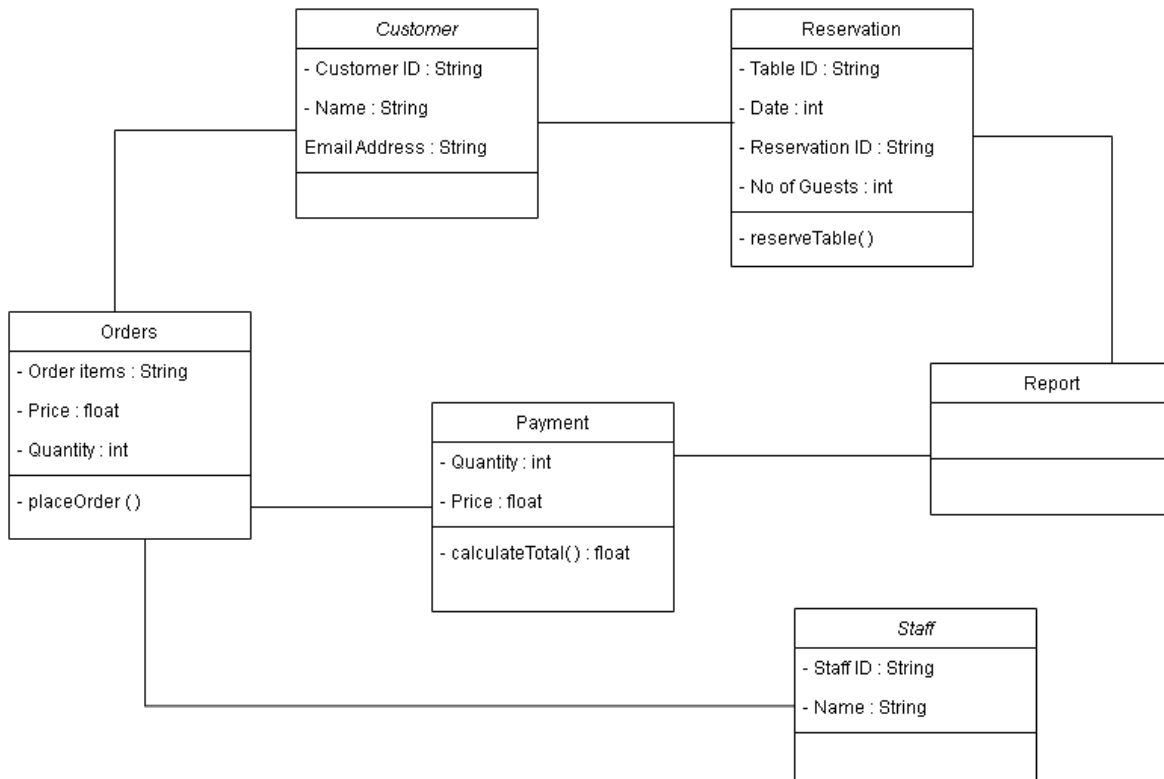


Figure 14.1 – Class Diagram

### 5.1.2.5 USE CASE DIAGRAM

The use case diagram shows the interactions between users (customers and administrators) and the system, highlighting functionalities like making reservation, placing orders, and managing menus.

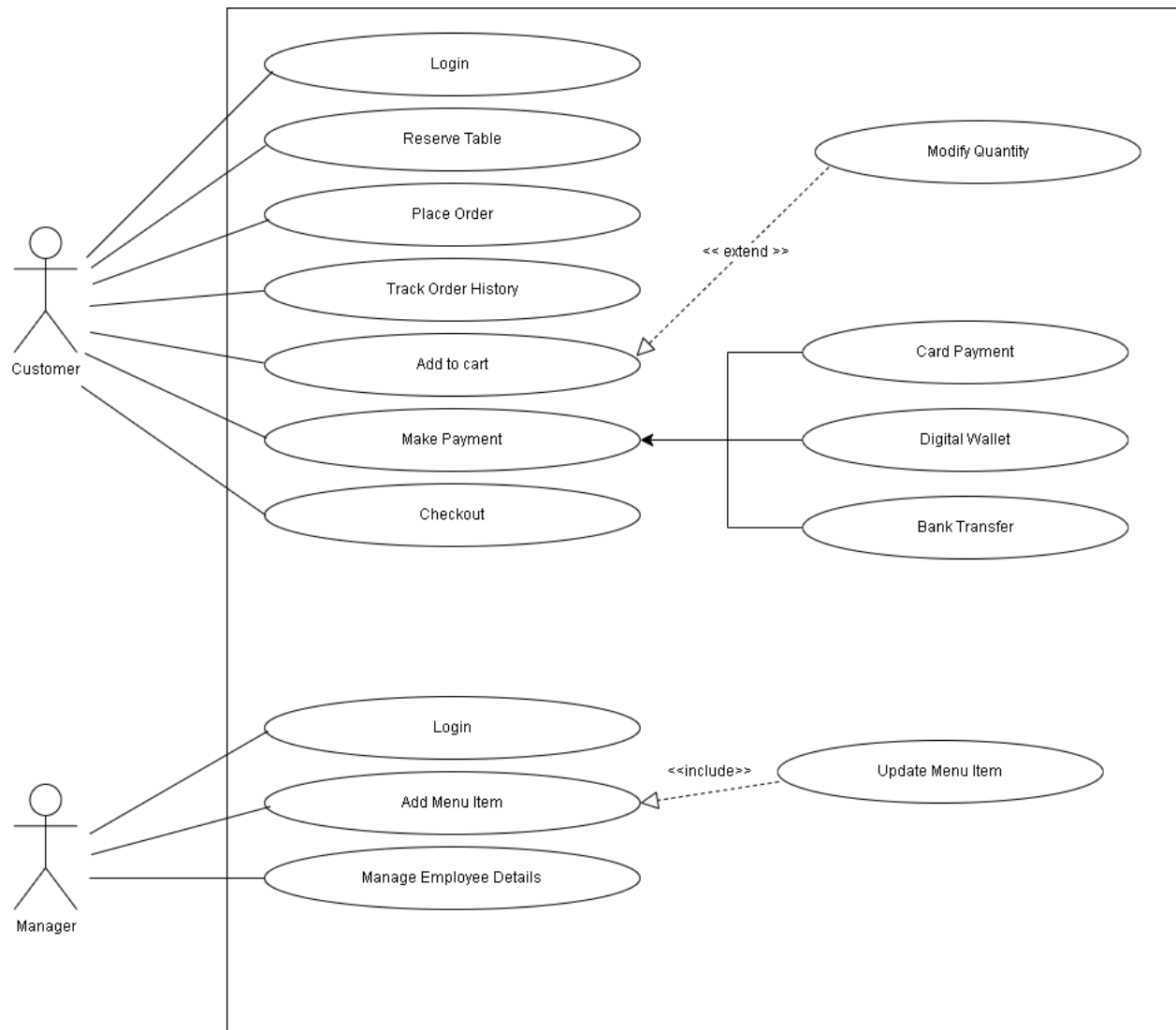


Figure 15.1 – Use Case Diagram

## CHAPTER – 6

### 6.1 PAGES

#### 6.1.1 USER LOGIN SYSTEM

The user login system is designed to authenticate users securely, allowing them to access their accounts and personalized features. It includes the following elements:

- **Username Field:** Allows users to enter their registered username.
- **Password Field:** Allows users to enter their password securely.
- **Login Button:** Submits the login form for authentication.

The login page allows users to authenticate themselves using their credentials.

- **HTML & CSS:**  
The page includes input fields for username, and password, styled with CSS for professional look.
- **JavaScript:**  
Client-side validation ensures that fields are not empty before submission.
- **PHP:**  
Server-side script verifies user credentials against the database.

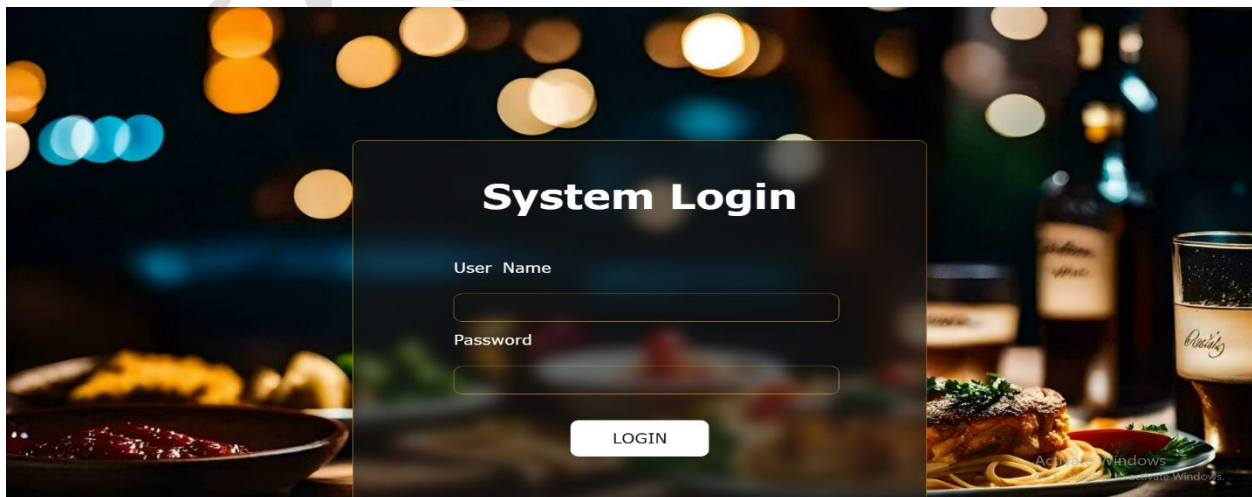


Figure 16.1 - User Login System

## 6.1.2 HOME PAGE

The home page serves as the main landing page for the "Unimo Family Friendly Restaurant Management System." It is designed to provide users with a welcoming and informative introduction to the restaurant's offerings. Key features include:

- **Navigation Bar:** Provides links to other sections of the website, such as Home, Menu, Reservations, Orders, Payments, Reports, and Admin Login Page.

The home page provides an overview of the restaurant's offerings and navigational links to other sections.

- **HTML & CSS:**  
Layout includes a navigation bar, promotional banners, and featured menu items.
- **JavaScript:**  
Enhances interactivity with sliders for promotions and animated elements.

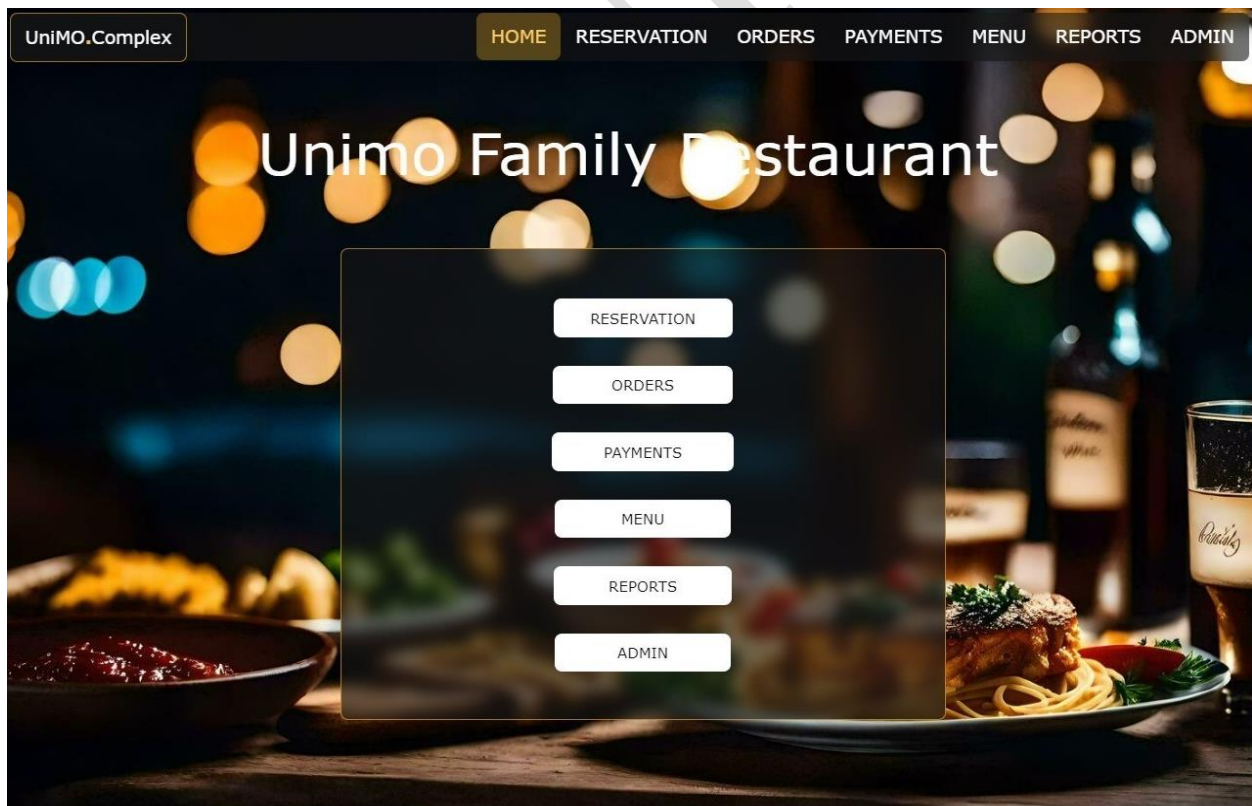


Figure 17.1 - Home Page



### 6.1.3 RESERVATION PAGE

The reservation page is designed to facilitate easy and efficient booking of tables for customers. It includes a form where users can provide the necessary details for their reservation.

- **Date Picker:** Allows users to select the desired date for their reservation.
- **Time Picker:** Allows users to choose the preferred time for their reservation.
- **Number of Guests:** A dropdown or input field to specify the number of guests.
- **Special Requests:** An optional text area for users to add any special requests or instructions.
- **Submit Button:** Submits the reservation form and confirms the booking.

Customers can make reservations for dining at the restaurant.

- **HTML & CSS:**  
Form fields for selecting date, time, number of guests, and special requests.
- **JavaScript:**  
Date picker and form validation.
- **PHP:**  
Handles form submission and stores reservation data in the database.

UniMO.Complex

HOMERESERVATIONORDERSPAYMENTSMENUREPORTSADMIN

# Reservation

First Name

Last Name

Number of Guests

E mail

Phone Number

Date

mm/dd/yyyy

Time

--:--:--

Table

Table ID

Table Number

Spacial Occasion

SUBMIT

Table Number	Table Status
1	Available
2	Available
3	Available
4	Available
5	Available
6	Available
7	Available
8	Available
9	Available
10	Available
11	Available
12	Available
13	Available
14	Available
15	Available
16	Available
17	Available
18	Available
19	Available
20	Available

## Reservation List

Reservation ID	First Name	Last Name	Number of Guests	E Mail	Phone Number	Date	Time	Table Number	Spacial Occasion	Delete
43	Jeewaka	Supun	2	jeewaka12345@gmail.com	715308663	2024-05-18	23:52:00	1	date	DELETE

Figure 18.1 - Reservation Page

#### 6.1.4 ORDERS PAGE

The orders page is designed to facilitate the ordering process for customers. Key features include:

- **Menu Display:** Lists all available food and beverage items categorized into sections (e.g., starters, mains, desserts, drinks).
- **Customization Options:** Allows customers to customize their orders (e.g., add extra toppings, specify cooking preferences).
- **Add to Cart Button:** Enables customers to add items to their shopping cart.
- **Cart Summary:** Displays a summary of items in the cart, with options to adjust quantities or remove items.
- **Checkout Button:** Proceeds to the payments page for order completion.

Users can place orders for food and beverages, with options for customization and special instructions.

- **HTML & CSS:**  
Menu items displayed with options to add to cart.
- **JavaScript:**  
Adds interactivity for order customization and cart management.
- **PHP:**  
Processes order submissions and updates database.

UniMO.Complex

HOMERESERVATIONORDERSPAYMENTSMENUREPORTSADMIN

Orders

Reservation ID

Table Number

Item Orderd

Item Amount

Quantity

CALCULATE

Total Amount

SUBMIT

Reservation ID	Table Number	Number of Guests
43	1	2
45	5	5
55	1	2

Orders List

Reservation ID	Order ID	Table Number	Item Orderd	Item Amount	Quantity	Total Amount	Delete
43	14	1	Submarine - UNIMO Special	1390.00	2	2780.00	DELETE
43	15	1	Beverages - Vanilla Milkshake	690.00	2	1380.00	DELETE
45	16	5	Chinese (Rice) - Fried Rice Chicken	890.00	5	4450.00	DELETE
55	25	1	Chicken - Crispy Chicken (SPCS)	1900.00	2	3800.00	DELETE

Figure 19.1 - Orders Page

26

### 6.1.5 MENU PAGE

The menu page is designed to display the restaurant's food and beverage offerings in an organized and user-friendly manner. It is divided into categories to help customers easily navigate and choose their desired items.

- **Category Sections:** Divides the menu into sections such as Starters, Mains, Desserts, and Drinks.
- **Menu Items:** Lists individual menu items with names, descriptions, and prices.
- **Images:** Includes images of menu items to enhance visual appeal.
- **Add to Cart Button:** Allows customers to add items to their order directly from the menu page.
- **Search Bar:** Provides a search function to quickly find specific items on the menu.

Displays the restaurant's menu, including categories and item descriptions.

- **HTML & CSS:**  
Structured layout for different menu categories.
- **JavaScript:**  
Filters and sorts menu items based on user preferences.

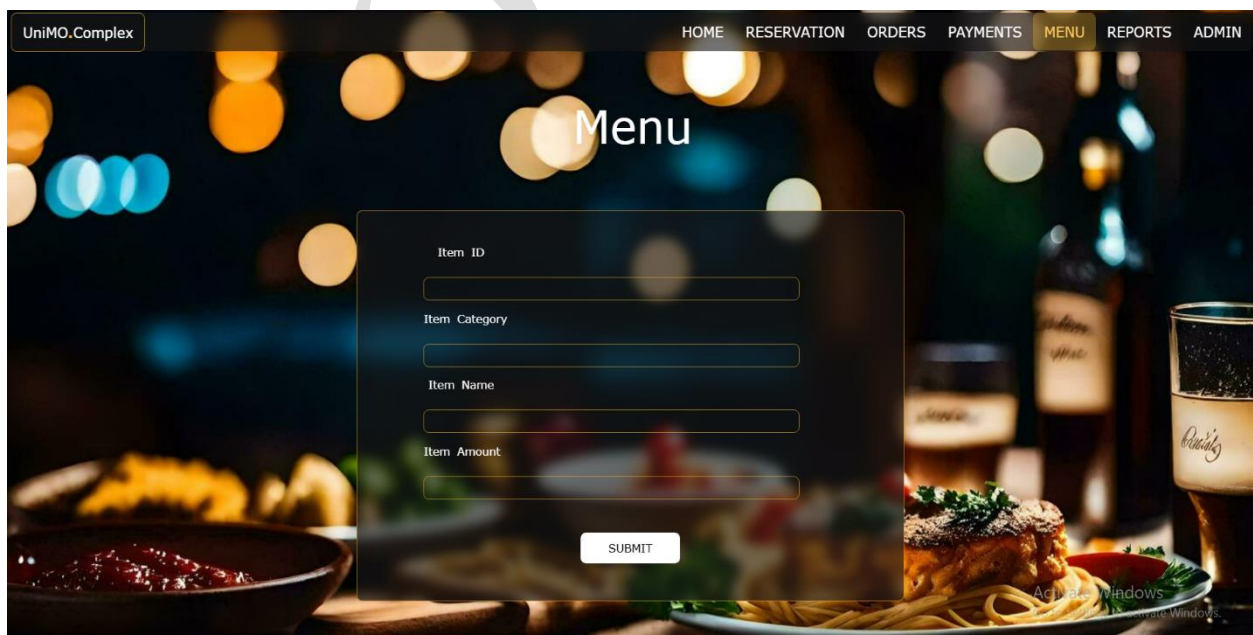


Figure 20.1 - Menu Page





## Menu List

Item ID	Item Category	Item Name	Item Amount	Delete	Edit
101	Starters	Chicken Sweet Corn Soup	690.00	DELETE	EDIT
102	Starters	Mix Green Salad	690.00	DELETE	EDIT
103	Starters	French Fries	690.00	DELETE	EDIT
104	Starters	French Fries with Chicken	790.00	DELETE	EDIT
105	Starters	French Fries with Nuggets	740.00	DELETE	EDIT
201	Submarine	Chicken	990.00	DELETE	EDIT
202	Submarine	Spicy Chicken	1140.00	DELETE	EDIT
203	Submarine	Beef	1190.00	DELETE	EDIT

## Edit Menu Items

Item Category

Starters

Item Name

Mix Green Salad

Item Amount

690.00

UPDATE

Figure 20.1 - Menu Page (Continued..)

## 6.1.6 PAYMENTS PAGE

The payments page ensures a smooth and secure payment process. Key elements include:

- **Payment Method Selection:** Allows users to choose their preferred payment method (e.g., credit card, PayPal).
- **Payment Details Form:** Collects necessary payment information (e.g., card number, expiration date, CVV).
- **Submit Payment Button:** Finalizes the payment and completes the order process.
- **Order Confirmation:** Displays a confirmation message with order details and estimated delivery time.

Reservation ID	Order ID	Table Number	Item Order	Item Amount	Quantity	Total Amount
43	14	1	Salmonsteak - UniMO Special	1390.00	2	2780.00
43	15	1	Beverages - Fanta, Mikschale	690.00	2	1380.00
45	16	5	Chicken (Rice) - Fried Rice Chicken	890.00	5	4450.00
55	25	1	Chicken - Grassy Chicken (SPCS)	1900.00	2	3800.00

Reservation ID	Table Number	Grand Total Amount	Payment Method	Delete
43	1	4160.00	Debit Card	DELETE
45	5	4450.00	Cash	DELETE

Figure 21.1 - Payments Page

### 6.1.7 REPORT PAGE

The reports page provides administrators with valuable insights into the restaurant's performance. Key features include:

- **Reservation Reports:** Detailed summaries of daily, weekly, and monthly reservations.
- **Orders Reports:** Information on current orders.
- **Customer Reports:** Data on customer orders, preferences, and feedback.
- **Payments Reports:** Data's of payments.
- **Employee Reports:** Data of staffs.

Admins can generate reports on various aspects of the restaurant's operations.

- **HTML & CSS:**  
Form for selecting report criteria.
- **JavaScript:**  
Enhances report generation with charts or tables.
- **PHP:**  
Queries database to generate report and displays results.

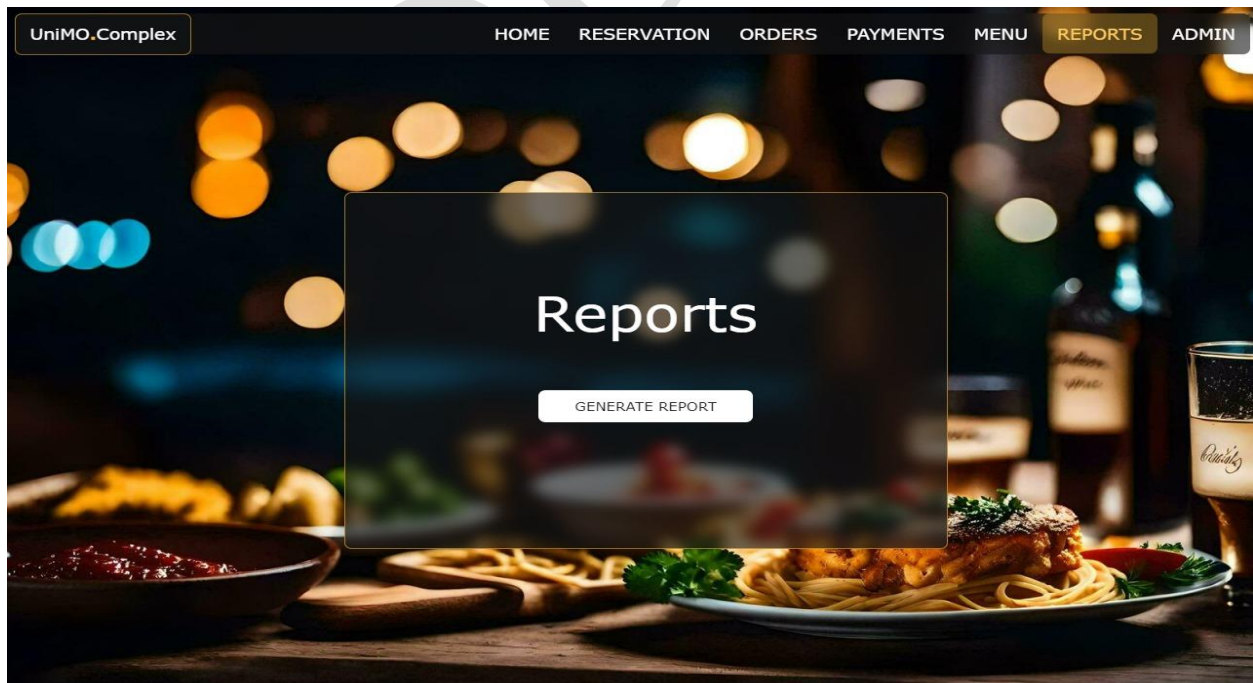


Figure 22.1 - Reports Page



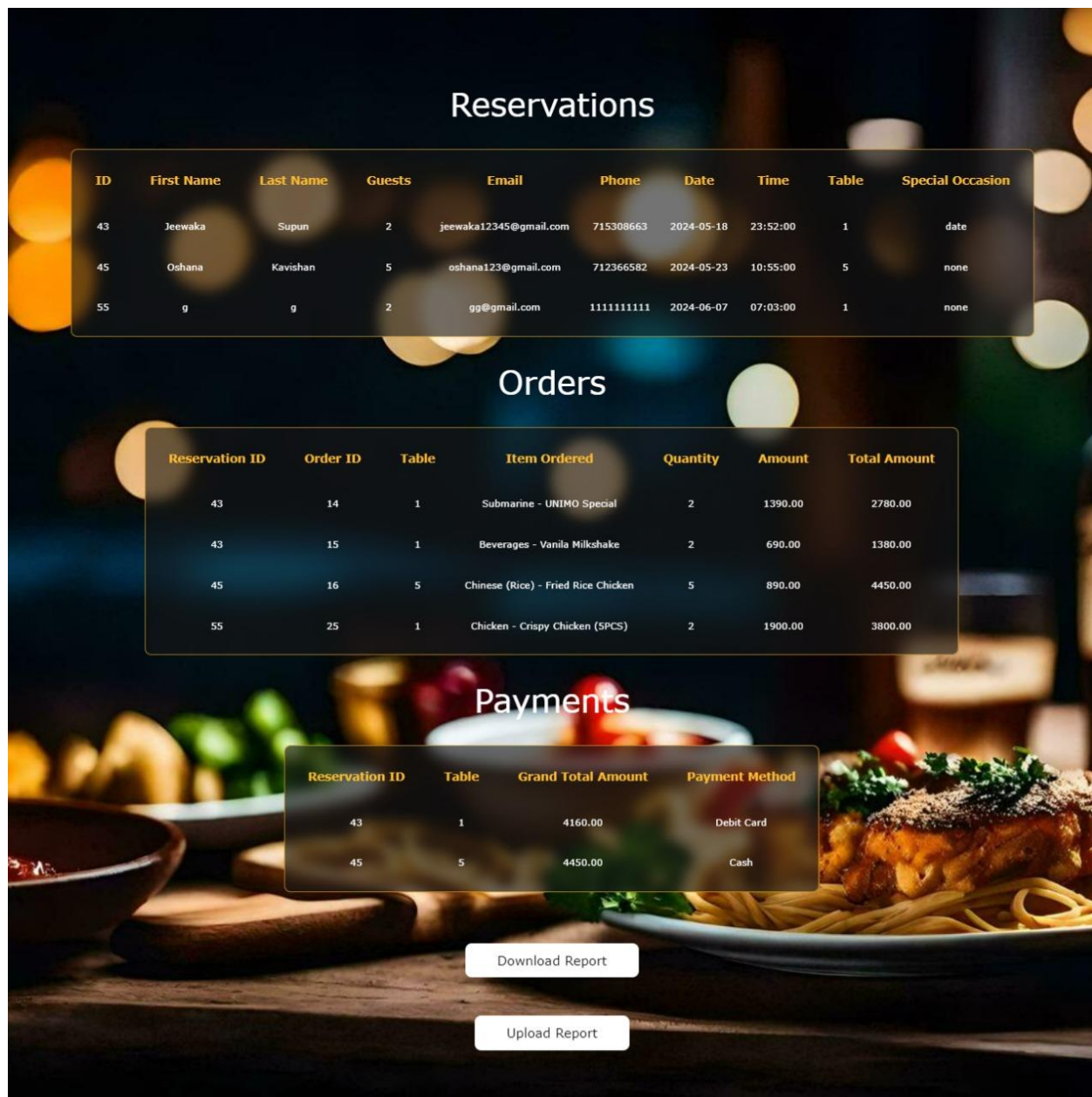


Figure 22.1 - Reports Page (Continued..)

### 6.1.8 ADMIN LOGIN SYSTEM

The admin login system provides a secure way for administrators to access the backend of the system. It includes:

- **Admin Username Field:** For entering the admin username.
- **Admin Password Field:** For entering the admin password.
- **Login Button:** Submits the admin login form for authentication.

A secure login page for administrators to access the backend management system.

- **HTML & CSS:**  
Styled similarly to the user login page for consistency.
- **JavaScript:**  
Client-side validation for admin credentials.
- **PHP:**  
Validates admin login and provides access to management features.

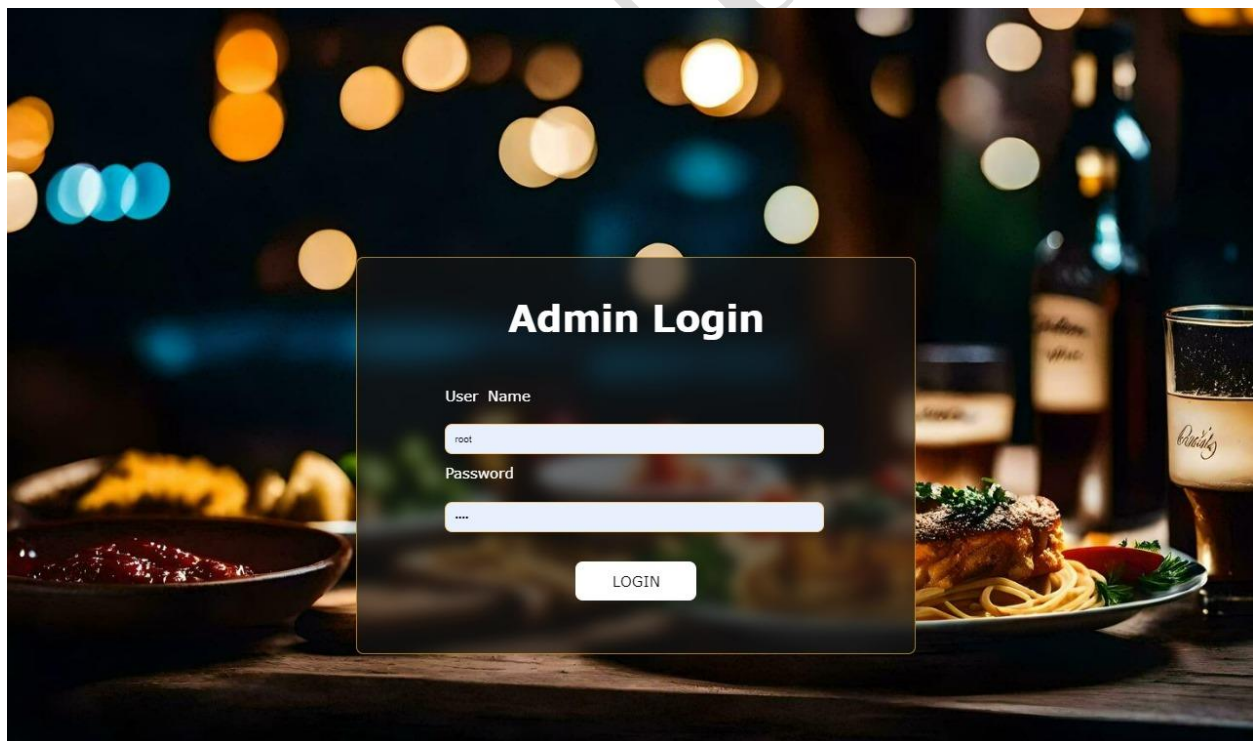


Figure 23.1 - Admin Login System

### 6.1.9. ADMIN PAGE

The admin page centralizes various administrative functions, making it easy to manage the system. Key features include:

- **Employee Management:** Administrator can manage employee details.
- **Add Employee Button:** Opens a form to add new employee details (e.g., name, position, contact information).
- **Edit Employee Button:** Allows editing of existing employee details.
- **Delete Employee Button:** Removes an employee from the system.
- **System Reports:** Access to sales, inventory, and customer reports for analysis.
- **Settings Configuration:** Options to configure system settings and preferences.

The screenshot displays the 'ADMIN' section of the 'UniMO.Complex' application. The main heading is 'Add Employees'. Below it is a form with the following fields: Employee ID, Full Name, Date of Birth (with a date picker), Address, Phone of Number, E mail, Job Title, Hours of Work, and Salary. A 'SUBMIT' button is at the bottom of the form. Below the form is the 'Employee List' table, which contains two rows of employee data. Each row has 'Delete' and 'Edit' buttons.

Employee ID	Full Name	Date of Birth	Address	Phone Number	E mail	Job Title	Hours of Work	Salary	Delete	Edit
101	A.N.G.J.Bandara	1989-03-10	22/A, Kandasale, Knady	7155663217	Bandara12345@gmail.com	chef	10	100000.00	DELETE	EDIT
102	P.V.W.Silva	1993-08-11	71/B, Mahaiywa, Knady	751234569	silva12345@gmail.com	Sous Chef	6	50000.00	DELETE	EDIT

Figure 24.1 - Admin Page

## **CHAPTER – 7**

### **7.1 CONCLUSION AND FURTHER WORKS**

#### **7.1.1 CONCLUSION**

The “Unimo Family Friendly Restaurant Management System” successfully addresses the key requirements of a modern restaurant by automating processes related to order management, inventory tracking, reservations, and customer management. The system’s user-friendly interface ensures that staff can easily interact with it, leading to enhanced efficiency and reduced errors. The introduction of reporting and analytics functionalities enables data-driven decision-making, further optimizing the restaurant’s operations. Overall, the system improves customer satisfaction and operational efficiency, contributing to the restaurant’s success.

#### **7.1.2 FURTHER WORKS**

To build on the current system, several enhancements can be considered for future development:

1. **Payment Gateway Integration:**  
Incorporating online payment options to facilitate cashless transactions and improve convenience for customers.
2. **Advanced Analytics:**  
Implementing sophisticated analytics tools to provide deeper insights into sales trends, customer preferences, and inventory management.
3. **Mobile Application Development:**  
Creating a mobile app version of the system to provide customers with an easy way to make reservations, place orders, and interact with the restaurant on-the-go.
4. **Multi-branch Support:**  
Expanding the system to support multiple restaurant locations, allowing centralized management and reporting across different branches.
5. **Loyalty Programs:**

Introducing loyalty programs and personalized offers to enhance customer retention and satisfaction.

6. Integration with Third-party Services:

Connecting the system with third-party services like food delivery platforms and reservation websites to broaden the restaurant's reach.

7. Enhanced Security Features:

Implementing advanced security measures to protect sensitive customer data and ensure compliance with data protection regulations.

8. Automated Marketing Tools:

Integrating marketing automation tools to help the restaurant manage promotional campaigns and engage with customers more effectively.

## CHAPTER – 8

### 8.1 SUMMARY

This report outlines the comprehensive development process for the “Unimo Family Friendly Restaurant Management System”, created by the Group 15. The project was undertaken to address the inefficiencies and errors in the restaurant’s current manual processes, which affected order management, inventory control, and customer reservations.

The system was developed using HTML, CSS, JavaScript, and PHP to create a robust-based application. The system was implemented using a combination of front-end and back-end technologies. The primary objective was to streamline restaurant operations, improve accuracy, and enhance customer satisfaction by automating key functions.

Key components and features of this system are order management, inventory management, reservation system, customer management, reporting and analytics, and user-friendly interface.

The design phase involved creating detailed wireframes, ER diagram, context diagram, DFDs, class diagram, and use case diagram. These visual tools helped in defining the system architecture and ensuring that all components were integrated seamlessly.

The project required a server to host the application and store data, along with computers for staff use. Software requirements included a web server, database management system, and development tools like IDEs.

The “Unimo Family Friendly Restaurant Management System” successfully meets the project goals by automating and streamlining various aspects of restaurant operations. It enhances efficiency, reduces errors, and improves customer satisfaction, making it a valuable asset for Unimo Family Friendly Restaurant.

Further enhancements to the system could include integrating online payment gateways, developing a mobile application, supporting multiple branches, introducing loyalty programs, and implementing advanced security and marketing features.

This project demonstrates the effective use of modern web technologies to solve real-world problems in the hospitality industry. The collaborative effort of the team members and the guidance of the project supervisor were crucial in achieving the project objectives. The “Unimo Family Friendly Restaurant Management System” is a testament to the team’s technical skills and dedication to improving restaurant management through automation.

GROUP - 15

## CHAPTER – 9

### 9.1 REFERENCES

#### HTML (Hyper Text Markup Language)

- [1] World Wide Web Consortium (W3C). (2014). *HTML5 specification*. Retrieved from <https://www.w3.org/TR/html5/>
- [2] Mozilla Developer Network (MDN). (n.d.). *HTML*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [3] Flanagan, D. (2006). *HTML and XHTML: The definitive guide*. O'Reilly Media.

#### CSS (Cascading Style Sheets)

- [1] World Wide Web Consortium (W3C). (2021). *CSS specifications*. Retrieved from <https://www.w3.org/Style/CSS/>
- [2] Mozilla Developer Network (MDN). (n.d.). *CSS*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [3] Meyer, E. A. (2017). *CSS: The definitive guide*. O'Reilly Media.

#### JavaScript

- [1] Mozilla Developer Network (MDN). (n.d.). *JavaScript*. Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [2] Crockford, D. (2008). *JavaScript: The good parts*. O'Reilly Media.
- [3] ECMAScript. (2024). *ECMAScript language specification*. Retrieved from <https://tc39.es/ecma262/>

#### PHP (Hypertext Preprocessor)

- [1] The PHP Group. (n.d.). *PHP manual*. Retrieved from <https://www.php.net/manual/en/>
- [2] Welling, L., & Thomson, L. (2016). *PHP and MySQL web development*. Addison-Wesley.
- [3] Lockhart, J. (2015). *Modern PHP: New features and good practices*. O'Reilly Media.



## Database Management

- [1] Oracle Corporation. (n.d.). *MySQL reference manual*. Retrieved from <https://dev.mysql.com/doc/>
- [2] Davis, M. E., & Phillips, J. (2010). *Learning PHP & MySQL*. O'Reilly Media.

## Web Development Best Practices

- [1] Krug, S. (2014). *Don't make me think: A common sense approach to web usability*. New Riders.
- [2] Marcotte, E. (2014). *Responsive web design*. A Book Apart.

## Security

- [1] Open Web Application Security Project (OWASP). (2024). *OWASP top ten*. Retrieved from <https://owasp.org/www-project-top-ten/>
- [2] Hoffman, A. (2020). *Web application security: Exploitation and countermeasures for modern web applications*. No Starch Press.

## Frameworks and Libraries

- [1] Bootstrap. (n.d.). *Bootstrap documentation*. Retrieved from <https://getbootstrap.com/docs/>
- [2] jQuery Foundation. (n.d.). *jQuery API documentation*. Retrieved from <https://api.jquery.com/>

## Project Management and Development Methodologies

- [1] Cohn, M. (2005). *Agile estimating and planning*. Prentice Hall.
- [2] Hunt, A., & Thomas, D. (2019). *The pragmatic programmer: Your journey to mastery*. Addison-Wesley.

## CHAPTER – 10

### 10.1 APPENDIXES

#### 10.1.1 APPENDIX – A: SAMPLE CODE SNIPPETS

```
4      <head>
5          <!-- CSS -->
6          <style>
7              body {
8                  background-image:url('../Images/restauarent/a-table-with-food-and-drinks-on-it-ai-generated-free-photo.jpg');
9                  background-repeat:no-repeat;
10                 background-attachment: fixed;
11                 background-size:cover;
12                 background-position: center;
13             }
14         </style>
15         <link rel="stylesheet" href="./CSS/main css.css">
16
17         <!-- Fontawsome -->
18         <link rel="stylesheet" href="./CSS/fontawesome-free-6.5.1-web/css/all.css">
19
20         <!-- JS -->
21         <script src="./JS/pre loader 2.js"></script>
22         <script src="./JS/login_form - admin.js"></script>
23
24     </head>
25     <body onload="myFunction()" style="margin:0;">
26         <div style="display:none;" id="myDiv" class="animate-bottom">
27
28
29     <!-- Midflex -->
30
31
32     <div class="d17">
33
34         <div class="d26">
35             <p class="center">Admin Login</p>
36         </div>
37
38         <form name="form">
39             <div class="row2">
40                 <div class="col-25">
41                     <label for="fname">User<span style="color: rgba(189, 62, 62, 0.014);">_</span>Name</label>
42                 </div>
43                 <div class="col-75">
44                     <input type="text" id="uname" name="username" >
45                 </div>
46             </div>
47
48             <div class="row2">
49                 <div class="col-25">
50                     <label for="password">Password</label>
51                 </div>
52                 <div class="col-75">
53                     <input type="password" id="password" name="password">
54                 </div>
55             </div>
56         </form>
57     </div>
```

Figure 25.1 - Login Interface Code

```

143 <?php
144
145
146 // Username is root
147 $user = 'root';
148 $password = '';
149
150 // Database name is geeksforgeeks
151 $database = 'menunew';
152
153 // Server is localhost with
154 // port number 3306
155 $servername='localhost:3306';
156 $mysqli = new mysqli($servername, $user, $password, $database);
157
158 // Checking for connections
159 if ($mysqli->connect_error) {
160     die('Connect Error (' .
161         $mysqli->connect_errno . ') '.
162         $mysqli->connect_error);
163 }
164
165 // SQL query to select data from database
166 $sql = " SELECT * FROM menunew ";
167 $result = $mysqli->query($sql);
168 $mysqli->close();
169 ?>
170
171
172 <section>
173     <div class="d2">
174         <p class="center">Menu List</p>
175     </div>
176     <br>
177     <!-- TABLE CONSTRUCTION -->
178     <table>
179         <tr>
180             <th>Item ID</th>
181             <th>Item Category</th>
182             <th>Item Name</th>
183             <th>Item Amount</th>
184             <th>Delete</th>
185             <th>Edit</th>
186         </tr>
187     </table>
188     <!-- PHP CODE TO FETCH DATA FROM ROWS -->
189     <?php
190         // LOOP TILL END OF DATA
191         while($rows=$result->fetch_assoc())
192         {
193             ?>
194             <tr>
195                 <!-- FETCHING DATA FROM EACH
196                 ROW OF EVERY COLUMN -->

```

Figure 26.1 - Menu Page Code

```

35 }
49 <body onload="myFunction()" style="margin:0;">
50 <div style="display:none;" id="myDiv" class="animate-bottom">
71 <div class="d220">
73 <form action="insert.php" method="post">
105 </div>
106 </div>
107
108 <div class="row2">
109 <div class="col-25r">
110 <label for="lname">E<span style="color: rgba(189, 62, 62, 0.014);">_</span>mail</label>
111 </div>
112 <div class="col-75r">
113 <input type="text" name="email" id="e">
114 </div>
115 </div>
116
117 <div class="row2">
118 <div class="col-25r">
119 <label for="lname">Phone<span style="color: rgba(189, 62, 62, 0.014);">_</span>Number</label>
120 </div>
121 <div class="col-75r">
122 <input type="text" name="phone" id="p">
123 </div>
124 </div>
125
126 <div class="row2">
127 <div class="col-25r">
128 <label for="lname">Date</label>
129 </div>
130 <div class="col-75r">
131 <input type="date" name="date" id="d">
132 </div>
133 </div>
134
135 <div class="row2">
136 <div class="col-25r">
137 <label for="lname">Time</label>
138 </div>
139 <div class="col-75r">
140 <input type="time" name="time" id="t">
141 </div>
142 </div>
143
144 <div class="row2">
145 <div class="col-25r">
146 <label for="lname">Table</label>
147 </div>
148 <div class="col-75r">
149 <input type="text" name="table_id" id="tbl" placeholder="Table ID">
150 <br>
151 <br>
152 <input type="text" name="table" id="tbl" placeholder="Table Number">
153 </div>
154 </div>

```

Figure 27.1 - Reservation Page Code

```

167 "Beverages - Mango Lassi":590.00,
168 "Beverages - Vanila Milkshake":690.00,
169 "Beverages - Strawberry Milkshake":690.00,
170 "Beverages - Mango Milkshake":690.00,
171 "Beverages - Durian Milkshake":690.00,
172 "Beverages - Lemon N Ginger":350.00,
173 "Beverages - Lime n Mint":350.00,
174 "Beverages - falooda":490.00,
175 "Beverages - water 1LTR":130.00,
176 "Beverages - Soft Drink 500ML":170.00,
177 "Beverages - Soft Drink 1LTR":270.00,
178
179 "Deserts - Durian Ice Cream":490.00,
180 "Deserts - Orio Ice Cream":490.00,
181 "Deserts - Strawberry Cheesecake Ice Cream":490.00,
182 "Deserts - Vanila with Honey Ice Cream":490.00,
183 "Deserts - Chocolate Fudge Ice Cream":490.00,
184 "Deserts - Mango Ice Cream":490.00,
185 "Deserts - Butter Scotch Ice Cream":490.00,
186 "Deserts - Durian Ice Cream":490.00,
187 };
188
189 let sortedItemNames = Object.keys(itemAmounts).sort();
190
191 let itemCatInput = document.getElementById("item_cat");
192 itemCatInput.addEventListener("keyup", (e) => {
193   removeElements(itemCatInput.nextElementSibling);
194   for (let i of sortedCategories) {
195     if (i.toLowerCase().startsWith(itemCatInput.value.toLowerCase()) && itemCatInput.value != "") {
196       let listItem = document.createElement("li");
197       listItem.classList.add("list-items");
198       listItem.style.cursor = "pointer";
199       listItem.setAttribute("onclick", "displayCategory('" + i + "')");
200       let word = "<b>" + i.substr(0, itemCatInput.value.length) + "</b>";
201       word += i.substr(itemCatInput.value.length);
202       listItem.innerHTML = word;
203       itemCatInput.nextElementSibling.appendChild(listItem);
204     }
205   }
206 });
207
208 function displayCategory(value) {
209   itemCatInput.value = value;
210   removeElements(itemCatInput.nextElementSibling);
211 }
212
213 let itemNameInput = document.getElementById("item_name");
214 itemNameInput.addEventListener("keyup", (e) => {
215   removeElements(itemNameInput.nextElementSibling);
216   for (let i of sortedItemNames) {
217     if (i.toLowerCase().startsWith(itemNameInput.value.toLowerCase()) && itemNameInput.value != "") {
218       let listItem = document.createElement("li");
219       listItem.classList.add("list-items");
220       listItem.style.cursor = "pointer";
221       listItem.setAttribute("onclick", "displayItemName('" + i + "')");

```

Figure 28.1 - Orders Page Code

```

C:\xampp\htdocs> Insert - admin - Employee.php
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5  |   <title>Insert Page page</title>
6  </head>
7
8  <body>
9  |   <center>
10 |       <?php
11 |
12 |           // servername => localhost
13 |           // username => root
14 |           // password => empty
15 |           // database name => staff
16 |           $conn = mysqli_connect("localhost", "root", "", "adminemployeeenew");
17 |
18 |           // Check connection
19 |           if($conn === false){
20 |               die("ERROR: Could not connect. "
21 |               |   . mysqli_connect_error());
22 |           }
23 |
24 |           // Taking all 5 values from the form data(input)
25 |           $employee_id = $_REQUEST['employee_id'];
26 |           $full_name = $_REQUEST['full_name'];
27 |           $date_of_birth = $_REQUEST['date_of_birth'];
28 |           $address = $_REQUEST['address'];
29 |           $phone = $_REQUEST['phone'];
30 |           $e_mail = $_REQUEST['e_mail'];
31 |           $job_title = $_REQUEST['job_title'];
32 |           $hours_of_work = $_REQUEST['hours_of_work'];
33 |           $salary = $_REQUEST['salary'];
34 |
35 |           // We are going to insert the data into our reservationnew table
36 |           $sql = "INSERT INTO adminemployeeenew VALUES ('$employee_id','$full_name','$date_of_birth','$address','$phone','$e_mail','$job_title','$hours_of_work','$s
37 |
38 |           // Check if the query is successful
39 |           if(mysqli_query($conn, $sql)){
40 |               echo '<script>alert("Record stored in a database successfully !")</script>';
41 |               echo '<script>window.location.href="admin.php";</script>';
42 |           } else{
43 |               echo "ERROR: Hushi Sorry $sql. "
44 |               |   . mysqli_error($conn);
45 |           }
46 |
47 |           // Close connection
48 |           mysqli_close($conn);
49 |       ?>
50 |   </center>
51 </body>
52
53 </html>

```

Figure 29.1 - Admin Page Code



```

1  <?php
2  $conn = mysqli_connect("localhost", "root", "", "menunew");
3
4  if($conn === false){
5      die("ERROR: Could not connect. " . mysqli_connect_error());
6  }
7
8  $id = $_GET['item_id'];
9
10 $sql = "DELETE FROM menunew WHERE item_id='$id'";
11
12 if(mysqli_query($conn, $sql)){
13     echo '<script>alert("Record deleted successfully !")</script>';
14     echo '<script>window.location.href="menu.php";</script>';
15 } else{
16     echo "ERROR: Could not able to execute $sql. " . mysqli_error($conn);
17 }
18
19 mysqli_close($conn);
20 ?>

```

Figure 30.1 - Delete Menu Code

```

C: > xampp > htdocs > update_menu.php
1  php
2  <?php
3  $conn = mysqli_connect("localhost", "root", "", "menunew");
4
5  if($conn === false){
6      die("ERROR: Could not connect. " . mysqli_connect_error());
7  }
8
9  $item_id = $_POST['item_id'];
10 $item_category = $_POST['item_category'];
11 $item_name = $_POST['item_name'];
12 $item_price = $_POST['item_price'];
13
14 $sql = "UPDATE menunew SET item_category='$item_category', item_name='$item_name', item_price='$item_price' WHERE item_id='$item_id'";
15
16 if(mysqli_query($conn, $sql)){
17     echo '<script>alert("Record updated successfully !")</script>';
18     echo '<script>window.location.href="menu.php";</script>';
19 } else{
20     echo "ERROR: Could not able to execute $sql. " . mysqli_error($conn);
21 }
22
23 mysqli_close($conn);
24 ?>

```

Figure 31.1 - Update Menu Code

```

108 //
109 <center></center>
110
111 $conn_reservation = mysqli_connect("localhost", "root", "", "reservationnew");
112
113 $reservation_id = $_GET['reservation_id'];
114
115 // Fetch data from paymentsnew
116 $sql_payments = "SELECT grandtotal_amount, payment_method FROM paymentsnew WHERE reservation_id = '$reservation_id'";
117 $result_payments = mysqli_query($conn_payments, $sql_payments);
118 $data_payments = mysqli_fetch_assoc($result_payments);
119
120 // Fetch data from ordersnew
121 $sql_orders = "SELECT * FROM ordersnew WHERE reservation_id = '$reservation_id'";
122 $result_orders = mysqli_query($conn_orders, $sql_orders);
123 $data_orders = mysqli_fetch_all($result_orders, MYSQLI_ASSOC);
124
125 // Fetch data from reservationnew
126 $sql_reservation = "SELECT first_name, last_name, numberofguests, date, time, spacialoccasion FROM reservationnew WHERE id = '$reservation_id'";
127 $result_reservation = mysqli_query($conn_reservation, $sql_reservation);
128 $data_reservation = mysqli_fetch_assoc($result_reservation);
129
130 // Output the HTML with the fetched data
131
132 echo "<h1>Unimo Family Restaurant</h1>";
133 echo "<h3>Bill Summary</h3>";
134
135 echo "<div class='section'>";
136 echo "<h2>Reservation Details</h2>";
137 echo "<table>";
138 echo "<tr><th>First Name</th><th>Last Name</th><th>Number of Guests</th><th>Date</th><th>Time</th><th>Special Occasion</th></tr>";
139 echo "<tr><td>{$data_reservation['first_name']}</td><td>{$data_reservation['last_name']}</td><td>{$data_reservation['numberofguests']}</td><td>{$data_reservation['date']}</td><td>{$data_reservation['time']}</td><td>{$data_reservation['spacialoccasion']}</td></tr>";
140 echo "</table>";
141 echo "</div>";
142
143 echo "<div class='section'>";
144 echo "<h2>Orders Details</h2>";
145 echo "<table>";
146 echo "<tr><th>Reservation ID</th><th>Order ID</th><th>Table Number</th><th>Item Ordered</th><th>Item Amount</th><th>Quantity</th><th>Total Amount</th></tr>";
147 foreach ($data_orders as $order) {
148     echo "<tr><td>{$order['reservation_id']}</td><td>{$order['order_id']}</td><td>{$order['table']}</td><td>{$order['item_ored_name']}</td><td>{$order['item_ored_amount']}</td><td>{$order['quantity']}</td><td>{$order['total_amount']}</td></tr>";
149 }
150 echo "</table>";
151 echo "</div>";
152
153 echo "<div class='section'>";
154 echo "<h2>Payments Details</h2>";
155 echo "<table>";
156 echo "<tr><th>Grand Total Amount</th><th>Payment Method</th></tr>";
157 echo "<tr><td>{$data_payments['grandtotal_amount']}</td><td>{$data_payments['payment_method']}</td></tr>";
158 echo "</table>";
159 echo "</div>";
160
161 mysqli_close($conn_payments);
162 mysqli_close($conn_orders);
163 mysqli_close($conn_reservation);
164
165 //

```

Figure 32.1 - Print Bill Code

## 10.1.2 APPENDIX – B: USER MANUAL

The user manual provides detailed instructions for using the system, including how to navigate the various pages and perform key tasks such as placing orders, making reservations, and viewing reports.

- **Home Page:**

Navigate the home page by using the navigation bar to access different sections of the website such as the menu, reservations, and contact information.

- **Reservation Page:**

Fill out the reservation form by selecting the desired date, time, number of guests, and any special requests. Click the "Submit" button to confirm the reservation.

- **Menu Page:**



Browse the menu items categorized by sections. Click on the "Add to Cart" button to add items to your order. Use the search bar to quickly find specific items.

- **User Login:**

Enter your username and password in the login fields and click the "Login" button to access your account.

- **Orders:**

View the available menu items, customize your order if necessary, and add items to your cart. Proceed to the checkout page to complete your order.

- **Payments:**

Enter your payment details, select your preferred payment method, and click the "Submit Payment" button to finalize your order.

- **Reports:**

Access the reports page to view sales, inventory, and customer data. Use the export options to download reports for further analysis.

- **Admin Functions:**

Log in as an administrator to manage employees, view system reports, and configure settings. Use the admin page to add, edit, or delete employee records.

### 10.1.3 APPENDIX – C: PROJECT PLAN TIMELINE

Week 1	Submitting the proposal
Week 2	Discussion with the supervisor
Week 3	Gathering client requirements
Week 4	Analysis Process
Week 5	System Design
Week 6 – 10	Implementation
Week 11	Testing
Week 12	Deployment

Table 1.3 - Project Plan Timeline