

# **OCEAN BREEZE RESTAURANT**

A Mini Project Report

Submitted for the partial fulfilment for the award of degree of

## **BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

Submitted by

**YOGESH R**

[Reg. No. A22CSDB49]

**SARAVANAVEL R**

[Reg. No. A22CSDB50]

**Under the Guidance of**

**Mr. ANAND RAJ I, M.C.A**

**(Assistant Professor, PG and Research Department of Computer Science & Artificial Intelligence)**



**PG and Research Department of Computer Science & Artificial Intelligence**

**ST. JOSEPH'S COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS)**

**CUDDALORE - 607001**

**APRIL - 2025**

## **CERTIFICATE**

This is to certify that the Mini-Project entitled

### **OCEAN BREEZE RESTAURANT**

Being submitted to

**ST. JOSEPH'S COLLEGE OF ARTS AND SCIENCE (AUTONOMOUS),**

**(Affiliated to Annamalai University-Annamalai Nagar, Chidambaram)**

Submitted By

**YOGESH R**

**[Reg. No. A22CSDB49]**

**SARAVANAVEL R**

**[Reg. No. A22CSDB50]**

For the partial fulfillment for the award of degree of

**BACHELOR OF SCIENCE IN COMPUTER SCIENCE**

Is a Bonafide record of work carried out by them,

Under my guidance and supervision.

**INTERNAL GUIDE**

**HEAD OF THE DEPARTMENT**

Submitted for the viva-voce examination held on -----

**EXAMINERS:**

1. \_\_\_\_\_

2. \_\_\_\_\_

## **ACKNOWLEDGEMENT**

It is our earnest and sincere desire and ambition to acquire profound knowledge in the study of Bachelor of science in Computer Science. We are grateful to God, the Almighty who has blessed us abundantly and guided us to complete this task.

We express our sincere thanks to our beloved, **Rev. Fr. M. SWAMINATHAN., D.C.L., D.JURIS., Secretary**, St. Joseph's College of Arts and Science (Autonomous), Cuddalore, for providing such a good congenial environment to enlighten my knowledge.

We express our sincere gratitude to **Dr. M. ARUMAI SELVAM, M.Sc., M.Phil., Ph.D., Principal** and Head, Post Graduate and Research Department of Computer Science and Artificial Intelligence, St. Joseph's College of Arts and Science (Autonomous), Cuddalore, for his support and encouragement

We take immense pleasure in conveying our sincere and heartfelt gratitude to our internal guide **Mr. ANAND RAJ I, M.C.A , Assistant Professor** for his valuable guidance, inspiration, consent and motivation throughout our project. His untiring encouragement made our thoughts enthusiastic in successfully completing this project.

At the Outset, we extend our sincere gratitude to all our teachers in the Department of Computer Science department who have taken pains to shape and mould us in all our endeavors. We will fail in our duty if we do not put on record our heartfelt gratitude to our beloved parents for their financial and material support as well their words of encouragement and motivation to complete this project successfully.

Last but not the Least, we place our deepest sense of gratitude to all our Friends who had inspired and supported us to complete this project more successfully.

**YOGESH R**  
[Reg No. A22CSDB49]  
**&**  
**SARAVANAVEL R**  
[Reg No. A22CSDB50]

## **ABSTRACT**

The Mini-Project titled “Ocean Breeze Restaurant” is a web-based application designed to enhance the online dining experience. This platform is ideal for food enthusiasts who are looking for delicious, fresh seafood, meats, and beverages but may not have the time to visit the restaurant in person. The web application enables users to explore the restaurant’s extensive menu, place orders, and have their meals delivered directly to their doorsteps, ensuring convenience and a seamless dining experience. The application allows customers to view the full menu, which includes fresh seafood, premium meats, and refreshing drinks, all curated for an exceptional dining experience. The platform also offers an easy-to-use online ordering system where customers can select their meals, customize their orders, and choose from various payment options, including cash on delivery. The restaurant is committed to providing the fastest delivery service, ensuring that every order is delivered hot and fresh. Additionally, the app includes a seat reservation feature, allowing customers to book a table in advance. Users can select the desired date, time, and number of seats, ensuring they have a guaranteed spot when they arrive at the restaurant. This feature is especially useful during peak hours or special events, providing a hassle-free dining experience. With this web application, Ocean Breeze Restaurant aims to bring the taste of the ocean directly to your home, offering a fast, reliable, and enjoyable dining experience with just a few clicks.

## TABLE OF CONTENTS

<b>CERTIFICATE</b>	<b>i</b>
<b>ACKNOWLEDGEMENT</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>

SL. NO	CONTENT	PAGE NO.
1.	<b>INTRODUCTION</b> 1.1 ABOUT THE PROJECT	1
2.	<b>PROBLEM DEFINITION</b> 2.1 EXISTING SYSTEM 2.2 LIMITATIONS OF EXISTING SYSTEM 2.3 PROPOSED SYSTEM 2.4 FEATURES OF PROPOSED SYSTEM	2
3.	<b>SOFTWARE REQUIREMENTS SPECIFICATION</b> 3.1 INTRODUCTION 3.2 MODULE DESCRIPTION 3.3 HARDWARE REQUIREMENTS 3.4 SOFTWARE REQUIREMENTS	4
4.	<b>SYSTEM ANALYSIS</b> 4.1 INTRODUCTION 4.2 WEBSITE ARCHITECTURE	7
5.	<b>SYSTEM DESIGN</b> 5.1 INTRODUCTION 5.2 USER INTERFACE DESIGN 5.3 DATABASE DESIGN	9
6.	<b>SYSTEM TESTING</b> 6.1 INTRODUCTION 6.2 UNIT TESTING 6.3 INTEGRATION TESTING 6.4 VALIDATION TESTING	15

	6.5 ALPHA AND BETA TESTING 6.6 SYSTEM TESTING	
7.	<b>CONCLUSION</b> 7.1 CONCLUSION 7.2 FUTURE ENHANCEMENTS	17
8.	<b>BIBLIOGRAPHY</b> 8.1 BOOKS 8.2 WEBSITE	18
	<b>APPENDICES</b> APPENDIX-A SOFTWARE DESCRIPTION APPENDIX-B SAMPLE SOURCE CODE APPENDIX-C SCREEN SHOTS	19 28 71

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 ABOUT THE PROJECT**

The main objective of the Ocean Breeze Restaurant web application is to provide a seamless and enhanced dining experience for customers by offering both online ordering and seat reservation services. This project aims to make it easier for customers to explore the restaurant's menu, place orders for delivery, and reserve a table for inhouse dining, all from the convenience of their own devices.

The application features a seat reservation system, allowing customers to book a table in advance for a hassle-free dining experience. Users can choose the date, time, and number of seats required, ensuring they have a reserved spot when they arrive. This is particularly useful during busy hours, events, or special occasions, helping customers avoid wait times.

Additionally, the platform offers a user-friendly interface for customers to view the menu, customize their orders, and choose from various payment methods, including cash on delivery. The restaurant is dedicated to providing fast and reliable delivery services, ensuring that each order arrives hot and fresh.

In the future, the application will expand to include online payment options, further enhancing the convenience and ease of use for customers. The Ocean Breeze Restaurant web application aims to revolutionize the dining experience by combining exceptional service, convenience, and efficiency.

## CHAPTER 2

### PROBLEM DEFINITION

#### **2.1 EXISTING SYSTEM**

Currently, the Ocean Breeze Restaurant relies on traditional methods for managing customer orders, reservations, and in-house dining experiences. The process is largely manual, requiring customers to either call in or visit the restaurant to make a reservation or place an order. This can be time-consuming for both customers and staff, particularly during peak hours or busy seasons.

For reservations, customers need to contact the restaurant directly by phone or in person to book a table, often resulting in potential errors or misunderstandings regarding availability. There is no real-time access to table availability, causing delays or overbooked situations.

For ordering, customers typically either visit the restaurant to dine in or place an order over the phone for home delivery. The ordering process is not streamlined and often lacks detailed customization options for customers. Additionally, payment for orders is generally done upon delivery, using cash on delivery, which can be an inconvenience for both the restaurant and the customer.

#### **2.2 LIMITATIONS OF EXISTING SYSTEM**

- **Manual Reservation Process:** The current system requires customers to call or visit the restaurant to make reservations, leading to possible errors in availability, doublebooking, or missed reservations. It lacks real-time booking and availability checking, causing delays or frustration for customers.
- **Limited Ordering Options:** Customers who wish to place orders for delivery must do so by phone or in person, which can be inconvenient and time-consuming. The ordering process lacks customization and flexibility, limiting customers' ability to choose their preferred meals or modify orders easily.
- **No Online Payment Integration:** Currently, the system only supports cash

on delivery, which can be inconvenient for customers and may cause delays in order processing. There is no option for secure online payments, making the ordering process less efficient.

## 2.3 PROPOSED SYSTEM

The proposed Ocean Breeze Restaurant web application aims to address the limitations of the existing system by implementing a comprehensive, user-friendly, and efficient digital platform. This system will enhance both customer experience and operational efficiency through the following.

This project consists of the following modules

- Login/ Signup
- Home
- Reservation
- Menu
- Cart
- My profile
- My Orders
- My reservation

## 2.4 FEATURES OF THE PROPOSED SYSTEM

- **Online Seat Reservation:** Customers can easily make real-time seat reservations through the web application. They can choose the date, time, and number of seats, ensuring a seamless booking experience with no overbooking issues.
- **Online Ordering System:** The system allows customers to view the full menu, customize their orders, and place them directly from the web platform. Users can add special requests and select their preferred ingredients for a personalized meal.
- **Multiple Payment Options:** The application will offer various payment methods, including secure online payments via credit/debit cards and digital wallets. This removes the reliance on cash on delivery, simplifying the payment process.

# **CHAPTER 3**

## **SOFTWARE REQUIREMENT SPECIFICATION**

### **3.1 INTRODUCTION**

The production of the requirements stage of the software development process is Software Requirements Specifications (SRS) (also called a requirements document). This report lays a foundation for software engineering activities. SRS is a formal report, which acts as a representation of software that enables the customers to review whether SRS is according to their requirements. Also, it comprises user requirements for a system as well as detailed specifications of the system requirements.

The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment. First, the SRS could be written by the client of a system. Second, the SRS could be written by a developer of the system.

### **3.2 MODULE DESCRIPTION**

Ocean Breeze project consists of different modules. The requirements of each module are described as follows.

- Login/ Signup
- Home
- Reservation
- Menu
- Cart
- My profile
- My Orders
- My reservation

### **3.2.1 LOGIN/SIGN UP MODULE**

This Module enables the user to sign up with a new mail ID or log in using a existing mail ID with the system using standard login features.

### **3.2.2 HOME**

This home module is used to display the details about the “Ocean Breeze” website and it navigates the users to use the website and shows the facilities provided by the website.

### **3.2.3 RESERVATION**

The Reservation Module contains a system for customers to select dates, times, and seat numbers for table bookings. It also includes real-time availability, confirmation notifications, and an admin panel for managing reservations.

### **3.2.4 MENU**

The Menu Module contains a categorized list of food and drink offerings with detailed descriptions, pricing, and images. It also includes customization options, allergy information, and search filters for easy navigation.

### **3.2.5 CART**

The Cart module allows customers to add, view, update, and remove items before proceeding to checkout. It ensures a seamless ordering experience by displaying item details, quantities, and prices in real time.

### **3.2.6 MY PROFILE**

The My Profile module enables users to view and manage their personal information, such as name, contact details, preferences, and order history. It allows for profile updates, password changes, and personalized settings for a seamless user experience.

### **3.2.7 MY ORDERS**

The My Orders module allows users to view their past and current orders, track order status, and access order details, including invoices and payment history. It enhances the user experience by providing easy order management and reordering options.

### **3.2.8 MY RESERVATION**

The My Reservation module enables users to view, manage, and cancel their table reservations. It provides details such as reservation date, time, guest count, and status, ensuring a smooth booking experience.

## **3.3 HARDWARE REQUIREMENTS**

PERIPHERAL	SPECIFICATION
MONITOR	HD LED
SCREEN RESOLUTION	1980 x 1080
KEYBOARD	STANDARD
HARDDISK	512 GB
MOUSE	TOUCHPAD

## **3.4 SOFTWARE REQUIREMENTS**

COMPONENT	TOOLS OR APPLICATION
TECHNOLOGY	WEB APPLICATION
FRONT-END APPLICATION	HTML, CSS, JS
BACK-END DATABASE	MYSQL SERVER
SERVER	PHP, PHP SERVER

# CHAPTER 4

## SYSTEM ANALYSIS

### 4.1 INTRODUCTION

System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. It is conducted for the purpose of studying a system in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

### 4.2 WEBSITE ARCHITECTURE DIAGRAM

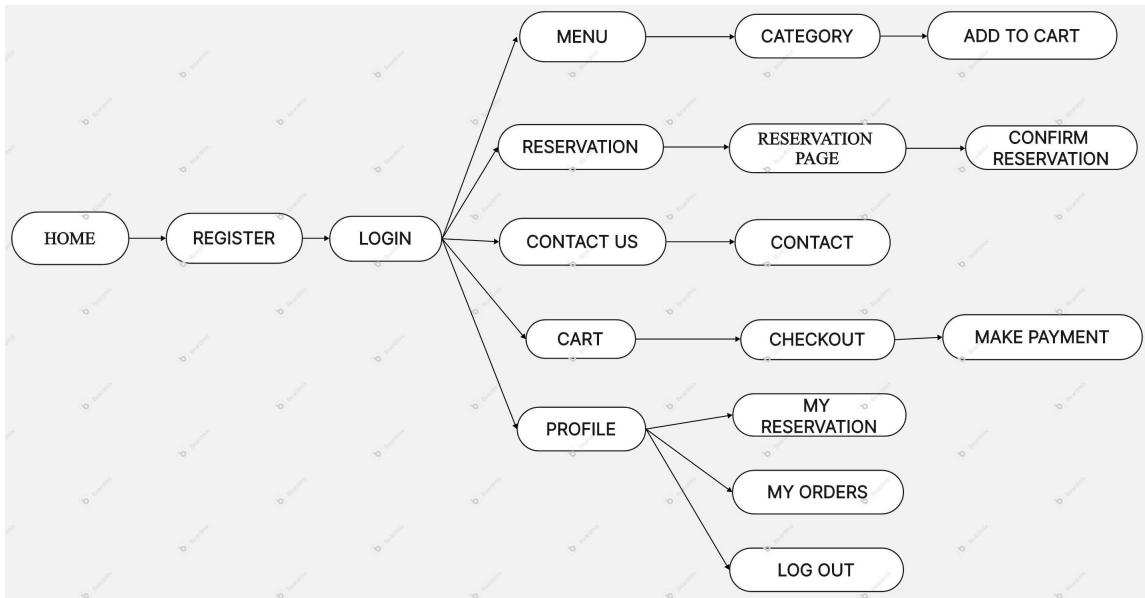
The website architecture diagram illustrates the different pages and sections of the **Ocean Breeze Restaurant** website, as well as how they are interconnected. This diagram helps in understanding the website structure and how users navigate through various sections. The diagram specifically represents an **Ocean Breeze Restaurant** website, showcasing its key sections:

- The **Home Page** connects to the **Menu Page, Reservations Page, About Us Page, and Contact Page**.
- The **Menu Page** provides access to different **South Indian Food Categories** (e.g., Meals, Chetinadu Delights, Biriyani) and individual **Dish Detail Pages**.
- The **Reservations Page** allows users to **book a table** and connects to the **Login/Signup Page** for registered users.
- The **Contact Page** includes a **Contact Form**, and **Customer Support Details**.
- The **Order Online Page** connects to the **Shopping Cart Page**, where customers can review their orders.

The **Shopping Cart Page** links to the **Checkout Page**, where users complete their payment and finalize their orders.

- The **Admin Panel** allows restaurant staff to manage **reservations, menu items, orders, and customer queries**.

# WEBSITE ARCHITECTURE OF OVEAN BREEZE RESTAURANT



# **CHAPTER 5**

## **SYSTEM DESIGN**

### **5.1 INTRODUCTION**

Oceanbreeze has been designed as an web application to shop various products seamlessly. Hence it's given careful analysis and after successful analysis and refinement, it's moved to the design stage. The design stage focuses on the need for the following component of the Application design for an organization.

- The Application User Interface design
- The Database design for accessing data

### **5.2 USER INTERFACE DESIGN**

"Ocean Breeze Restaurant" is an official application designed to enhance the dining experience for customers. It aims to provide a seamless, user-friendly interface that addresses all the essential needs of those who use the system. The app focuses on making it easier for customers to explore the menu, place orders, and access special offers while ensuring a smooth, enjoyable, and efficient restaurant experience.

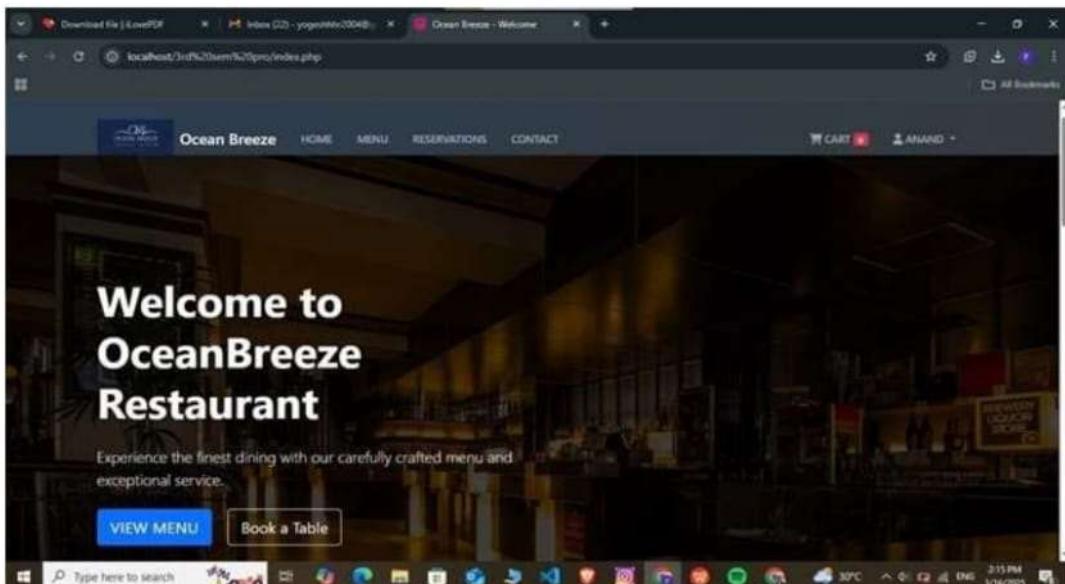
#### **5.2.1 STATIC PAGES**

An Ocean Breeze Restaurant static web page (sometimes referred to as a flat or stationary page) is a web page that is delivered to the user exactly as it is stored, unlike dynamic web pages that are generated by a web application. As a result, a static web page for the restaurant shows the same content to all users, regardless of context, unless configured to negotiate content-type or language variations by the web server.

Static web pages are often HTML documents stored as files in the file system and made available by the web server over HTTP. While URLs ending in ".php" are not necessarily static, static pages can also include those stored in a database or formatted using a template served through an application server, as long as the page is unchanging and served as stored.

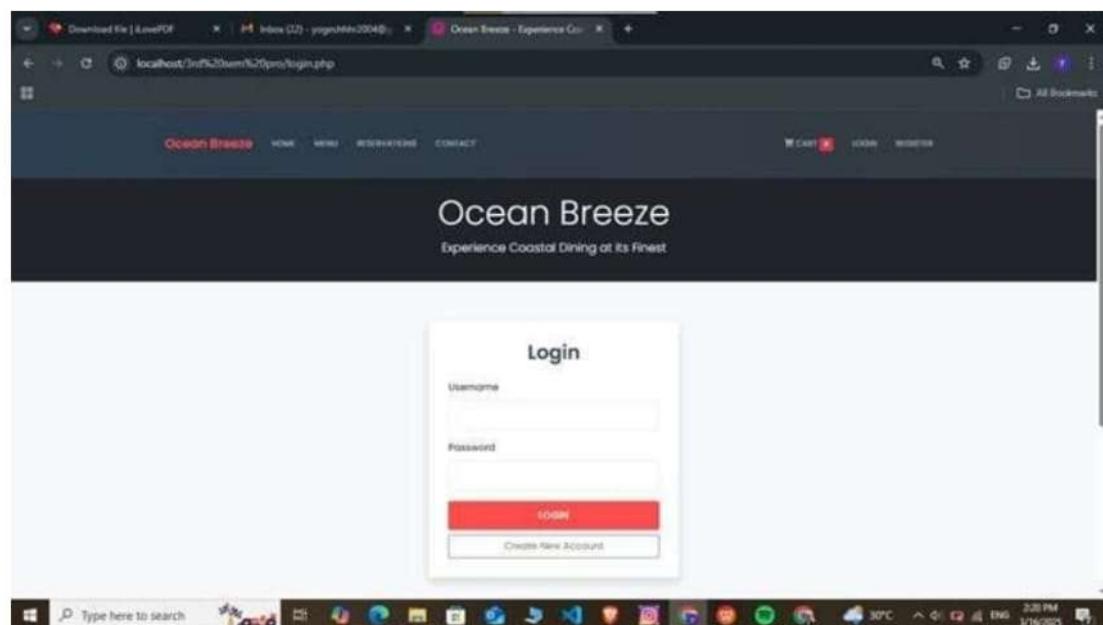
The static pages used on the Ocean Breeze Restaurant website are listed below:

A Sample design of the Static Pages is given below:

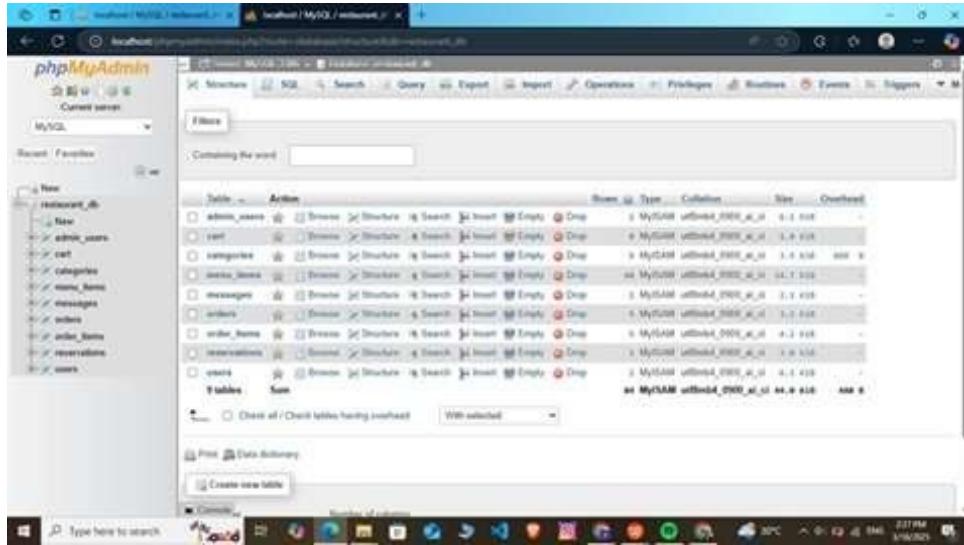


### 5.2.2 DYNAMIC PAGES

A dynamic web page for the Ocean Breeze Restaurant displays different content based on user interaction and session. For example, the login page changes depending on the user's status. A returning customer might see a greeting like "Welcome back, [Username]" and options to view past orders, while a new user would see a login form with a "Sign Up" option. If incorrect credentials are entered, the page dynamically shows an error message, like "Incorrect username or password," prompting the user to try again. This makes the page adapt and provide a personalized experience.



## 5.3 DATABASE DESIGN



### 5.3.1 DATABASE NAME : restaurant\_db

The Following Tables were created.

#### 5.3.1.1 TABLE NAME : Admin\_users

Field Name	Data Type
ID	INT
USERNAME	VARCHAR
PASSWORD	VARCHAR
FULL_NAME	VARCHAR
EMAIL	VARCHAR
CREATED_AT	TIMESTAMP

### **5.3.1.2 TABLE NAME : Cart**

<b>Field Name</b>	<b>Data Type</b>
ID	INT
USER_ID	INT
MENU_ITEM_ID	INT
QUANTITY	INT
CREATED_AT	TIMESTAMP

### **5.3.1.3 TABLE NAME : Categories**

<b>Field Name</b>	<b>Data Type</b>
ID	INT
NAME	VARCHAR
DESCRIPTION	TEXT
CREATED_ID	TIMESTAMP

### **5.3.1.4 TABLE NAME : Menu\_items**

<b>Field Name</b>	<b>Data Type</b>
ID	INT
CATEGORY_ID	INT
NAME	VARCHAR
DESCRIPTION	TEXT
PRICE	DECIMAL
IMAGE	VARCHAR
IS_AVAILABLE	TINYINT
CREATED_AT	TIMESTAMP

### **5.3.1.5 TABLE NAME : Messages**

<b>Field Name</b>	<b>Data Type</b>
ID	INT
NAME	VARCHAR
EMAIL	VARCHAR
SUBJECT	VARCHAR
MESSAGE	TEXT
STATUS	ENUM
CREATED_AT	TIMESTAMP

### **5.3.1.6 TABLE NAME : Orders**

<b>Field Name</b>	<b>Data Type</b>
ID	INT
USER_ID	INT
TOTAL_AMOUNT	DECIMAL
STATUS	ENUM
PAYMENT_STATUS	ENUM
PAYMENT_METHOD	VARCHAR
DELIVERY_ADDRESS	TEXT
SPECIAL_INSTRUCTION	TEXT
CREATED_AT	TIMESTAMP

### **5.3.1.7 TABLE NAME : Order\_items**

<b>Field Name</b>	<b>Data Type</b>
ID	INT
ORDER_ID	INT
MENU_ITEM_ID	INT
QUANTITY	INT
UNIT_PRICE	DECIMAL
SUBTOTAL	DECIMAL

CREATED_AT	TIMESTAMP
------------	-----------

### 5.3.1.8 TABLE NAME : Reservation

Field Name	Data Type
ID	INT
USER_ID	INT
RESERVATION_DATE	DATE
RESERVATION_TIME	TIME
NUM_GUEST	INT
STATUS	ENUM
SPECIAL_REQUEST	TEXT
CREATED_AT	TIMESTAMP

### 5.3.1.9 TABLE NAME : User

Field Name	Data Type
ID	INT
USERNAME	VARCHAR
PASSWORD	VARCHAR
EMAIL	VARCHAR
FULL_NAME	VARCHAR
PHONE	VARCHAR
ADDRESS	TEXT
ROLE	ENUM
CREATED_AT	TIMESTAMP

# **CHAPTER 6**

## **SYSTEM TESTING**

### **6.1 INTRODUCTION**

Testing is the process by which developer will generate a set of test data, which gives maximum probability of finding all types of errors that can occur in the software. Testing is an important phase in the software development cycle. For any software that is newly developed, primary importance has been given to testing of the system. It is the responsibility of the developer to detect all possible errors in the software before handing it to the user or Customer.

Being a Web Application, “OCEANBREEZE” has to be undergone with the following tests:

### **6.2 UNIT TESTING**

Unit testing focuses on the smallest unit of Software component or module. For each module interface, local data structure, boundary condition, independent paths and all error handlings paths are tested. In this project, there are different modules like Sign Up, Login , admin module and so on. Each one is tested separately and the errors are rectified. For example, in a registration form each of the element (for example a text box, list box, radio button and other elements) should be tested.

### **6.3 INTEGRATION TESTING**

Integration testing is a systematic technique conducted to test errors associated with interfacing. The program is constructed and tested in a small increment. In this project, login module and registration module are integrated and tested. Similarly, all the modules are integrated and tested.

#### **6.3.1 REGRESSION TESTING**

After the successful integration of modules, it also check for any side effects that may lead to introduction of any new error in different modules.

## **6.4 VALIDATION TESTING**

Validation testing begins at the culmination (climax) of integration testing. It focuses on user-visible actions and user-recognizable output from the system. It tries to uncover errors, but the focus is at the requirements level.

## **6.5 ALPHA AND BETA TESTING**

The alpha test is conducted at the developer's site by the customer. It is conducted in a controlled environment.

The beta test is conducted at one or more end-user sites. Unlike alpha testing, the developer generally is not present. Therefore, the beta test is a "live" application of the software in an environment that cannot be controlled by the developer.

The customer records all problems (real or imagined) that are encountered during beta testing and reports these to the developer at regular intervals.

## **6.6 SYSTEM TESTING**

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system.

# **CHAPTER 7**

## **CONCLUSION & FUTURE ENHANCEMENTS**

### **7.1 CONCLUSION**

The Ocean Breeze Restaurant web application has been successfully analyzed, designed, and developed with great attention to detail. Built as a web-based platform, it ensures accessibility, convenience, and a seamless experience for users. After undergoing a thorough testing phase, the application is now live for public use, delivering a smooth and efficient interface for customers to explore the restaurant's offerings.

This project was developed as part of a mini-project, so the initial version includes only a select number of features necessary for core functionality. Despite this, the application lays a strong foundation for future enhancements. Upcoming updates may introduce new features such as online reservations, an interactive menu with detailed descriptions, customer reviews, and a loyalty rewards system. Additionally, improvements in design, performance, and personalization will further elevate user satisfaction. With a commitment to continuous development, the Ocean Breeze Restaurant platform aims to evolve into a comprehensive digital experience, making it easier for customers to connect with the restaurant, explore its delicious offerings, and enjoy an enhanced dining experience.

### **7.2 FUTURE ENHANCEMENTS**

Future enhancements for the Ocean Breeze Restaurant web application will focus on improving user convenience and engagement. Planned updates include real-time seat availability, allowing customers to view and reserve tables instantly. An interactive menu with images, descriptions, and prices will enhance the browsing experience. Customer reviews and ratings will help diners make informed choices. A loyalty rewards program will encourage repeat visits, while personalized recommendations based on order history will enhance user satisfaction. Additionally, multilingual support, social media integration, and performance optimizations will make the platform more accessible, engaging, and user-friendly for a wider audience.

# **CHAPTER 8**

## **BIBLIOGRAPHY**

### **8.1 BOOKS PHP**

- PHP for absolute beginners – Jason Lengstorf – Apress Publications
  - Distributed by Springer Online
- Build Your own Database driven Application using PHP and MYSQL – Kevin Yank – 3 rd edition – Sitepoint publications
- Deitel & Deitel - “Internet and WWW – How to program? Pearson Education, 2005 15 Core PHP Programming – Leo Atkinson – II Edition – PHI Edition
- PHP and MySQL – Vikram Vaswami – Mc Graw Hill
- 17 MySQL/PHP Database Applications – Brad Bulger, Jay Greenspan, David Wall – Second Edition – Wiley Publication.
- 18 Teach yourself PHP within 24 Hours – SAMS Publication

### **8.2 WEBSITES**

- [www.w3schools.com](http://www.w3schools.com)
- [phpmaster.com/](http://phpmaster.com/)
- [php.net/manual/en/getting-started.php](http://php.net/manual/en/getting-started.php)
- [www.learnphptutorial.com/](http://www.learnphptutorial.com/)
- [www.learnphp.in/](http://www.learnphp.in/)

## APPENDIX A

### SOFTWARE DESCRIPTION

#### 1. PHP

PHP stands for Hypertext Preprocessor and is a server-side language. This means that the script is run on your web server, not on the user's browser, so you do not need to worry about compatibility issues. PHP is relatively new (compared to languages such as Perl (CGI) and Java) but is quickly becoming one of the most popular scripting languages on the internet. What distinguishes PHP from something like clientside JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

You may be wondering why you should choose PHP over other languages such as Perl or even why you should learn a scripting language at all. I will deal with learning scripting languages first. Learning a scripting language, or even understanding one, can open up huge new possibilities for your website. Although you can download premade scripts from sites like Hotscripts, these will often contain advertising for the author or will not do exactly what you want. With an understanding of a scripting language you can easily edit these scripts.

Using scripts on your website allows you to add many new 'interactive' features like feedback forms, guestbooks, message boards, counters and even more advanced features like portal systems, content management, advertising managers etc. With these sorts of things on your website you will find that it gives a more professional image. As well as this, anyone wanting to work in the site development industry will find that it is much easier to get a job if they know a scripting language.

As mentioned earlier, PHP is a server-side scripting language. This means that, although your users will not need to install new software, your web host will need to have PHP set up on their server. It should be listed as part of your package

but if you don't know if it is installed you can find out using the first script in this tutorial. If your server does not support PHP you can ask your web host to install it for you as it is free to download and install. If you need a low cost web host which supports PHP I would recommend Host Rocket

Writing PHP on your computer is actually very simple. You don't need any special software, except for a text editor (like Notepad in Windows). Run this and you are ready to write your first PHP script.

The first PHP script you will be writing is very basic. All it will do is print out all the information about PHP on your server. Type the following code into your text editor:

```
<?
    Phpinfo()
?>
```

As you can see this is actually just one line of code. It is a standard PHP function called `phpinfo` which will tell the server to print out a standard table of information giving you information on the setup of the server.

## 1.1 PHP WITH HTML FORMS

If you know a little HTML, then you know that the FORM tags can be used to interact with your users. Things that can be added to a form are the likes of text boxes, radio buttons, and check boxes, drop down lists, text areas, and submit buttons. A basic HTML form with a textbox and a Submit button looks like this:

```
<html>
    <head>
        <title>A BASIC HTML FORM</title>
    </head>
    <body>
        <FORM NAME ="form1" METHOD =" " ACTION = "">
            <INPUT TYPE = "TEXT" VALUE
```

```

        ="username"> 21

<INPUT TYPE = "Submit" Name = "Submit1" VALUE = "Login">
</FORM>
</body>
</html>

```

Some familiarity with the above is assumed. But we'll discuss the METHOD, ACTION and SUBMIT attributes in the form above, because they are important.

The above form can be found in the files you download. It's in the scripts folder, and is called bas©icForm.php. Use it as a template, if you like.

So, create the form above. Save your work as basicForm.php. (This name will be VERY important!) Start your server, and make sure the form loads ok in your browser. You should be able to see a text box and a Submit button. Here's what it should look like:



If a user comes to your site and has to login, for example, then you'll need to get the details from textboxes. Once you get the text that the user entered, you then test it against a list of your users (this list is usually stored on a database, which we'll see how to code for in a later section). First, you need to know about the HTML attributes METHOD, ACTION and SUBMIT. We'll explore these in the next few sections.

If you look at the first line of our form from the previous page, you'll notice a METHOD attribute:

The Method attribute is used to tell the browser how the form information should

be sent. The two most popular methods you can use are GET and POST. But our METHOD is blank. So change it to this:

To see what effect using GET has, save your work again and then click the Submit button on your form. You should see this:

The thing to notice here is the address bar. After basicForm.php, we have the following:



The thing to notice here is the address bar. After basicForm.php, we have the following:

**? Submit1=Login**

This is a consequence of using the GET method. The data from the form ends up in the address bar. You'll see a question mark, followed by form data. In the image above, Submit1 was the NAME of the button, and Login was the VALUE of the button (the text on the button). This is what is being returned by the GET method. You use the GET method when the data you want returned is not crucial information that needs protecting.

## 1.2 MYSQL

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching, and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

So now a days we use relational database management systems (RDBMS) to store and manager huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A Relational Database Management System (RDBMS) is software that:

- Enables you to implement a database with tables, columns, and indexes.
- Guarantees the Referential Integrity between rows of various tables.
- Updates the indexes automatically
  - Interprets an SQL query and combines information from various tables.

MySQL is a fast, easy-to-use RDBMS used being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.

- MySQL is released under an open-source license. So you have nothing to pay to use it.
  - MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
  - MySQL uses a standard form of the well-known SQL data language.
  - MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA etc.
  - MySQL works very quickly and works well even with large data sets.
  - MySQL is very friendly to PHP, the most appreciated language for web development.
  - MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
  - MySQL is customizable. The open source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

You can establish MySQL database using mysql binary at command prompt.

Example:

Here is a simple example to connect to MySQL server from command prompt:

```
[root@host]#mysql -u root -p
```

Enter password:\*\*\*\*\*

This will give you mysql> command prompt where you will be able to execute any SQL command.

Following is the result of above command:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 2854760 to server version: 5.0.9

Type 'help;' or 'h' for help. Type '\c' to clear the buffer.

In above example we have used root as a user but you can use any other user. Any user will be able to perform all the SQL operation which are allowed to that user.

You can disconnect from MySQL database any time using exit command at mysql> prompt. mysql>

Exit

Bye

### **MySQL Connection using PHP Script:**

PHP provides mysql\_connect() function to open a database connection. This function takes five parameters and returns a MySQL link identifier on success, or FALSE on failure.

#### **Syntax:**

```
connection mysql_connect(server,user,passwd,new_link,client_flag);
```

Parameter	Description
server	Optional - The host name running database server. If not specified then default value is localhost:3036.
user	Optional - The username accessing the database. If not specified then default is the name of the user that owns the server process.
passwd	Optional - The password of the user accessing the database. If not specified then default is an empty password.

You can disconnect from MySQL database anytime using another PHP function `mysql_close()`. This function takes a single parameter which is a connection returned by `mysql_connect()` function

**Syntax:**

```
bool mysql_close( resource $link_identifier );
```

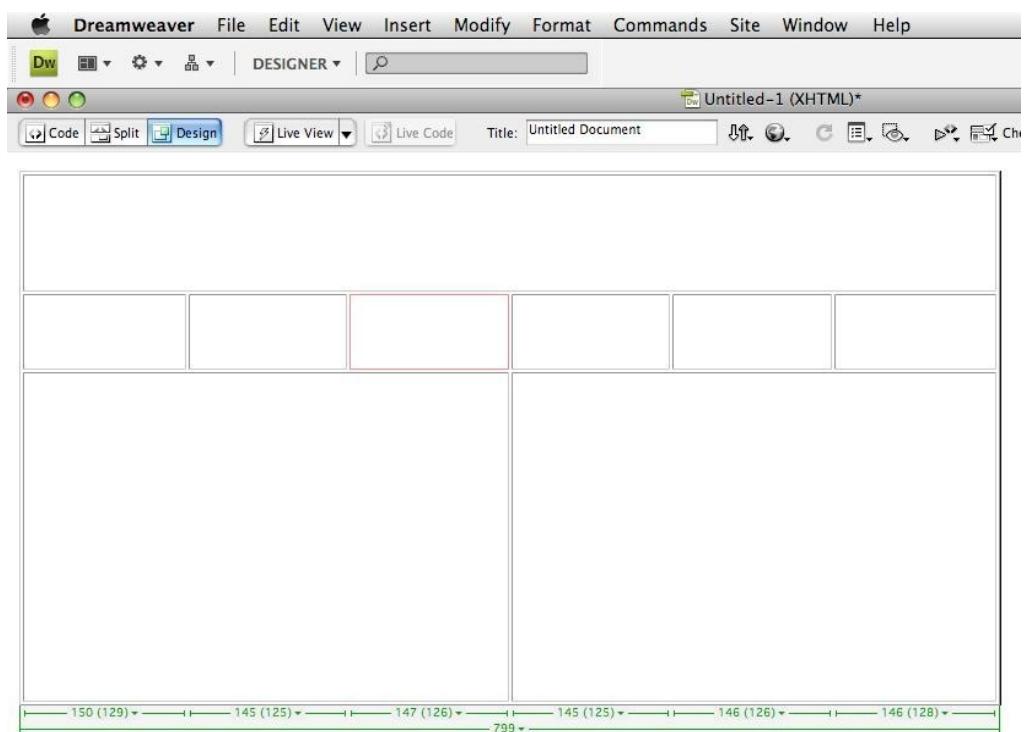
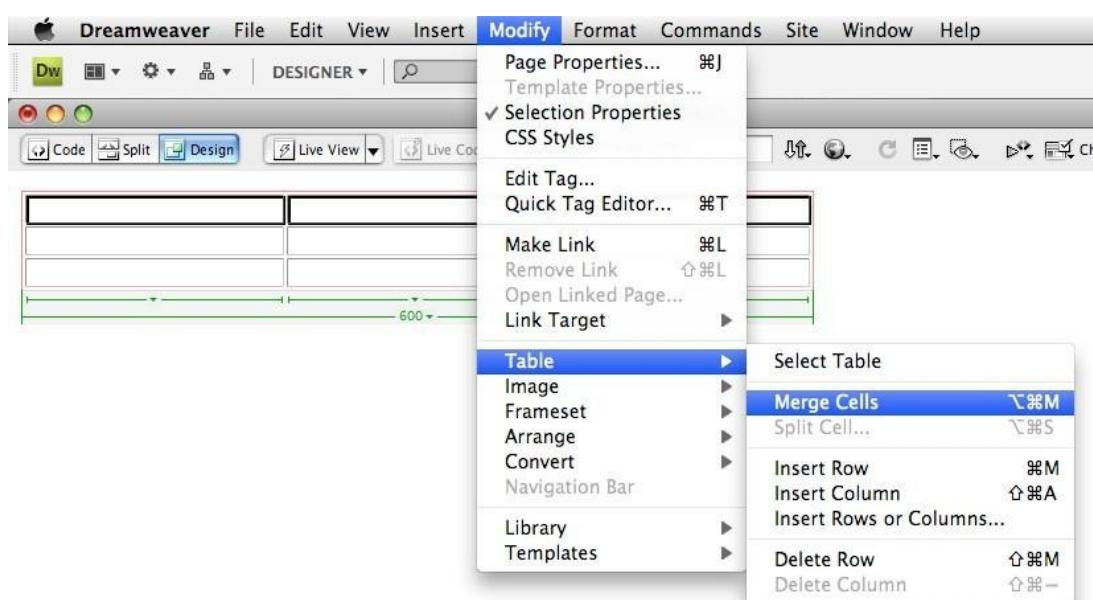
If a resource is not specified then last opened database is closed. This function returns true if it closes connection successfully otherwise it returns false.

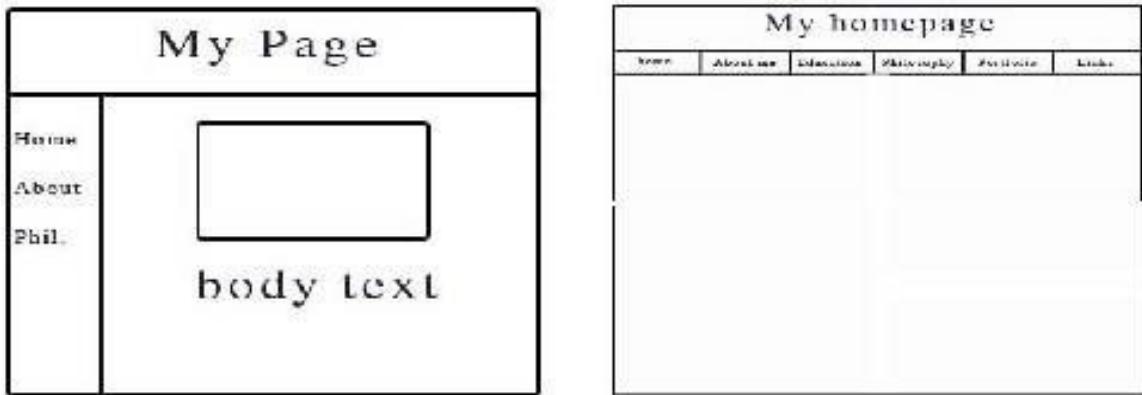
**Example:**

Try out following example to connect to a MySQL server:

```
<html>
<head>
<title>Connecting MySQL Server</title>
</head>
<body>
<?php
$dbhost = 'localhost:3036';
$dbuser = 'guest';
$dbpass = 'guest123';
$conn = mysql_connect($dbhost, $dbuser,
$dbpass); if(! $conn )
{
die('Could not connect: ' . mysql_error());
}
echo 'Connected successfully';
mysql_close($conn);
?>
</body>
</html>
```

## **DREAMWEAVER CS4:**





- Steps to creating a site in Dreamweaver CS4

Step 1: Define a site

Step 2: Create a template

Step 3. Design your template page

Step 4: Design your css style sheet

Step 5: Insert Editable Regions

Step 6: Create the pages of your site using the template

Step 7: Link all the pages together

Step 8: Create a form

Step 9: Check your site in the browser

Step 10: Upload your site and make it live

## APPENDIX B

### SOURCE CODE

#### SAMPLE SOURCE CODE

##### CART.PHP

```
<?php
require_once 'config/db_connect.php';

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header('Location: login.php');
    exit();
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Shopping Cart - <?php echo RESTAURANT_NAME; ?></title>
<link rel="stylesheet" href="assets/css/style.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
</head>
<body>
<?php include 'includes/header.php'; ?>

<div class="container py-5">
<h1 class="text-center mb-5">Shopping Cart</h1>

<div class="row">
```

```

<div class="col-lg-8">
  <!-- Cart Items -->
  <div class="card mb-4">
    <div class="card-body">
      <div id="cart-items">
        <!-- Cart items will be loaded here via JavaScript -->
      </div>
    </div>
  </div>
  </div>
</div>

<div class="col-lg-4">
  <!-- Order Summary -->
  <div class="card">
    <div class="card-body">
      <h5 class="card-title">Order Summary</h5>
      <div class="d-flex justify-content-between mb-3">
        <span>Subtotal:</span>
        <span id="subtotal"><?php echo CURRENCY_SYMBOL; ?>0.00</span>
      </div>
      <div class="d-flex justify-content-between mb-3">
        <span>Delivery Fee:</span>
        <span id="delivery-fee"><?php echo CURRENCY_SYMBOL; ?>50.00</span>
      </div>
      <hr>
      <div class="d-flex justify-content-between mb-3">
        <strong>Total:</strong>
        <strong id="total"><?php echo CURRENCY_SYMBOL; ?>0.00</strong>
      </div>
    </div>
  </div>

  <!-- Checkout Form -->
  <form id="checkout-form" action="process_order.php" method="POST">
    <input type="hidden" name="total_amount" id="total-amount">
    <input type="hidden" name="cart_items" id="cart-items-input">

```

```

<div class="mb-3">
    <label for="delivery-address" class="form-label">Delivery Address</label>
    <textarea class="form-control" id="delivery-address" name="delivery_address" rows="3" required></textarea>
</div>

<div class="mb-3">
    <label for="payment-method" class="form-label">Payment Method</label>
    <select class="form-select" id="payment-method" name="payment_method" required>
        <option value="">Select payment method</option>
        <option value="cash">Cash on Delivery</option>
        <option value="card">Credit/Debit Card</option>
        <option value="upi">UPI Payment</option>
    </select>
</div>

<!-- PayPal Button Container -->
<div id="paypal-button-container" style="display: none;"></div>

<!-- Regular Checkout Button -->
<button type="submit" class="btn btn-primary w-100" id="checkout-button">
    Place Order
</button>
</form>
</div>
</div>
</div>
</div>

<?php include 'includes/footer.php'; ?>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></
script>
<script>
const CURRENCY_SYMBOL = '<?php echo CURRENCY_SYMBOL; ?>';
const DELIVERY_FEE = 50.00; // Updated delivery fee to ₹50

// Load cart items
function loadCartItems() {
  const cartItems = JSON.parse(localStorage.getItem('cart')) || [];
  const cartContainer = document.getElementById('cart-items');
  let subtotal = 0;

  if (cartItems.length === 0) {
    cartContainer.innerHTML = '<p class="text-center">Your cart is empty</p>';
    updateOrderSummary(0);
    return;
  }

  let html = '';
  cartItems.forEach(item => {
    const itemTotal = item.price * item.quantity;
    subtotal += itemTotal;

    html += `
<div class="cart-item mb-3">
<div class="d-flex justify-content-between align-items-center">
<div>
<h6 class="mb-0">${item.name}</h6>
<small class="text-muted">${CURRENCY_SYMBOL}${item.price.toFixed(2)}</small>
each</small>
</div>
<div class="d-flex align-items-center">
<div class="quantity-control">

```

```

<button class="btn btn-sm btn-outline-secondary"
    onclick="updateQuantity(${item.id}, -1)">-</button>
<span class="mx-2">${item.quantity}</span>
<button class="btn btn-sm btn-outline-secondary"
    onclick="updateQuantity(${item.id}, 1)">+</button>
</div>
<span class="ms-3">${CURRENCY_SYMBOL}${itemTotal.toFixed(2)}</span>
</div>
</div>
</div>
`;
});

cartContainer.innerHTML = html;
updateOrderSummary(subtotal);

// Update hidden cart items input
document.getElementById('cart-items-input').value = JSON.stringify(cartItems);
}

// Update order summary
function updateOrderSummary(subtotal) {
const total = subtotal + DELIVERY_FEE;

document.getElementById('subtotal').textContent =
`${CURRENCY_SYMBOL}${subtotal.toFixed(2)}`;
document.getElementById('delivery-fee').textContent =
`${CURRENCY_SYMBOL}${DELIVERY_FEE.toFixed(2)}`;
document.getElementById('total').textContent =
`${CURRENCY_SYMBOL}${total.toFixed(2)}`;
document.getElementById('total-amount').value = total.toFixed(2);
}

// Handle payment method change

```

```

document.getElementById('payment-method').addEventListener('change', function(e)
{
  const paypalContainer = document.getElementById('paypal-button-container');
  const checkoutButton = document.getElementById('checkout-button');

  if (e.target.value === 'upi') {
    paypalContainer.style.display = 'block';
    checkoutButton.style.display = 'none';
  } else {
    paypalContainer.style.display = 'none';
    checkoutButton.style.display = 'block';
  }
});

// Handle form submission
document.getElementById('checkout-form').addEventListener('submit', function(e) {
  const cartItems = JSON.parse(localStorage.getItem('cart')) || [];
  if (cartItems.length === 0) {
    e.preventDefault();
    alert('Your cart is empty. Please add items before checking out.');
    return;
  }

  // Clear cart after successful submission
  localStorage.removeItem('cart');
});

// Load cart items on page load
document.addEventListener('DOMContentLoaded', loadCartItems);
</script>
</body>
</html>

```

## MENU.PHP

```
<?php
require_once 'config/db_connect.php';

// Fetch categories
$categories_query = "SELECT * FROM categories ORDER BY name";
$categories_result = mysqli_query($conn, $categories_query);

// Fetch menu items
$category_id = isset($_GET['category']) ? (int)$_GET['category'] : 0;
$menu_query = "SELECT * FROM menu_items";
if ($category_id > 0) {
    $menu_query .= " WHERE category_id = $category_id";
}
$menu_query .= " ORDER BY name";
$menu_result = mysqli_query($conn, $menu_query);
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Menu - Ocean Breeze</title>
<link rel="stylesheet" href="assets/css/style.css">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
</head>
<body>
<?php include 'includes/header.php'; ?>
```

```

<div class="container py-5">
<h1 class="text-center mb-5">Our Menu</h1>

<!-- Categories -->
<div class="row mb-4">
<div class="col-12">
<div class="d-flex justify-content-center flex-wrap gap-2">
<a href="menu.php" class="btn <?php echo !$category_id ? 'btn-primary' : 'btn-
outline-primary'; ?>">
    All
</a>
<?php while ($category = mysqli_fetch_assoc($categories_result)): ?>
<a href="menu.php?category=<?php echo $category['id']; ?>">
    class="btn <?php echo $category_id == $category['id'] ? 'btn-primary' : 'btn-outline-
primary'; ?>">
        <?php echo htmlspecialchars($category['name']); ?>
    </a>
<?php endwhile; ?>
</div>
</div>
</div>

<!-- Menu Items -->
<div class="row">
<?php while ($item = mysqli_fetch_assoc($menu_result)): ?>
<div class="col-md-6 col-lg-4 mb-4">
<div class="card h-100">
<?php if ($item['image']): ?>

    class="card-img-top"
    alt="<?php echo htmlspecialchars($item['name']); ?>"
    style="height: 200px; object-fit: cover;">
<?php else: ?>

<?php endif; ?>
<div class="card-body">
<h5 class="card-title"><?php echo htmlspecialchars($item['name']); ?></h5>
<p class="card-text"><?php echo htmlspecialchars($item['description']); ?></p>
<p class="card-text"><strong><?php echo CURRENCY_SYMBOL . number_format($item['price'], 2); ?></strong></p>
<div class="d-flex justify-content-between align-items-center">
<?php if ($item['is_available']): ?>
<button class="btn btn-primary"
onclick="addToCart(<?php echo $item['id']; ?>, '<?php echo addslashes($item['name']); ?>', <?php echo $item['price']; ?>)">
Add to Cart
</button>
<?php else: ?>
<button class="btn btn-secondary" disabled>Out of Stock</button>
<?php endif; ?>
</div>
</div>
</div>
</div>
<?php endwhile; ?>
</div>
</div>

<!-- Toast Container -->
<div class="position-fixed bottom-0 end-0 p-3" style="z-index: 11">
<div id="toast" class="toast" role="alert" aria-live="assertive" aria-atomic="true">
<div class="toast-header">
<strong class="me-auto">Notification</strong>

```

```

<button type="button" class="btn-close" data-bs-dismiss="toast" aria-
label="Close"></button>
</div>
<div class="toast-body"></div>
</div>
</div>

<?php include 'includes/footer.php'; ?>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></
script>
<script src="assets/js/main.js"></script>
</body>
</html>

```

## **MY ORDERS.PHP**

```

<?php
require_once 'config/db_connect.php';
include 'includes/header.php';

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header('Location: login.php');
    exit();
}

// Fetch user's orders
$user_id = $_SESSION['user_id'];
$query = "SELECT o.*,
    GROUP_CONCAT(CONCAT(oi.quantity, 'x ', mi.name) SEPARATOR ',') as items
    FROM orders o
    LEFT JOIN order_items oi ON o.id = oi.order_id

```

```
LEFT JOIN menu_items mi ON oi.menu_item_id = mi.id  
WHERE o.user_id = ?  
GROUP BY o.id  
ORDER BY o.created_at DESC";
```

```
$stmt = mysqli_prepare($conn, $query);  
mysqli_stmt_bind_param($stmt, "i", $user_id);  
mysqli_stmt_execute($stmt);  
$result = mysqli_stmt_get_result($stmt);  
?>  
  
<div class="container py-5">  
  <div class="row">  
    <div class="col-md-12">  
      <h2 class="mb-4">My Orders</h2>  
  
      <?php if (mysqli_num_rows($result) > 0): ?>  
      <div class="table-responsive">  
        <table class="table table-hover">  
          <thead class="table-dark">  
            <tr>  
              <th>Order ID</th>  
              <th>Items</th>  
              <th>Total Amount</th>  
              <th>Status</th>  
              <th>Payment Status</th>  
              <th>Date</th>  
              <th>Actions</th>  
            </tr>  
          </thead>  
          <tbody>  
            <?php while ($order = mysqli_fetch_assoc($result)): ?>  
            <tr>  
              <td>#<?php echo $order['id']; ?></td>
```

```

<td>
<small><?php echo htmlspecialchars($order['items']); ?></small>
</td>
<td>$<?php echo number_format($order['total_amount'], 2); ?></td>
<td>
<span class="badge bg-<?php
echo match($order['status']) {
'pending' => 'warning',
'confirmed' => 'info',
'delivered' => 'success',
'cancelled' => 'danger',
default => 'secondary'
};
?>">
<?php echo ucfirst($order['status']); ?>
</span>
</td>
<td>
<span class="badge bg-<?php
echo match($order['payment_status']) {
'pending' => 'warning',
'paid' => 'success',
'failed' => 'danger',
default => 'secondary'
};
?>">
<?php echo ucfirst($order['payment_status']); ?>
</span>
</td>
<td><?php echo date('M d, Y H:i', strtotime($order['created_at'])); ?></td>
<td>
<button type="button" class="btn btn-sm btn-primary"
onclick="viewOrderDetails(<?php echo $order['id']; ?>)">
View Details

```

```

        </button>
    </td>
</tr>
<?php endwhile; ?>
</tbody>
</table>
</div>
<?php else: ?>
<div class="alert alert-info">
<i class="fas fa-info-circle"></i> You haven't placed any orders yet.
<a href="menu.php" class="alert-link">Browse our menu</a> to place your first
order!
</div>
<?php endif; ?>
</div>
</div>
</div>

<!-- Order Details Modal -->
<div class="modal fade" id="orderDetailsModal" tabindex="-1">
<div class="modal-dialog modal-lg">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Order Details</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal"></button>
</div>
<div class="modal-body">
<div id="orderDetailsContent">
Loading...
</div>
</div>
</div>
</div>
</div>

```

```

<script>

function viewOrderDetails(orderId) {
    const modal = new bootstrap.Modal(document.getElementById('orderDetailsModal'));
    const contentDiv = document.getElementById('orderDetailsContent');

    // Show modal with loading state
    modal.show();
    contentDiv.innerHTML = '<div class="text-center"><i class="fas fa-spinner fa-spin"></i> Loading order details...</div>';

    // Fetch order details
    fetch(`get_order_items.php?order_id=${orderId}`)
        .then(response => response.json())
        .then(data => {
            let html = '<div class="table-responsive"><table class="table">';
            html +=
            '<thead><tr><th>Item</th><th>Quantity</th><th>Price</th><th>Subtotal</th></tr>'
            '</thead><tbody>';

            let total = 0;
            data.forEach(item => {
                const subtotal = parseFloat(item.unit_price) * parseInt(item.quantity);
                total += subtotal;
                html += `<tr>
                <td>${item.name}</td>
                <td>${item.quantity}</td>
                <td>${parseFloat(item.unit_price).toFixed(2)}</td>
                <td>${subtotal.toFixed(2)}</td>
                </tr>`;
            });

            html += `<tr class="table-dark">
```

```

<td colspan="3" class="text-end"><strong>Total:</strong></td>
<td><strong>$$ {total.toFixed(2)}</strong></td>
</tr>`;

html += '</tbody></table></div>';
contentDiv.innerHTML = html;
})
.catch(error => {
contentDiv.innerHTML = '<div class="alert alert-danger">Error loading order details.
Please try again.</div>';
console.error('Error:', error);
});
}
</script>

```

```
<?php include 'includes/footer.php'; ?>
```

## **RESERVATION.PHP**

```

<?php
require_once 'config/db_connect.php';

// Check if user is logged in
if (!isset($_SESSION['user_id'])) {
    header('Location: login.php');
    exit();
}

// Include header after session check
require_once 'includes/header.php';

$success = $error = "";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {

```

```

$date = mysqli_real_escape_string($conn, $_POST['date']);
$time = mysqli_real_escape_string($conn, $_POST['time']);
$guests = (int)$_POST['guests'];
$special_requests = mysqli_real_escape_string($conn, $_POST['special_request']);

// Validate date and time
$datetime = new DateTime("$date $time");
$now = new DateTime();

if ($datetime <= $now) {
    $error = 'Please select a future date and time';
} else {
    // Check if the selected time slot is available
    $check_query = "SELECT * FROM reservations
    WHERE reservation_date = ?
    AND reservation_time = ?
    AND status != 'cancelled'";
    $stmt = mysqli_prepare($conn, $check_query);
    mysqli_stmt_bind_param($stmt, "ss", $date, $time);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

    if (mysqli_num_rows($result) >= 5) { // Assuming max 5 reservations per time slot
        $error = 'Selected time slot is fully booked. Please choose another time.';
    } else {
        // Insert reservation
        $user_id = $_SESSION['user_id'];
        $insert_query = "INSERT INTO reservations (user_id, reservation_date,
        reservation_time, num_guests, special_requests, status)
        VALUES (?, ?, ?, ?, ?, 'pending')";

        $stmt = mysqli_prepare($conn, $insert_query);
        mysqli_stmt_bind_param($stmt, "issis", $user_id, $date, $time, $guests,
        $special_requests);
    }
}

```

```
if(mysqli_stmt_execute($stmt)) {  
    $success = 'Reservation submitted successfully! We will confirm your reservation  
shortly.';  
} else {  
    $error = 'Failed to submit reservation. Please try again.';  
}  
}  
}  
}  
}  
}  
?  
?
```

```
<div class="container py-5">  
    <div class="row justify-content-center">  
        <div class="col-md-8 col-lg-6">  
            <div class="card shadow">  
                <div class="card-body">  
  
                    <h2 class="text-center mb-4">Table Reservation</h2>  
  
                    <?php if ($success): ?>  
                    <div class="alert alert-success alert-dismissible fade show">  
                        <?php echo $success; ?>  
                        <button type="button" class="btn-close" data-bs-dismiss="alert"></button>  
                    </div>  
                    <?php endif; ?>  
  
                    <?php if ($error): ?>  
                    <div class="alert alert-danger alert-dismissible fade show">  
                        <?php echo $error; ?>  
                        <button type="button" class="btn-close" data-bs-dismiss="alert"></button>  
                    </div>  
                    <?php endif; ?>
```

```

<form method="POST" action="" id="reservation-form" class="needs-validation"
novalidate>

<div class="mb-3">
<label for="date" class="form-label">Date</label>
<input type="date" class="form-control" id="date" name="date" required
min="<?php echo date('Y-m-d'); ?>" value="<?php echo $_POST['date'] ?? ''; ?>">
<div class="invalid-feedback">Please select a valid date.</div>
</div>

<div class="mb-3">
<label for="time" class="form-label">Time</label>
<select class="form-select" id="time" name="time" required>
<option value="">Select time</option>
<?php
// Generate time slots from 11 AM to 10 PM
$start = strtotime('11:00');
$end = strtotime('22:00');
$interval = 30 * 60; // 30 minutes

for ($time = $start; $time <= $end; $time += $interval) {
$time_value = date('H:i:s', $time);
$time_display = date('h:i A', $time);
$selected = ($selected_time === $time_value) ? 'selected' : '';
echo "<option value=\"$time_value\" $selected>$time_display</option>";
}
?>
</select>
<div class="invalid-feedback">Please select a time slot.</div>
</div>

<div class="mb-3">
<label for="guests" class="form-label">Number of Guests</label>
<select class="form-select" id="guests" name="guests" required>
<option value="">Select number of guests</option>

```

```

<?php
$selected_guests = $_POST['guests'] ?? "";
for ($i = 1; $i <= 10; $i++):
$selected = ($selected_guests == $i) ? 'selected' : "";
?>
<option value=<?php echo $i; ?> <?php echo $selected; ?>><?php echo
$i; ?></option>
<?php endfor; ?>
</select>
<div class="invalid-feedback">Please select the number of guests.</div>
</div>

<div class="mb-3">
<label for="special_request" class="form-label">Special Requests
(Optional)</label>
<textarea class="form-control" id="special_request" name="special_request"
rows="3"><?php echo htmlspecialchars($_POST['special_request']) ?? ""; ?></textarea>
</div>

<div class="d-grid">
<button type="submit" class="btn btn-primary">Submit Reservation</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>

<script>
// Form validation
(function() {

```

```

'use strict';

const forms = document.querySelectorAll('.needs-validation');

Array.from(forms).forEach(form => {
  form.addEventListener('submit', event => {
    if (!form.checkValidity()) {
      event.preventDefault();
      event.stopPropagation();
    }
  }

  form.classList.add('was-validated');
}, false);
});

})0;

// Disable past dates
document.getElementById('date').min = new Date().toISOString().split('T')[0];

// Auto-hide alerts after 5 seconds
document.addEventListener('DOMContentLoaded', function() {
  setTimeout(function() {
    const alerts = document.querySelectorAll('.alert');
    alerts.forEach(function(alert) {
      const bsAlert = new bootstrap.Alert(alert);
      bsAlert.close();
    });
  }, 5000);
});

```

</script>

<?php include 'includes/footer.php'; ?>

## **ORDER CONFIRMATION.PHP**

```
<?php  
require_once 'config/db_connect.php';  
  
// Check if order ID is provided  
if (!isset($_GET['order_id'])) {  
    header('Location: index.php');  
    exit();  
}  
  
$order_id = intval($_GET['order_id']);  
  
// Get order details with user information  
$order_query = "SELECT o.*, u.full_name, u.email, u.username  
FROM orders o  
JOIN users u ON o.user_id = u.id  
WHERE o.id = ?";  
$stmt = mysqli_prepare($conn, $order_query);  
mysqli_stmt_bind_param($stmt, "i", $order_id);  
mysqli_stmt_execute($stmt);  
$order_result = mysqli_stmt_get_result($stmt);  
$order = mysqli_fetch_assoc($order_result);  
  
// Get order items  
$items_query = "SELECT oi.*, m.name, m.price  
FROM order_items oi  
JOIN menu_items m ON oi.menu_item_id = m.id  
WHERE oi.order_id = ?";  
$stmt = mysqli_prepare($conn, $items_query);  
mysqli_stmt_bind_param($stmt, "i", $order_id);  
mysqli_stmt_execute($stmt);  
$items_result = mysqli_stmt_get_result($stmt);  
?>
```

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Order Confirmation</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
<style>
body {
background: linear-gradient(135deg, #f8f9fa 0%, #e9ecef 100%);
min-height: 100vh;
display: flex;
align-items: center;
justify-content: center;
padding: 20px;
}

.confirmation-card {
background: #fff;
border-radius: 15px;
box-shadow: 0 0 30px rgba(0,0,0,0.1);
overflow: hidden;
max-width: 600px;
width: 100%;
text-align: center;
animation: slideIn 0.5s ease-out;
}

@keyframes slideIn {
from {
transform: translateY(-20px);
}

```

```
    opacity: 0;
}
to {
    transform: translateY(0);
    opacity: 1;
}

.confirmation-header {
    background: linear-gradient(135deg, #4CAF50 0%, #45a049 100%);
    color: white;
    padding: 40px 30px;
}

.confirmation-header i {
    font-size: 5rem;
    margin-bottom: 20px;
    animation: checkmark 0.5s ease-in-out;
}

@keyframes checkmark {
    0% { transform: scale(0); opacity: 0; }
    50% { transform: scale(1.2); }
    100% { transform: scale(1); opacity: 1; }
}

.confirmation-body {
    padding: 30px;
}

.order-number {
    font-size: 1.2rem;
    color: #6c757d;
    margin: 20px 0;
```

```
}

.user-name {
    font-size: 1.5rem;
    color: #333;
    margin-bottom: 10px;
}

.btn-continue {
    background: #4CAF50;
    color: white;
    padding: 15px 40px;
    border-radius: 30px;
    border: none;
    font-size: 1.1rem;
    text-transform: uppercase;
    letter-spacing: 1px;
    transition: all 0.3s ease;
    margin-top: 20px;
}

.btn-continue:hover {
    background: #45a049;
    transform: translateY(-2px);
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    color: white;
}

</style>
</head>
<body>
<div class="confirmation-card">
<div class="confirmation-header">
<i class="fas fa-check-circle"></i>
<h2>Order Confirmed!</h2>
```

```

</div>
<div class="confirmation-body">
<div class="user-name">
Thank you, <?php echo htmlspecialchars($order['full_name']); ?>!
</div>
<p class="lead">Your order has been successfully placed.</p>
<div class="order-number">
Order #<?php echo str_pad($order['id'], 6, '0', STR_PAD_LEFT); ?>
</div>
<p>We'll start preparing your delicious meal right away!</p>
<a href="menu.php" class="btn btn-continue">
<i class="fas fa-utensils me-2"></i>Continue Shopping
</a>
</div>
</div>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></
script>
</body>
</html>

```

## **MENU ITEMS.PHP(ADMIN)**

```

<?php
require_once '../config/db_connect.php';

// Check if admin is logged in
if (!isset($_SESSION['admin_id'])) {
    header('Location: admin_login.php');
    exit();
}

// Handle form submissions
if ($_SERVER['REQUEST_METHOD'] == 'POST') {

```

```

if (isset($_POST['action'])) {
    switch ($_POST['action']) {
        case 'add':
        case 'edit':
            $id = isset($_POST['id']) ? $_POST['id'] : null;
            $name = mysqli_real_escape_string($conn, $_POST['name']);
            $category_id = mysqli_real_escape_string($conn, $_POST['category_id']);
            $description = mysqli_real_escape_string($conn, $_POST['description']);
            $price = mysqli_real_escape_string($conn, $_POST['price']);
            $is_available = isset($_POST['is_available']) ? 1 : 0;

            // Handle image upload
            $image = "";
            if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
                $allowed = ['image/png', 'image/jpeg', 'image/jpg'];
                if (in_array($_FILES['image']['type'], $allowed)) {
                    $filename = time() . '_' . $_FILES['image']['name'];
                    $upload_path = '../uploads/menu/';

                    // Create directory if it doesn't exist
                    if (!file_exists($upload_path)) {
                        mkdir($upload_path, 0777, true);
                    }

                    if (move_uploaded_file($_FILES['image']['tmp_name'], $upload_path . $filename)) {
                        $image = $filename;
                    }
                }
            }

            if ($id) {
                // Update existing item
                $image_sql = $image ? ", image='$image'" : "";
                $sql = "UPDATE menu_items SET

```

```

name='$name',
category_id='$category_id',
description='$description',
price='$price',
is_available='$is_available'
$image_sql
WHERE id=$id";
} else {
// Add new item
$sql = "INSERT INTO menu_items (name, category_id, description, price, image,
is_available)
VALUES ('$name', '$category_id', '$description', '$price', '$image', '$is_available')";
}

if(mysqli_query($conn, $sql)) {
$_SESSION['success'] = "Menu item " . ($id ? "updated" : "added") . " successfully!";
} else {
$_SESSION['error'] = "Error: " . mysqli_error($conn);
}
break;

case 'delete':
$id = mysqli_real_escape_string($conn, $_POST['id']);
// Get image filename before deleting
$result = mysqli_query($conn, "SELECT image FROM menu_items WHERE
id=$id");
if ($row = mysqli_fetch_assoc($result)) {
if ($row['image']) {
@unlink('../uploads/menu/' . $row['image']);
}
}
if (mysqli_query($conn, "DELETE FROM menu_items WHERE id=$id")) {
$_SESSION['success'] = "Menu item deleted successfully!";
} else {

```

```

$_SESSION['error'] = "Error deleting menu item: " . mysqli_error($conn);
}
break;
}
header('Location: menu_items.php');
exit();
}
}

// Fetch categories
$categories = mysqli_query($conn, "SELECT * FROM categories ORDER BY
name");

// Fetch menu items with category names
$menu_items = mysqli_query($conn, "SELECT m.*, c.name as category_name
FROM menu_items m
LEFT JOIN categories c ON m.category_id = c.id
ORDER BY c.name, m.name");

require_once 'includes/admin_header.php';
?>

<div class="container-fluid px-4">
<h1 class="mt-4">Menu Items Management</h1>
<ol class="breadcrumb mb-4">
<li class="breadcrumb-item"><a href="index.php">Dashboard</a></li>
<li class="breadcrumb-item active">Menu Items</li>
</ol>

<?php if (isset($_SESSION['success'])): ?>
<div class="alert alert-success alert-dismissible fade show" role="alert">
<?php
echo $_SESSION['success'];
unset($_SESSION['success']);

```

```

?>
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if (isset($_SESSION['error'])): ?>
<div class="alert alert-danger alert-dismissible fade show" role="alert">
<?php
echo $_SESSION['error'];
unset($_SESSION['error']);
?>
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<!-- Add/Edit Menu Item Form -->
<div class="card mb-4">
<div class="card-header">
<i class="fas fa-plus me-1"></i>
Add/Edit Menu Item
</div>
<div class="card-body">
<form id="menuItemForm" method="POST" enctype="multipart/form-data">
<input type="hidden" name="action" value="add">
<input type="hidden" name="id" value="">

<div class="row mb-3">
<div class="col-md-6">
<label for="name" class="form-label">Item Name</label>
<input type="text" class="form-control" id="name" name="name" required>
</div>
<div class="col-md-6">
<label for="category_id" class="form-label">Category</label>
<select class="form-select" id="category_id" name="category_id" required>

```

```

<option value="">Select Category</option>
<?php while ($category = mysqli_fetch_assoc($categories)): ?>
<option value="<?php echo $category['id']; ?>">
<?php echo htmlspecialchars($category['name']); ?>
</option>
<?php endwhile; ?>
</select>
</div>
</div>

<div class="row mb-3">
<div class="col-md-6">
<label for="price" class="form-label">Price <?php echo CURRENCY_SYMBOL; ?></label>
<input type="number" class="form-control" id="price" name="price" step="0.01" required>
</div>
<div class="col-md-6">
<label for="image" class="form-label">Image (PNG/JPG)</label>
<input type="file" class="form-control" id="image" name="image" accept="image/png,image/jpeg,image/jpg">
<div id="currentImage" class="mt-2" style="display: none;">
<img src="" alt="Current image" style="height: 100px;">
<br>
<small class="text-muted">Leave empty to keep current image</small>
</div>
</div>
</div>

<div class="mb-3">
<label for="description" class="form-label">Description</label>
<textarea class="form-control" id="description" name="description" rows="3" required></textarea>
</div>

```

```

<div class="mb-3">
  <div class="form-check">
    <input class="form-check-input" type="checkbox" id="is_available"
      name="is_available" checked>
    <label class="form-check-label" for="is_available">
      Item is available
    </label>
  </div>
</div>

<button type="submit" class="btn btn-primary">Save Menu Item</button>
<button type="button" class="btn btn-secondary" onclick="resetForm()">Clear
  Form</button>
</form>
</div>
</div>

<!-- Menu Items List -->
<div class="card mb-4">
  <div class="card-header">
    <i class="fas fa-table me-1"></i>
    Menu Items List
  </div>
  <div class="card-body">
    <table id="menuItemsTable" class="table table-striped table-bordered">
      <thead>
        <tr>
          <th>Image</th>
          <th>Name</th>
          <th>Category</th>
          <th>Description</th>
          <th>Price</th>
          <th>Status</th>
        </tr>
      <thead>
      <tbody>
        <tr>
          <td><img alt="Placeholder image" style="width: 50px;"></td>
          <td>Item A</td>
          <td>Food</td>
          <td>A delicious meal.</td>
          <td>$10.00</td>
          <td>Available</td>
        </tr>
        <tr>
          <td><img alt="Placeholder image" style="width: 50px;"></td>
          <td>Item B</td>
          <td>Food</td>
          <td>A healthy meal.</td>
          <td>$12.00</td>
          <td>Available</td>
        </tr>
        <tr>
          <td><img alt="Placeholder image" style="width: 50px;"></td>
          <td>Item C</td>
          <td>Food</td>
          <td>A hearty meal.</td>
          <td>$15.00</td>
          <td>Available</td>
        </tr>
        <tr>
          <td><img alt="Placeholder image" style="width: 50px;"></td>
          <td>Item D</td>
          <td>Food</td>
          <td>A light meal.</td>
          <td>$8.00</td>
          <td>Available</td>
        </tr>
        <tr>
          <td><img alt="Placeholder image" style="width: 50px;"></td>
          <td>Item E</td>
          <td>Food</td>
          <td>A special meal.</td>
          <td>$18.00</td>
          <td>Available</td>
        </tr>
      <tbody>
    </table>
  </div>
</div>

```

```

<th>Actions</th>
</tr>
</thead>
<tbody>
<?php while ($item = mysqli_fetch_assoc($menu_items)): ?>
<tr>
<td>
<?php if ($item['image']): ?>
" style="height: 50px; width: 50px; object-fit: cover;">
<?php else: ?>

<?php endif; ?>
</td>
<td><?php echo htmlspecialchars($item['name']); ?></td>
<td><?php echo htmlspecialchars($item['category_name']); ?></td>
<td><?php echo htmlspecialchars(substr($item['description'], 0, 100)) . '...'; ?></td>
<td><?php echo CURRENCY_SYMBOL . number_format($item['price'], 2); ?></td>
<td>
<span class="badge bg-<?php echo $item['is_available'] ? 'success' : 'danger'; ?>">
<?php echo $item['is_available'] ? 'Available' : 'Not Available'; ?>
</span>
</td>
<td>
<button type="button" class="btn btn-primary btn-sm" onclick="editItem(<?php echo htmlspecialchars(json_encode($item)); ?>)">
Edit
</button>
<button type="button" class="btn btn-danger btn-sm"

```

```

    onclick="deleteItem(<?php echo $item['id']; ?>, '<?php echo
    htmlspecialchars($item['name']); ?>')"

Delete
</button>
</td>
</tr>
<?php endwhile; ?>
</tbody>
</table>
</div>
</div>
</div>

<!-- Delete Confirmation Modal -->
<div class="modal fade" id="deleteModal" tabindex="-1">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<h5 class="modal-title">Confirm Delete</h5>
<button type="button" class="btn-close" data-bs-dismiss="modal"></button>
</div>
<div class="modal-body">
Are you sure you want to delete this menu item?
</div>
<div class="modal-footer">
<form method="POST">
<input type="hidden" name="action" value="delete">
<input type="hidden" name="id" id="deleteItemId">
<button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Cancel</button>
<button type="submit" class="btn btn-danger">Delete</button>
</form>
</div>
</div>

```

```

</div>
</div>

<script>
// Initialize DataTable
$(document).ready(function() {
  $('#menuItemsTable').DataTable({
    order: [[2, 'asc'], [1, 'asc']] // Sort by category then name
  });
});

// Edit menu item
function editItem(item) {
  const form = document.getElementById('menuItemForm');
  form.elements['action'].value = 'edit';
  form.elements['id'].value = item.id;
  form.elements['name'].value = item.name;
  form.elements['category_id'].value = item.category_id;
  form.elements['description'].value = item.description;
  form.elements['price'].value = item.price;
  form.elements['is_available'].checked = item.is_available == 1;

  // Show current image if exists
  const currentImage = document.getElementById('currentImage');
  if (item.image) {
    currentImage.style.display = 'block';
    currentImage.querySelector('img').src = './uploads/menu/' + item.image;
  } else {
    currentImage.style.display = 'none';
  }

  // Scroll to form
  form.scrollIntoView({ behavior: 'smooth' });
}

```

```

// Delete menu item

function deleteItem(id, name) {
    document.getElementById('deleteItemId').value = id;
    const modal = new bootstrap.Modal(document.getElementById('deleteModal'));
    document.querySelector('#deleteModal .modal-body').innerHTML =
        `Are you sure you want to delete <strong>${name}</strong>?`;
    modal.show();
}

// Reset form

function resetForm() {
    const form = document.getElementById('menuItemForm');
    form.reset();
    form.elements['action'].value = 'add';
    form.elements['id'].value = "";
    document.getElementById('currentImage').style.display = 'none';
}
</script>

<?php require_once 'includes/admin_footer.php'; ?>
```

## LOGIN.PHP

```

<?php
require_once 'config/db_connect.php';

// Check if already logged in
if (isset($_SESSION['user_id'])) {
    header('Location: index.php');
    exit();
}

$error = ";
```

```

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = mysqli_real_escape_string($conn, $_POST['username']);
    $password = $_POST['password'];

    $query = "SELECT * FROM users WHERE username = ?";
    $stmt = mysqli_prepare($conn, $query);
    mysqli_stmt_bind_param($stmt, "s", $username);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

    if ($user = mysqli_fetch_assoc($result)) {
        if (password_verify($password, $user['password'])) {
            $_SESSION['user_id'] = $user['id'];
            $_SESSION['username'] = $user['username'];
            $_SESSION['role'] = $user['role'];
            header('Location: index.php');
            exit();
        } else {
            $error = 'Invalid username or password';
        }
    } else {
        $error = 'Invalid username or password';
    }
}

require_once 'includes/header.php';
?>

<!-- Restaurant Name Container -->
<div class="container-fluid bg-dark text-white py-4 text-center mb-4">
    <h1 class="display-4"><?php echo RESTAURANT_NAME; ?></h1>
    <p class="lead"><?php echo RESTAURANT_TAGLINE; ?></p>
</div>

```

```

<div class="container py-5">
  <div class="row justify-content-center">
    <div class="col-md-6 col-lg-4">
      <div class="card shadow">
        <div class="card-body">
          <h2 class="card-title text-center mb-4">Login</h2>

          <?php if ($error): ?>
          <div class="alert alert-danger alert-dismissible fade show">
            <?php echo $error; ?>
            <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
          </div>
          <?php endif; ?>

          <form method="POST" action="" class="needs-validation" novalidate>
            <div class="mb-3">
              <label for="username" class="form-label">Username</label>
              <input type="text" class="form-control" id="username" name="username" required>
              <div class="invalid-feedback">Please enter your username.</div>
            </div>

            <div class="mb-3">
              <label for="password" class="form-label">Password</label>
              <input type="password" class="form-control" id="password" name="password" required>
              <div class="invalid-feedback">Please enter your password.</div>
            </div>

            <div class="d-grid gap-2">
              <button type="submit" class="btn btn-primary">Login</button>
              <a href="register.php" class="btn btn-outline-secondary">Create New Account</a>
            </div>
          </form>

```

```

</div>
</div>
</div>
</div>
</div>

<script>
// Form validation
(function() {
  'use strict';

  const forms = document.querySelectorAll('.needs-validation');
  Array.from(forms).forEach(form => {
    form.addEventListener('submit', event => {
      if (!form.checkValidity()) {
        event.preventDefault();
        event.stopPropagation();
      }
      form.classList.add('was-validated');
    }, false);
  });
})();

// Auto-hide alerts after 5 seconds
document.addEventListener('DOMContentLoaded', function() {
  setTimeout(function() {
    const alerts = document.querySelectorAll('.alert');
    alerts.forEach(function(alert) {
      const bsAlert = new bootstrap.Alert(alert);
      bsAlert.close();
    });
  }, 5000);
});
</script>

```

```

<?php include 'includes/footer.php'; ?>

REGISTER.PHP

<?php
require_once 'config/db_connect.php';

// Check if already logged in
if (isset($_SESSION['user_id'])) {
    header('Location: index.php');
    exit();
}

$error = $success = "";

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $username = mysqli_real_escape_string($conn, $_POST['username']);
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $full_name = mysqli_real_escape_string($conn, $_POST['full_name']);
    $phone = mysqli_real_escape_string($conn, $_POST['phone']);

    // Validate input
    if ($password !== $confirm_password) {
        $error = 'Passwords do not match';
    } else {
        // Check if username exists
        $check_query = "SELECT id FROM users WHERE username = ? OR email = ?";
        $stmt = mysqli_prepare($conn, $check_query);
        mysqli_stmt_bind_param($stmt, "ss", $username, $email);
        mysqli_stmt_execute($stmt);
        $result = mysqli_stmt_get_result($stmt);

        if (mysqli_num_rows($result) > 0) {

```

```

$error = 'Username or email already exists';
} else {
// Insert new user
$hashed_password = password_hash($password, PASSWORD_DEFAULT);
$insert_query = "INSERT INTO users (username, password, email, full_name, phone)
VALUES (?, ?, ?, ?, ?)";
$stmt = mysqli_prepare($conn, $insert_query);
mysqli_stmt_bind_param($stmt, "sssss", $username, $hashed_password, $email,
$full_name, $phone);

if (mysqli_stmt_execute($stmt)) {
$success = 'Registration successful! You can now login.';
} else {
$error = 'Registration failed. Please try again.';
}

}

require_once 'includes/header.php';
?>

<!-- Restaurant Name Container -->
<div class="container-fluid bg-dark text-white py-4 text-center mb-4">
<h1 class="display-4"><?php echo RESTAURANT_NAME; ?></h1>
<p class="lead"><?php echo RESTAURANT_TAGLINE; ?></p>
</div>

<div class="container py-5">
<div class="row justify-content-center">
<div class="col-md-6">
<div class="card shadow">
<div class="card-body">
<h2 class="card-title text-center mb-4">Create Account</h2>

```

```

<?php if ($success): ?>
<div class="alert alert-success alert-dismissible fade show">
<?php echo $success; ?>
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<?php if ($error): ?>
<div class="alert alert-danger alert-dismissible fade show">
<?php echo $error; ?>
<button type="button" class="btn-close" data-bs-dismiss="alert"></button>
</div>
<?php endif; ?>

<form method="POST" action="" class="needs-validation" novalidate>
<div class="mb-3">
<label for="full_name" class="form-label">Full Name</label>
<input type="text" class="form-control" id="full_name" name="full_name" required
value="<?php echo htmlspecialchars($_POST['full_name'] ?? ""); ?>">
<div class="invalid-feedback">Please enter your full name.</div>
</div>

<div class="mb-3">
<label for="username" class="form-label">Username</label>
<input type="text" class="form-control" id="username" name="username" required
value="<?php echo htmlspecialchars($_POST['username'] ?? ""); ?>">
<div class="invalid-feedback">Please choose a username.</div>
</div>

<div class="mb-3">
<label for="email" class="form-label">Email</label>
<input type="email" class="form-control" id="email" name="email" required
value="<?php echo htmlspecialchars($_POST['email'] ?? ""); ?>">

```

```

<div class="invalid-feedback">Please enter a valid email address.</div>
</div>

<div class="mb-3">
<label for="phone" class="form-label">Phone Number</label>
<input type="tel" class="form-control" id="phone" name="phone" required
value=<?php echo htmlspecialchars($_POST['phone']) ?? ">">
<div class="invalid-feedback">Please enter your phone number.</div>
</div>

<div class="mb-3">
<label for="password" class="form-label">Password</label>
<input type="password" class="form-control" id="password" name="password"
required>
<div class="invalid-feedback">Please enter a password.</div>
</div>

<div class="mb-3">
<label for="confirm_password" class="form-label">Confirm Password</label>
<input type="password" class="form-control" id="confirm_password"
name="confirm_password" required>
<div class="invalid-feedback">Please confirm your password.</div>
</div>

<div class="d-grid gap-2">
<button type="submit" class="btn btn-primary">Register</button>
<a href="login.php" class="btn btn-outline-secondary">Already have an account?
Login</a>
</div>
</form>
</div>
</div>
</div>
</div>

```

```

</div>

<script>
// Form validation
(function() {
  'use strict';

  const forms = document.querySelectorAll('.needs-validation');
  Array.from(forms).forEach(form => {
    form.addEventListener('submit', event => {
      if (!form.checkValidity()) {
        event.preventDefault();
        event.stopPropagation();
      }
      form.classList.add('was-validated');
    }, false);
  });
})();

// Auto-hide alerts after 5 seconds
document.addEventListener('DOMContentLoaded', function() {
  setTimeout(function() {
    const alerts = document.querySelectorAll('.alert');
    alerts.forEach(function(alert) {
      const bsAlert = new bootstrap.Alert(alert);
      bsAlert.close();
    });
  }, 5000);
});
</script>

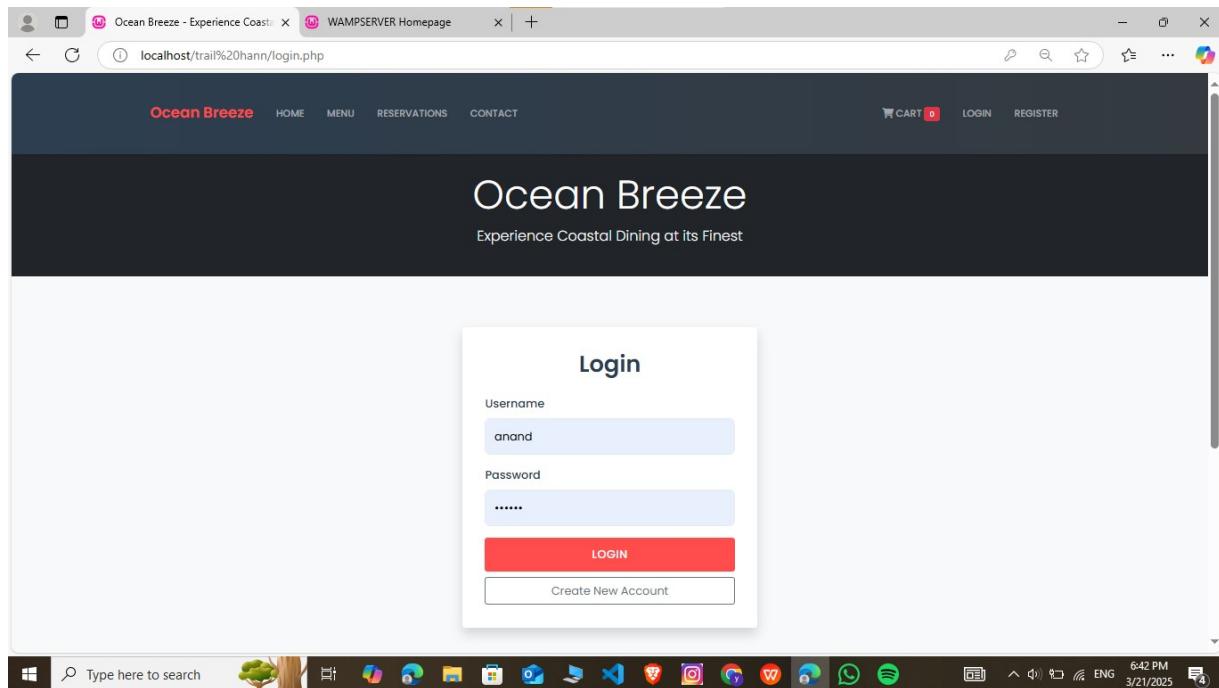
<?php include 'includes/footer.php'; ?>

```

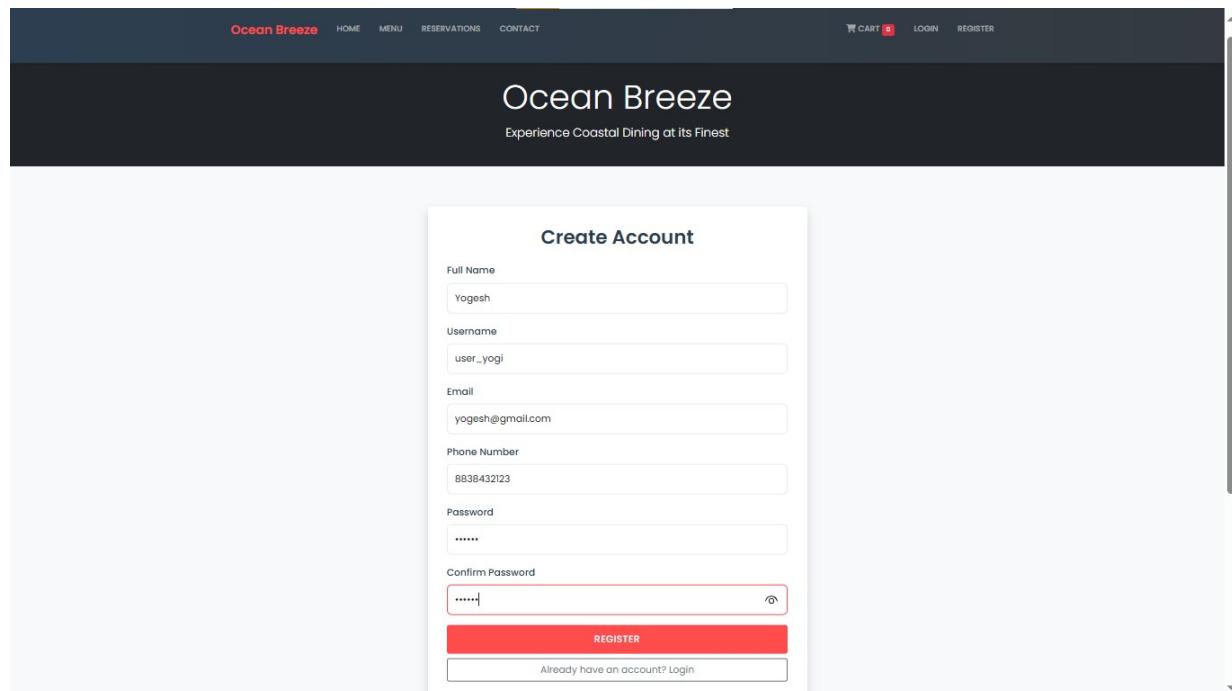
# APPENDIX C

## SCREENSHOT

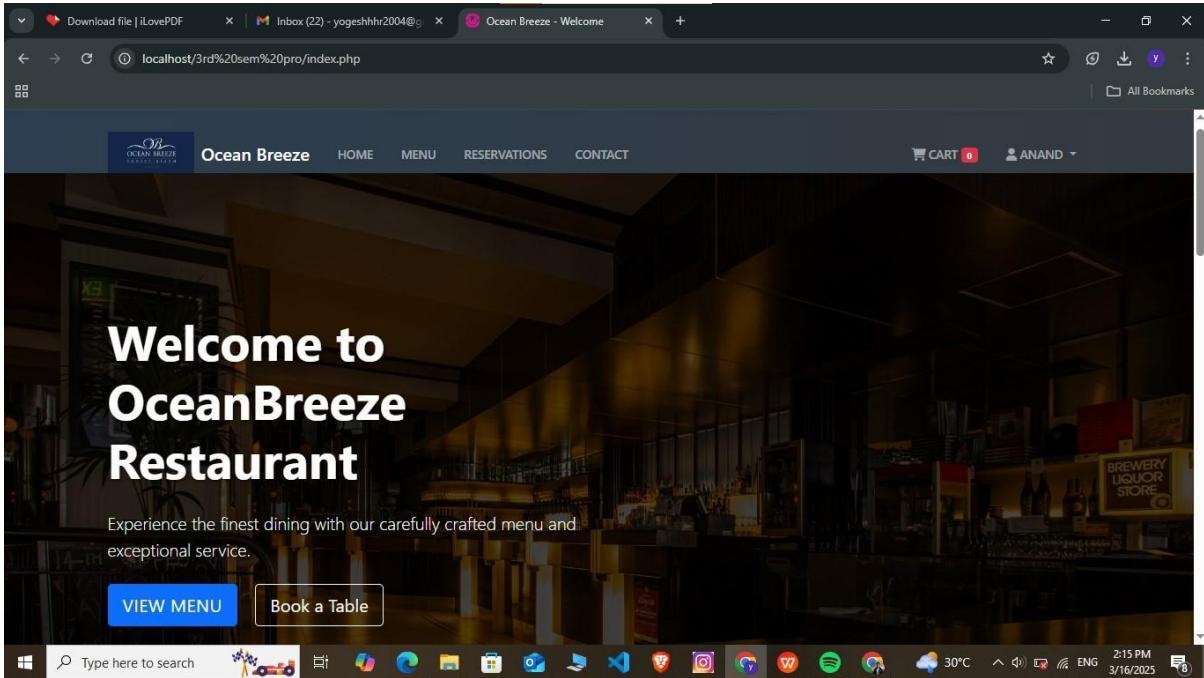
### LOGIN PAGE:



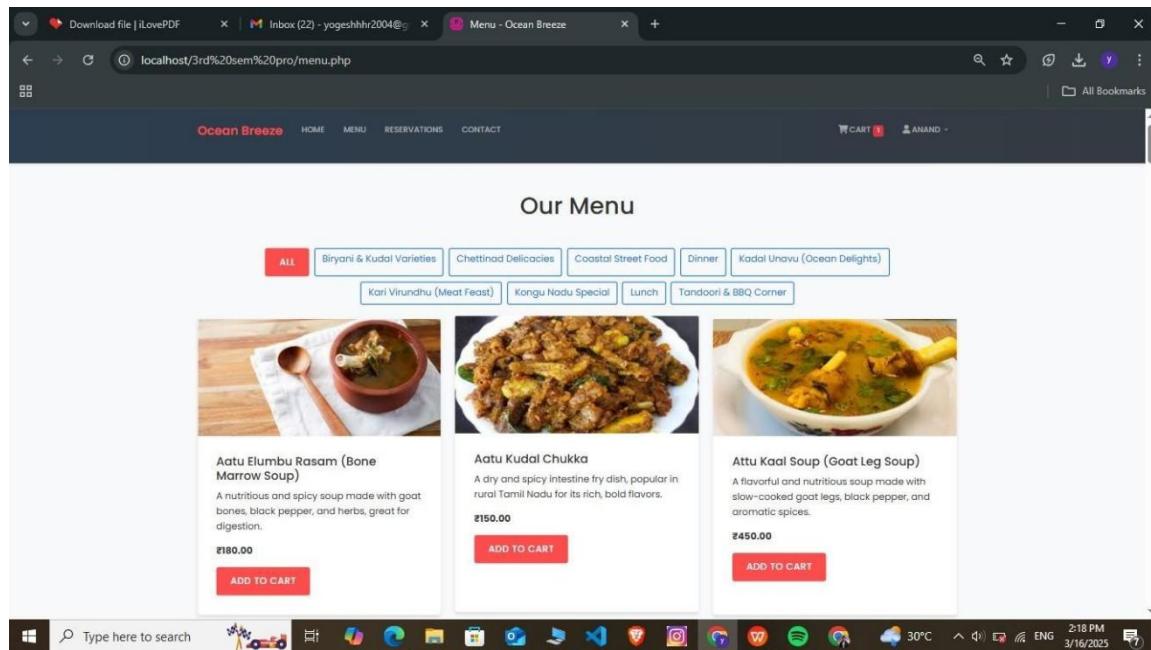
### SIGN UP PAGE:



## HOME PAGE:



## MENU PAGE:



## RESERVATION PAGE:

The screenshot shows a web browser window for the 'Ocean Breeze - Experience Coast' website. The URL in the address bar is 'localhost/trail%20hann/reservation.php'. The page title is 'Table Reservation'. It contains fields for Date (12/05/2024), Time (02:00 PM), Number of Guests (4), and Special Requests (Book This for a VIP). A red 'SUBMIT RESERVATION' button is at the bottom.

## CONTACT US PAGE:

The screenshot shows a web browser window for the 'Contact Us - Ocean Breeze' website. The URL in the address bar is 'localhost/trail%20hann/contact.php'. The page title is 'Contact Us'. It features a message: 'Have questions? We'd love to hear from you. Send us a message and we'll respond as soon as possible.' On the left, there are input fields for Name (anand), Email Address (anand123@gmail.com), Subject (In convenient space regard), and Message (please expand the dining hall, such a inconvenient space). A green 'Send Message' button is at the bottom. On the right, there is a 'Get in Touch' section with address (123 Restaurant Street, City, State 12345), phone number (+1 (555) 123-4567), email (info@yourrestaurant.com), and operating hours (Mon - Fri: 9:00 AM - 10:00 PM; Sat - Sun: 10:00 AM - 11:00 PM).

## CART PAGE:

The screenshot shows the shopping cart page for the Ocean Breeze restaurant. At the top, there's a dark header with the restaurant's name "Ocean Breeze" and links for "HOME", "MENU", "RESERVATIONS", and "CONTACT". On the right side of the header, there are icons for a shopping cart (with a red notification bubble) and user account information ("ANAND"). Below the header, the main content area has a title "Shopping Cart". On the left, there's a product listing for "Aatu Elumbu Rasam (Bone Marrow Soup)" at ₹180.00 each, with a quantity selector set to 1. On the right, there's an "Order Summary" section showing a subtotal of ₹180.00, a delivery fee of ₹50.00, and a total of ₹230.00. It also includes fields for "Delivery Address" (containing "east street cuddalore") and "Payment Method" (set to "Cash on Delivery"). A large red "PLACE ORDER" button is at the bottom. At the very bottom of the page, there's a footer with sections for "Contact Us", "Quick Links", and "Follow Us", along with the restaurant's address ("123 Restaurant Street") and social media links.

## USER PROFILE:

The screenshot shows the user profile page for the user "anand". The top navigation bar is identical to the cart page, with "Ocean Breeze", "HOME", "MENU", "RESERVATIONS", "CONTACT", a shopping cart icon (with a red notification bubble), and user account information ("ANAND"). The main content area features a "Profile Information" form. On the left, there's a user profile card with a blue circular icon, the name "anand", and the text "Member since March 2025". Below this is a red "CHANGE PASSWORD" button. On the right, the "Profile Information" form contains fields for "Username" (set to "anand"), "Full Name" (set to "user\_anand"), "Email" (set to "anand123@gmail.com"), and "Phone Number" (set to "8838432342"). There's also a "Delivery Address" field containing "42, east street cuddalore". At the bottom of the form is a red "SAVE CHANGES" button. The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators.

## MY ORDERS PAGE:

The screenshot shows a web browser window with multiple tabs open at the top. The main content area displays the "My Orders" section of the Ocean Breeze website. The header includes the restaurant's logo, navigation links for HOME, MENU, RESERVATIONS, and CONTACT, and a user account section for ANAND. Below the header is a table titled "My Orders" with the following data:

Order ID	Items	Total Amount	Status	Payment Status	Date	Actions
#5	ix Attu Kaal Soup (Goat Leg Soup), ix Elumbu Roast (Bone Marrow Roast)	\$604.00	Pending		Mar 16, 2025 12:00	<button>VIEW DETAILS</button>

At the bottom of the page, there is a footer with sections for "Contact Us", "Quick Links", and "Follow Us", along with a search bar and various icons.

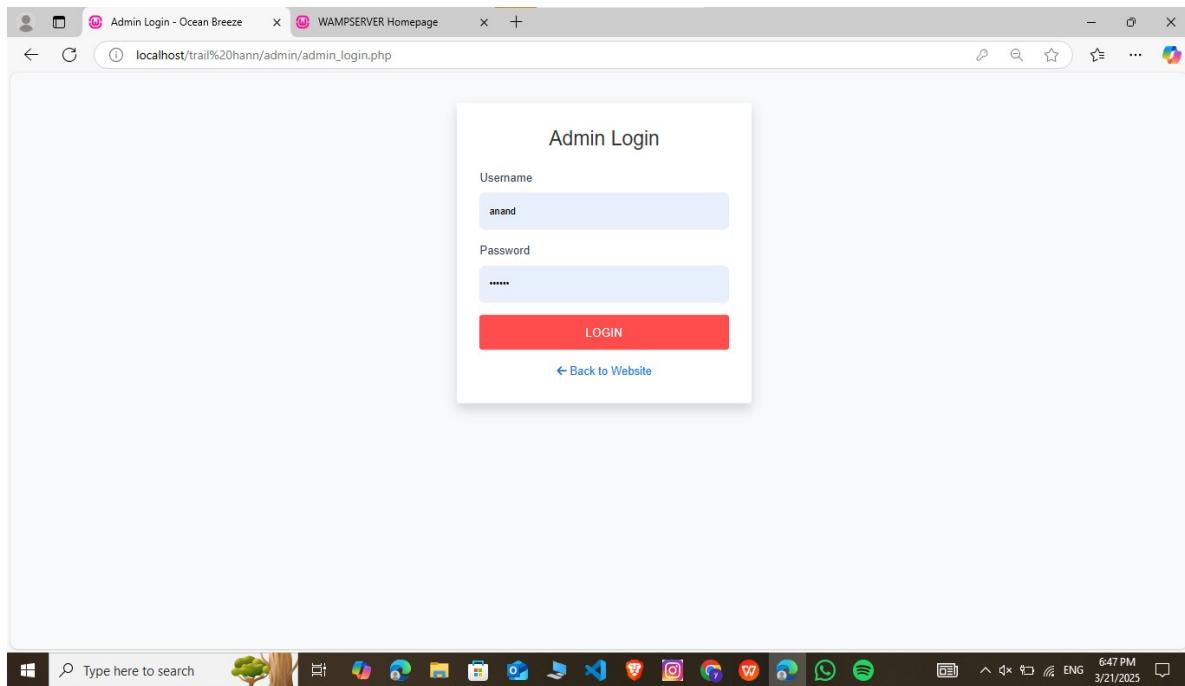
## MY RESERVATION PAGE:

The screenshot shows a web browser window with multiple tabs open at the top. The main content area displays the "My Reservations" section of the Ocean Breeze website. The header includes the restaurant's logo, navigation links for HOME, MENU, RESERVATIONS, and CONTACT, and a user account section for ANAND. Below the header is a table titled "My Reservations" with the following data:

Reservation ID	Date	Time	Guests	Status	Special Requests	Actions
#2	Dec 12, 2025	03:00 PM	5 people	Pending	No	<button>X Cancel</button>

At the bottom of the page, there is a footer with sections for "Contact Us", "Quick Links", and "Follow Us", along with a search bar and various icons.

## ADMIN LOGIN PAGE:



## ORDER MANAGEMENT PAGE:

A screenshot of a web browser window showing the 'Manage Orders' page. The URL in the address bar is 'localhost/3rd%20sem%20pro/admin/orders.php'. On the left, there is a dark sidebar menu titled 'Admin Panel' with options: Dashboard, Orders (which is selected), Menu Items, Categories, Reservations, and Users. The main content area is titled 'Manage Orders' and shows a table of 'Orders List'. The table has columns: Order ID, Customer, Contact, Amount, Status, Payment, Date, and Actions. There are five rows of data:

Order ID	Customer	Contact	Amount	Status	Payment	Date	Actions
#5	angonda anand	test1@gmail.com	\$604.00	Pending	Pending	Mar 16, 2025 12:00	<a href="#">View Details</a>
#4	Thirumaran Maran	mpk@gmail.com	\$13.99	Order Cancelled Order was cancelled	Cancelled	Mar 15, 2025 21:06	<a href="#">View Details</a>
#3	Thirumaran Maran	mpk@gmail.com	\$13.99	Order Cancelled Order was cancelled	Cancelled	Mar 15, 2025 20:21	<a href="#">View Details</a>
#2	Thirumaran Maran	mpk@gmail.com	\$34.99	Order Cancelled Order was cancelled	Cancelled	Mar 15, 2025 20:20	<a href="#">View Details</a>
#1	Thirumaran Maran	mpk@gmail.com	\$34.99	Order Cancelled Order was cancelled	Cancelled	Mar 15, 2025 20:17	<a href="#">View Details</a>

The browser's top navigation bar shows other tabs like 'Download file | iLovePDF' and 'Inbox (22) - yogeshhh2004@gmail.com'. The taskbar at the bottom of the screen shows various application icons.

## RESERVATION MANAGEMENT PAGE:

The screenshot shows a web-based reservation management system. On the left is a dark sidebar menu titled "Admin Panel" with options: Dashboard, Orders, Menu Items, Categories, Reservations (which is selected and highlighted in grey), and Users. Below the menu is a user info section with "admin" and a dropdown arrow. The main content area has a title "Manage Reservations". It features four colored boxes at the top: blue for Pending (0), green for Confirmed (0), red for Cancelled (1), and grey for Completed (0). Below these is a table with columns: ID, Customer, Contact, Date, Time, Guests, Special Requests, Status, and Actions. One row is visible in the table, showing ID 1, Customer Thirumaran, Contact Phone: N/A Email: mpk@gmail.com, Date Dec 04, 2025, Time 01:00 PM, Guests 4, Status Cancelled (with a red button), and Actions (with a blue "Update Status" button). The bottom of the screen shows a Windows taskbar with various icons and system status.

## ADMIN DASHBOARD PAGE:

The screenshot shows the Admin Dashboard. The sidebar menu is identical to the previous page: Dashboard, Orders, Menu Items, Categories, Reservations, and Users. The user info section shows "admin". The main content area has a title "Dashboard" and a subtitle "Dashboard". It features four colored boxes: blue for Total Orders (9), yellow for Pending Orders (2), green for Registered Users (3), and cyan for Menu Items (60). Each box has a "View Details" button. Below these is a section titled "Recent Orders" with a table. The table has columns: Order ID, Customer, Amount, Status, Date, and Action. Five recent orders are listed: #9 (angonda, ₹350.00, Pending, Mar 20, 2025 13:01, View Details); #8 (angonda, ₹250.00, Pending, Mar 17, 2025 18:28, View Details); #7 (saravanan, ₹200.00, In Progress, Mar 17, 2025 11:28, View Details); #6 (angonda, ₹250.00, In Progress, Mar 17, 2025 10:05, View Details); and #5 (angonda, ₹604.00, In Progress, Mar 16, 2025 12:00, View Details). The bottom of the screen shows a Windows taskbar with various icons and system status.

