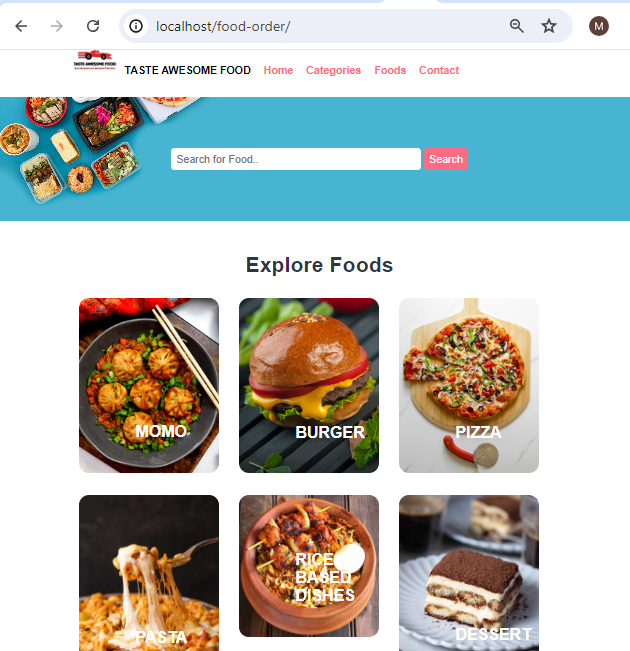
horizontal line



FOOD ORDERING WEBSITE PROJECT

# 

# Submitted by:

**─Mohammed Shohel**

Diploma In Computer Science and Application

Batch: 2022, Bangladesh Open University

Study Center: Daffodil Institute of IT, Chattogram.

Cell Phone: 01676252572, Student ID: **22150892005**

E-Mail: [mohammedshohel.info@gmail.com](mailto:mohammedshohel.info@gmail.com)

## **Table of Contents**

1. **Introduction**
2. **Motivation**
3. **Objectives**
4. **System Overview**
5. **Database Schema**
6. **Technologies Used**
7. **System Features**
8. **User Interfaces**
9. **Implementation Details**

**9.1 Database Design**

* 1. **Detailed explanation of database schema choices and normalization.**
  2. **SQL queries for creating tables and establishing relationships.**

**9.2 Screenshots**

* 1. **Visual representation of key application screens.**
  2. **Annotations highlighting important features.**

**9.3 Source Code**

* 1. **Overview of key source code files and modules.**
  2. **Code snippets illustrating critical functionalities.**

**9.4 Detailed System Architecture**

* 1. **Explanation of system components and their interactions.**
  2. **Deployment architecture (server setup, web server configuration).**

1. **Testing and Quality Assurance**
2. **Future Enhancements**
3. **Conclusion**
4. **References**

## **1. Introduction**

The Food Order Website aims to revolutionize the food service industry by offering an efficient online platform for ordering food. Customers can browse through a variety of food categories, select items of their choice, place orders securely, and track their delivery in real-time.

## **2. Motivation**

The motivation behind developing this system is to address the inefficiencies of traditional food ordering processes. By leveraging technology, the website aims to streamline order management, reduce errors, and enhance overall customer satisfaction.

## **3. Objectives**

The primary objectives of the Food Order Website include:

* **Automation**: Automate the food ordering process to minimize manual intervention.
* **Efficiency**: Improve order processing time and delivery logistics.
* **User-Friendly Experience**: Provide an intuitive interface for both customers and administrators.
* **Security**: Ensure secure transactions and data privacy.
* **Scalability**: Design a scalable system capable of handling increased user demand.

## **4. System Overview**

### **Architecture**

The system follows a client-server architecture where clients (customers) interact with the server (website) to place orders. The server communicates with the database to retrieve food items, manage orders, and update delivery status.

### **Workflow**

* **Customer Interface**: Customers browse food categories, select items, add them to the cart, proceed to checkout, and make payments.
* **Admin Interface**: Administrators manage food categories, add new items, view orders, and update order statuses.

## **5. Database Schema**

The database schema includes the following tables:

* **tbl\_admin**: Stores administrator details for managing the website.
* **tbl\_category**: Contains food categories with associated images.
* **tbl\_food**: Holds information about food items, including title, description, price, and category.
* **tbl\_order**: Stores order details such as food items ordered, quantities, total amount, customer information, and order status.

Refer to the provided SQL script for detailed schema and sample data.

## **6. Technologies Used**

* **Backend**: PHP
* **Database**: MySQL
* **Frontend**: HTML, CSS, JavaScript

## **7. System Features**

### **Customer-Facing Features**

**Customer Interaction**

* **Explore Food Options**: Customers can immerse themselves in a seamless browsing experience where they can explore a wide variety of food options. This exploration is facilitated through well-organized categories and detailed food menus. Each food item comes with high-quality images that give a visual representation of what they can expect, and detailed descriptions that provide information about ingredients, flavors, and other essential details.

**Browse and Search**

* **Browsing Food Categories**: Customers can easily navigate through different food categories, making it simple to find the type of cuisine or specific dishes they are interested in. Whether they are looking for appetizers, main courses, desserts, or beverages, the intuitive categorization helps them quickly locate their preferred options.
* **Search for Specific Items**: The robust search functionality allows customers to find specific food items by typing in keywords. This feature ensures that even if a customer is unsure about which category a particular dish belongs to, they can still find it effortlessly by using the search bar.

**Order Placement**

* **Placing Orders for Specific Items**: Once customers have found their desired food items, they can proceed to place an order. Each item page displays the price clearly, ensuring transparency and helping customers make informed decisions. Customers can also adjust the quantity of each item they wish to order, allowing for flexible ordering based on their needs, whether they are ordering for themselves or for a group.

**Order Confirmation**

* **Receiving a Success Message**: After customers confirm their order, they receive an immediate success message on the screen. This message serves to reassure them that their order has been successfully placed.
* **Order Delivery Notification**: Along with the success message, customers are also informed that their order will be delivered soon. This notification provides an estimated delivery time, keeping customers informed about the status of their order and setting the right expectations for when they can expect their food to arrive.

### **Admin-Facing Features**

* **Dashboard**: The dashboard provides admins with a comprehensive overview of the restaurant's operations. Admins can see the number of orders placed recently, monitor the status of orders currently being processed, and track the delivery of orders. The dashboard also displays canceled orders, the total number of orders, and the revenue earned, giving admins a clear picture of the restaurant's performance.
* **Manage Categories and Food Items**: Admins have full control over the menu. They can add new food categories and items, update existing ones with new information or images, and delete those that are no longer offered. This ensures that the menu is always up-to-date and reflects the current offerings of the restaurant.
* **Order Management**: This feature allows admins to efficiently handle customer orders. Admins can view the details of each order, including the items ordered, customer information, and order status. They can update the status of orders to reflect their progress, such as marking them as ordered, processing, delivered, or canceled. This helps in keeping track of all orders and ensuring timely delivery.
* **User Management**: Admins can manage all administrator accounts and their permissions. This includes creating new admin accounts, updating existing account information, and setting or modifying permissions to control what each admin can access and manage within the system. This feature ensures that the right people have the appropriate access to perform their roles effectively.

## **8. User Interfaces**

### **Customer Interface**

**Homepage**

* **Featured Food Categories**: The homepage welcomes customers with an attractive display of featured food categories. These categories are selected based on popularity, seasonal trends, or special promotions, offering a curated selection that invites exploration.

**Category Page**

* **List of Food Items**: When customers select a category, they are taken to a category page that lists all the food items within that category. Each item is accompanied by a high-quality image, a brief description, and the price, making it easy for customers to browse and choose.

**Foods Page**

* **Comprehensive Food Menu**: The foods page provides a comprehensive menu of all available food items, regardless of category. This page is perfect for customers who prefer to see all options at once. Items are organized and can be filtered by various criteria such as popularity, price, or dietary preferences.

**Contact Page**

* **Customer Support and Inquiries**: The contact page offers customers a way to get in touch with the restaurant for any inquiries or support. It typically includes a contact form, phone number, email address, and possibly a map showing the restaurant's location. This page ensures that customers can easily reach out for reservations, feedback, or any questions they may have.

### **Admin Interface**

**Dashboard**

* **Summary of Orders**: The dashboard provides a comprehensive overview of all orders, including the number of recent orders, orders in progress, delivered orders, and canceled orders. This allows admins to quickly assess the current state of operations.
* **Sales Summary**: Detailed insights into sales performance, including total revenue earned, average order value, and sales trends over time. This helps admins make informed business decisions.
* **Recent Activities**: A real-time feed of recent activities, such as new orders placed, status updates, and any changes made to the menu or categories. This keeps admins updated on the latest events and activities within the system.

**Category and Food Management**

* **Add New Categories and Food Items**: Admins can easily add new food categories and items to keep the menu fresh and updated. This includes uploading images, setting prices, and writing descriptions.
* **Update Existing Entries**: Existing categories and food items can be updated with new information, such as price changes, updated descriptions, or new images, ensuring that the menu remains accurate and appealing.
* **Delete Categories and Items**: Admins have the ability to remove categories and food items that are no longer available or relevant, helping to maintain a clean and accurate menu.

**Order Management**

* **View Orders**: A detailed list of all orders placed, including customer details, order items, quantities, prices, and order dates. This allows admins to review and manage each order effectively.
* **Update Order Statuses**: Admins can update the status of each order, marking them as ordered, processing, delivered, or canceled. This keeps both the kitchen and the customers informed about the current status of their orders.
* **Order Details**: Access to complete order details for in-depth order management, including special instructions from customers and payment status.

## **9. Implementation Details**

* **9.1 Database Design**

The database design plays a crucial role in this project by providing a structured framework for managing and accessing data efficiently. This document outlines the key aspects of our database schema and its implementation.

#### **Entity-Relationship Diagram (ERD)**

Entity-Relationship Diagram (ERD) illustrates the overall structure of our database. It visually represents how different entities are organized and their relationships. Each entity (table) is labeled with its attributes, including primary keys, and relationships are depicted with lines connecting related tables.

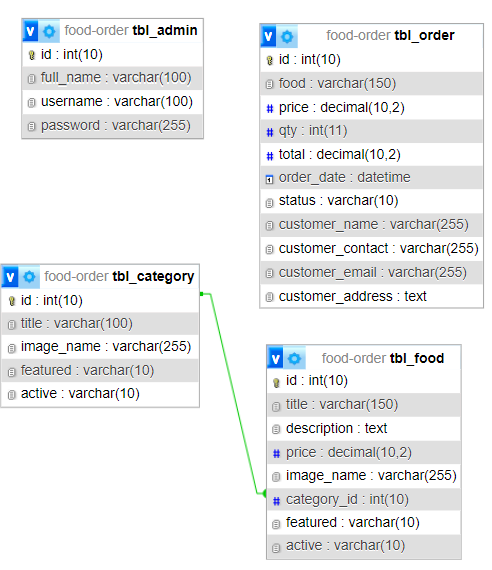


Figure 1:

**Table Details**

**Table: tbl\_food**

**Description**: This table stores details of individual food items available in our system.

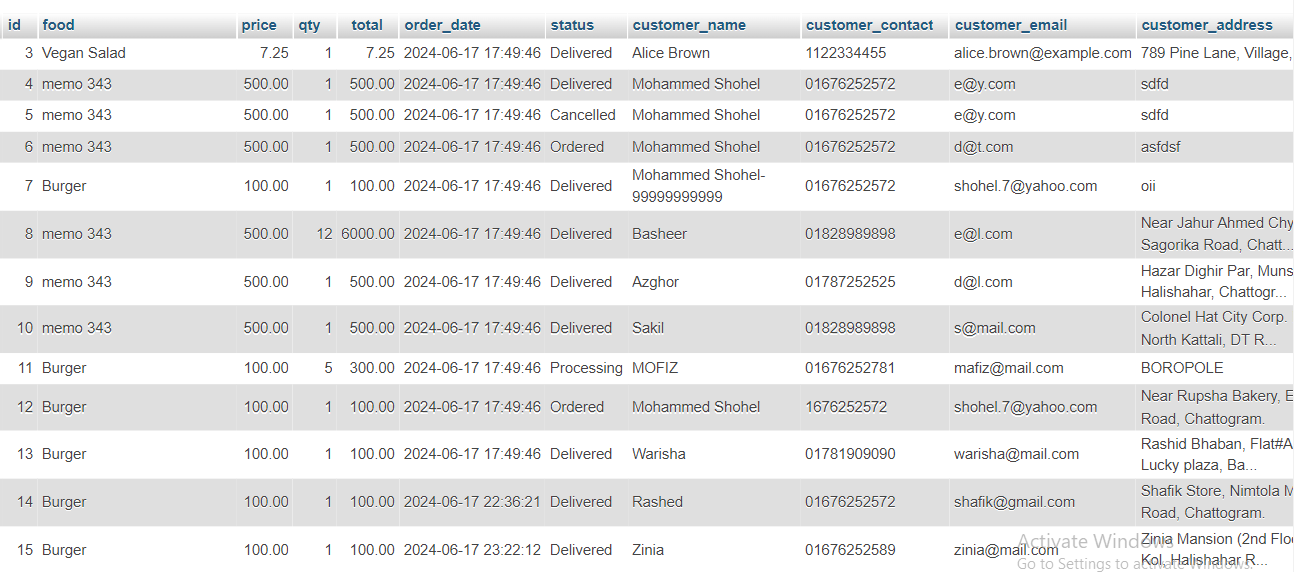
**Attributes**:

* id (PK): Unique identifier for each food item.
* title: Name of the food item.
* description: Brief description of the food item.
* price: Price of the food item.
* image\_name: Name of the image associated with the food item.
* category\_id (FK): Foreign key referencing the category of the food item (tbl\_category.id).
* featured: Flag indicating if the food item is featured.
* active: Flag indicating if the food item is active and available for ordering.

**Relationships**:

* category\_id references tbl\_category.id (Many-to-One relationship).

**Example Data**:



#### **Normalization**

Our database design follows Third Normal Form (3NF) to ensure data integrity and minimize redundancy. Each table is organized to eliminate duplicative data and potential anomalies.

#### **Indexes and Constraints**

We have implemented indexes on frequently queried columns to optimize performance. Additionally, constraints such as primary keys, foreign keys, and unique constraints are applied to maintain data consistency.

#### **Database Management System (DBMS)**

We are utilizing MySQL version 8.0 as our database management system. MySQL was chosen for its robust features and compatibility with our project requirements.

#### **Backup and Recovery**

To ensure data durability, we have established regular backup procedures using MySQL's built-in tools for scheduled backups and recovery.

#### **Security**

Database security measures include role-based access control and encryption of sensitive data fields to protect against unauthorized access and data breaches.

#### **Conclusion**

In conclusion, this database design documentation provides a comprehensive overview of our database schema, emphasizing its role in supporting the functionality and scalability of our project. By adhering to best practices in database design and management, we aim to maintain efficient data handling and ensure the reliability of our application.

* **9.2 Screenshots**

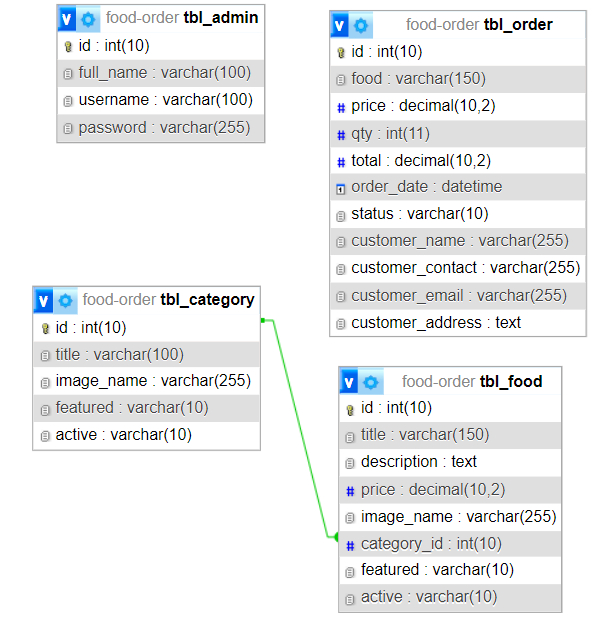
### **9.2 Screenshots: Visual Representation of Key Application Screens**

In this section, we provide visual representations of key application screens for both server-side and client-side components. Each screenshot is annotated to highlight important features and functionalities.

#### **Server-Side Components**

**Figure 1: Server-Side Architecture Overview**

**Annotations:**

* **Database Schema (ERD)**: Illustrates the structure of our database with entities and relationships.
* **Server Configuration**: Shows the setup details including server specifications and environment variables.

// Create constants to store non-repeating values

define('SITEURL', 'http://localhost/food-order/');

define('LOCALHOST', 'localhost');

define('DB\_USERNAME', 'root');

define('DB\_PASSWORD', '');

define('DB\_NAME', 'food-order');

// Database connection

$conn = mysqli\_connect(LOCALHOST, DB\_USERNAME, DB\_PASSWORD) or die(mysqli\_error($conn));

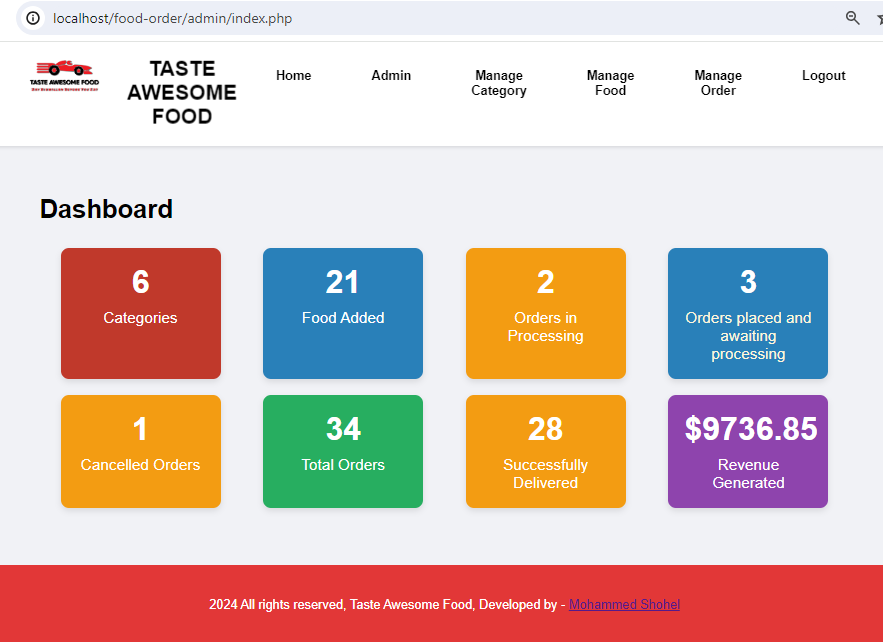
// Selecting database

$db\_select = mysqli\_select\_db($conn, DB\_NAME) or die(mysqli\_error($conn));

#### **Client-Side Components**

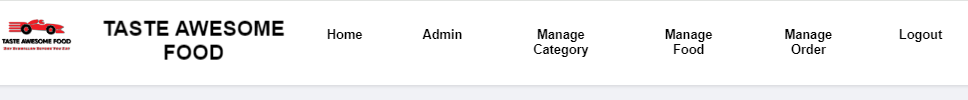
**Annotations:**

* **Dashboard Overview**: Provides an overview of the client dashboard interface.



**Figure 2: Client Dashboard Interface**

* **Navigation Bar**: Shows navigation options for different sections of the application.



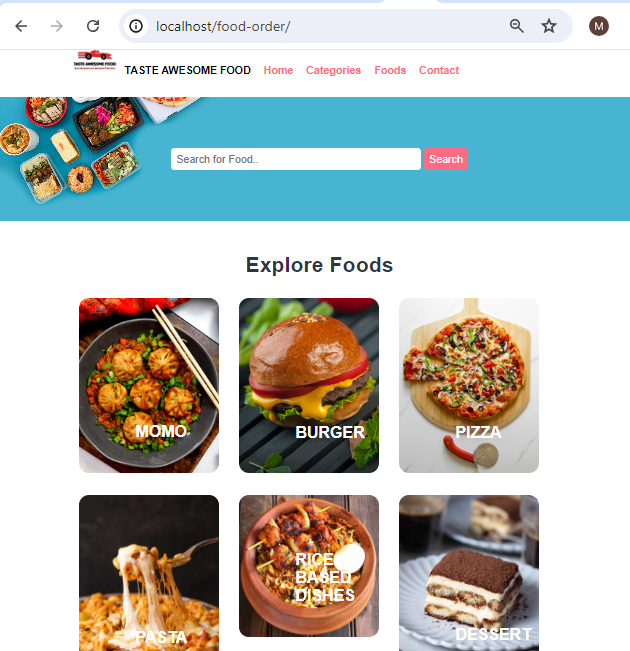
* **Data Visualization**: Includes charts and graphs displaying real-time data.

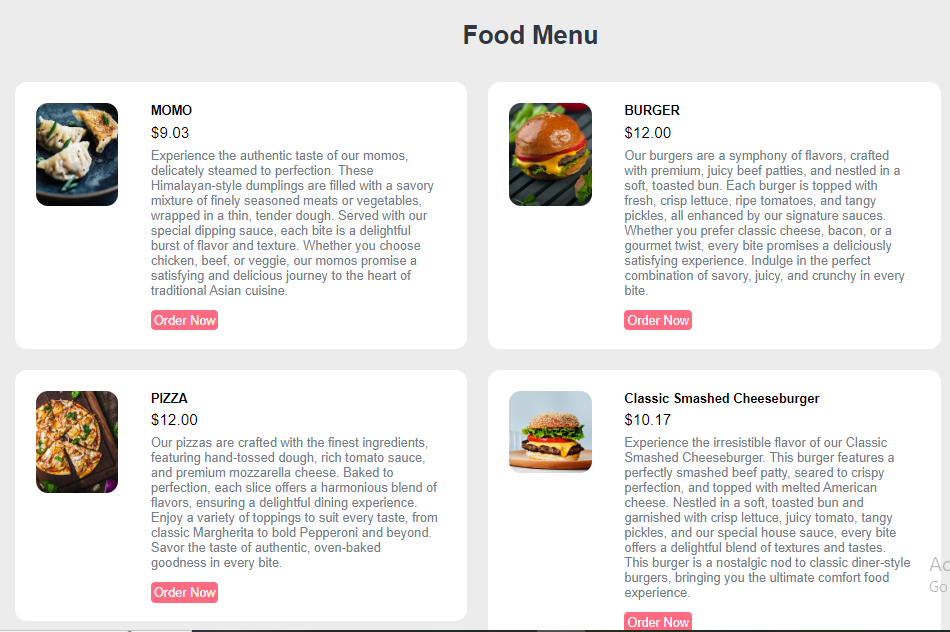


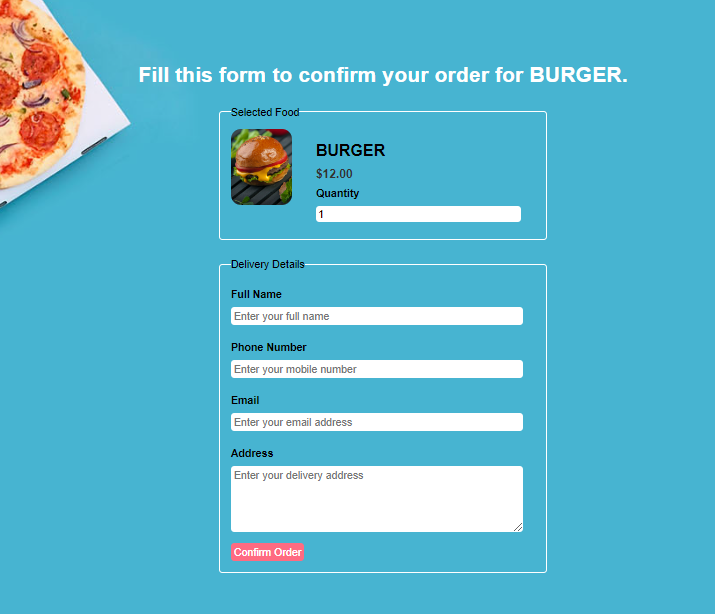
#### **Frontend**

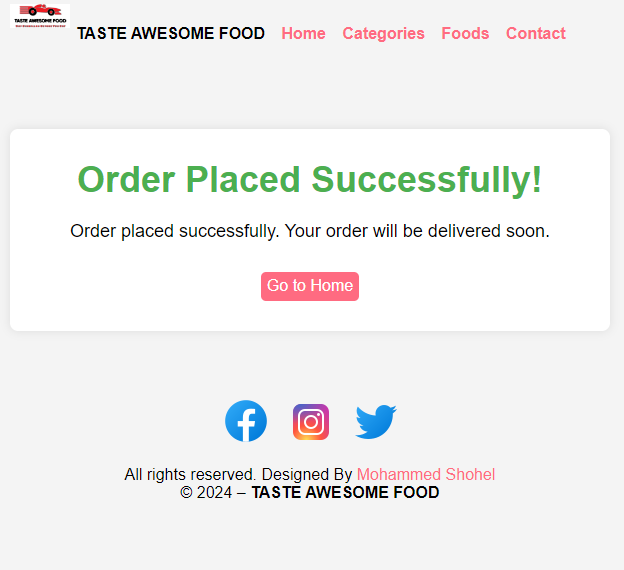
**Figure 3: Frontend Component**

**Annotations:**

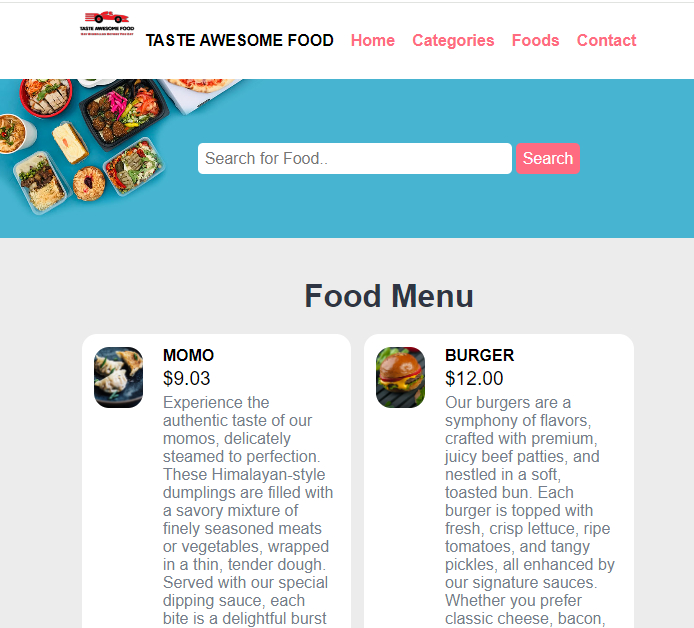
* **User Interface Elements**: Annotations highlight UI components such as buttons, forms, and dropdowns.



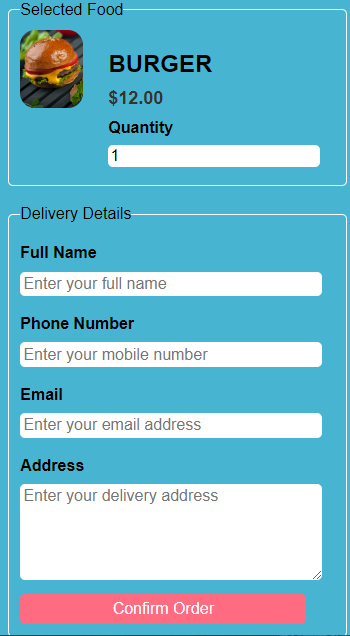


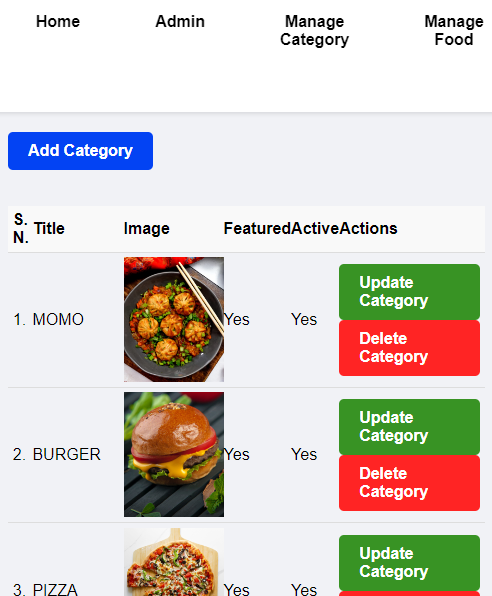


* **Responsive Design**: Demonstrates how the frontend adapts to different screen sizes.



* **Interactive Features**: Shows interactive elements like sliders or maps.



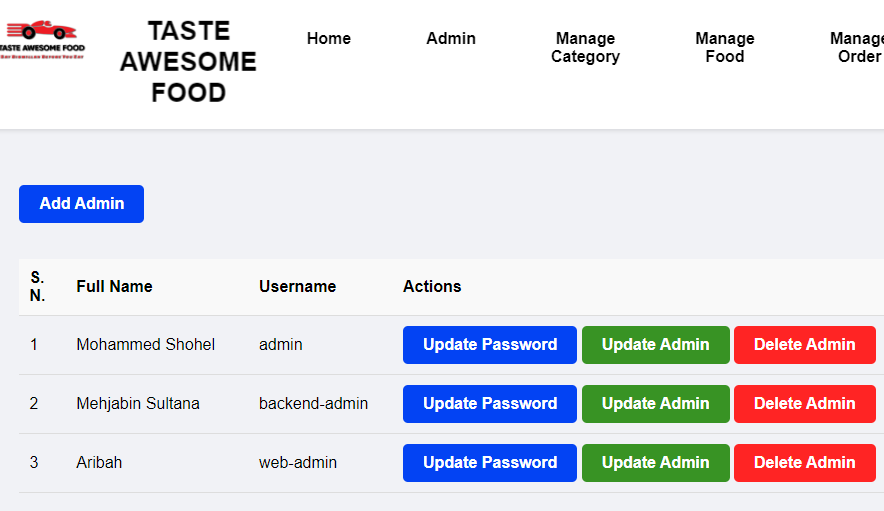


#### **Backend**

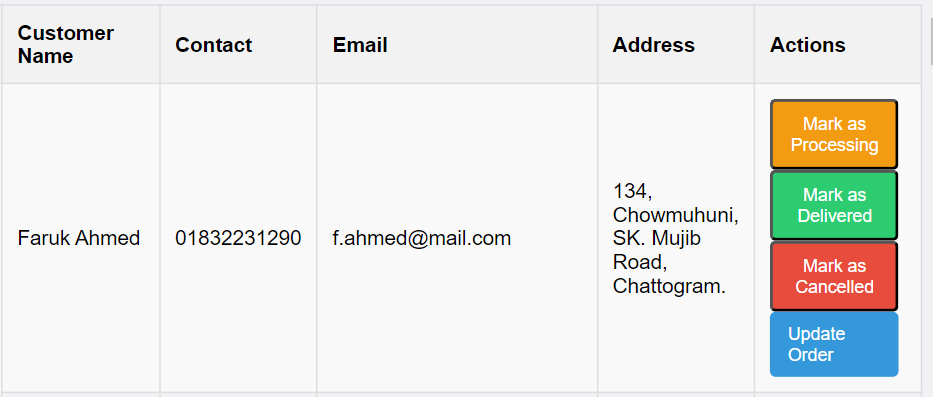
**Figure 4: Backend Management Interface**

**Annotations:**

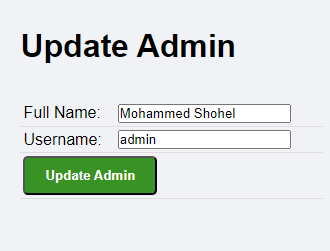
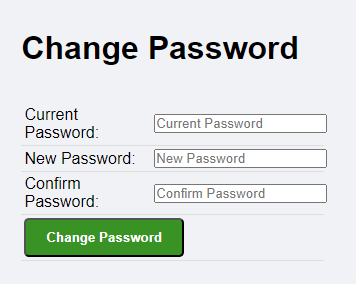
* **Admin Panel**: Annotates the backend interface used for managing application settings.

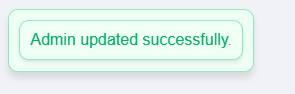
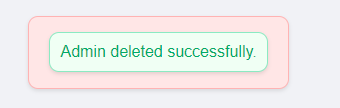


* **Database Management**: Shows tools and functionalities for database administration.

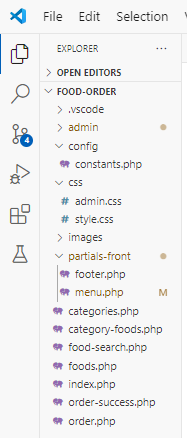
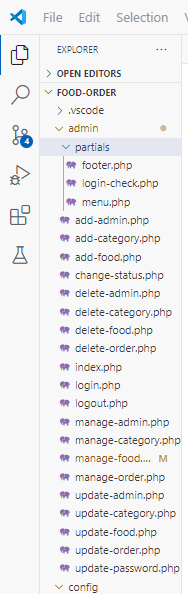


* **Security Features**: Highlights features such as user roles and access controls.





* **9.3 Source Code**
  + Overview of key source code files and modules.
  + Code snippets illustrating critical functionalities.





<?php include('partials/menu.php'); ?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Add Admin</h1>

<br><br>

<?php

if (isset($\_SESSION['add'])) {

if (stripos($\_SESSION['add'], 'Successfully') !== false) {

echo "<div class='success'>" . $\_SESSION['add'] . "</div>";

} else {

echo "<div class='error'>" . $\_SESSION['add'] . "</div>";

echo "<meta http-equiv='refresh' content='4;url=" . SITEURL . "admin/manage-admin.php'>";

}

unset($\_SESSION['add']);

}

?>

<form action="" method="POST">

<table class="tbl-30">

<tr>

<td>Full Name:</td>

<td>

<input type="text" name="full\_name" placeholder="Enter your full name">

</td>

</tr>

<tr>

<td>Username:</td>

<td>

<input type="text" name="username" placeholder="Enter your username">

</td>

</tr>

<tr>

<td>Password:</td>

<td>

<input type="password" name="password" placeholder="Enter your password">

</td>

</tr>

<tr>

<td colspan="2">

<input type="submit" name="submit" value="Add Admin" class="btn-secondary">

</td>

</tr>

</table>

</form>

</div>

</div>

<!-- Main Content ends here -->

<?php include('partials/footer.php'); ?>

<?php

// Process the value from form and save it in the database

// Check whether the submit button is clicked or not

if (isset($\_POST['submit'])) {

// Button clicked

// Get the data from form

$full\_name = $\_POST['full\_name'];

$username = $\_POST['username'];

$password = md5($\_POST['password']); // Password encryption with md5

// SQL query to check if the username exists

$sql\_check = "SELECT \* FROM tbl\_admin WHERE username='$username'";

$res\_check = mysqli\_query($conn, $sql\_check);

if (mysqli\_num\_rows($res\_check) > 0) {

// Username exists

$\_SESSION['add'] = "Username already exists. Please choose a different username.";

// Redirect to add admin page

header("location:" . SITEURL . 'admin/add-admin.php');

} else {

// SQL query to save the data into the database

$sql = "INSERT INTO tbl\_admin SET

full\_name='$full\_name',

username='$username',

password='$password'";

// Execute query and save data into database

$res = mysqli\_query($conn, $sql) or die(mysqli\_error($conn));

// Check whether the query is executed and data is inserted or not

if ($res == TRUE) {

// Data inserted

$\_SESSION['add'] = "Admin Added Successfully";

// Redirect to manage admin page

header("location:" . SITEURL . 'admin/manage-admin.php');

} else {

// Failed to insert data

$\_SESSION['add'] = "Failed to Add Admin";

// Redirect to add admin page

header("location:" . SITEURL . 'admin/add-admin.php');

}

}

}

?>



<?php

// Start session

session\_start();

// Include necessary files

include('partials/menu.php');

// Process form submission

if(isset($\_POST['submit'])) {

// Get form data

$title = $\_POST['title'];

// Validate title

if(empty($title)) {

$\_SESSION['add-fail'] = "<div class='error'>Title cannot be blank.</div>";

header('location:'.SITEURL.'admin/add-category.php');

exit();

}

// Handle image upload

if(isset($\_FILES['image']['name']) && $\_FILES['image']['name'] != "") {

$image\_name = $\_FILES['image']['name'];

$image\_tmp = $\_FILES['image']['tmp\_name'];

$upload\_path = "../images/category/";

$uploaded = move\_uploaded\_file($image\_tmp, $upload\_path.$image\_name);

if(!$uploaded) {

$\_SESSION['upload'] = "<div class='error'>Failed to upload image.</div>";

header('location:'.SITEURL.'admin/add-category.php');

exit();

}

} else {

$\_SESSION['upload'] = "<div class='error'>Image not selected.</div>";

header('location:'.SITEURL.'admin/add-category.php');

exit();

}

// Process radio inputs

$featured = isset($\_POST['featured']) ? $\_POST['featured'] : "No";

$active = isset($\_POST['active']) ? $\_POST['active'] : "No";

// Insert into database (assuming $conn is your database connection)

$sql = "INSERT INTO tbl\_category (title, image\_name, featured, active) VALUES ('$title', '$image\_name', '$featured', '$active')";

$res = mysqli\_query($conn, $sql);

if($res) {

$\_SESSION['add'] = "<div class='success'>Category Added Successfully.</div>";

header('location:'.SITEURL.'admin/manage-category.php');

exit();

} else {

$\_SESSION['add'] = "<div class='error'>Failed to Add Category.</div>";

header('location:'.SITEURL.'admin/add-category.php');

exit();

}

}

?>

<!-- HTML content -->

<div class="main-content">

<div class="wrapper">

<h1>Add Category</h1>

<br><br>

<!-- Display session messages -->

<?php

if(isset($\_SESSION['add'])) {

echo $\_SESSION['add'];

unset($\_SESSION['add']);

}

if(isset($\_SESSION['upload'])) {

echo $\_SESSION['upload'];

unset($\_SESSION['upload']);

}

if(isset($\_SESSION['add-fail'])) {

echo $\_SESSION['add-fail'];

unset($\_SESSION['add-fail']);

}

?>

<br><br>

<!-- Add category form starts here -->

<form action="" method="POST" enctype="multipart/form-data">

<table class="tbl-30">

<tr>

<td>Title: </td>

<td>

<input type="text" name="title" placeholder="Category Title">

</td>

</tr>

<tr>

<td>Select Image:</td>

<td>

<input type="file" name="image">

</td>

</tr>

<tr>

<td>Featured: </td>

<td>

<input type="radio" name="featured" value="Yes"> Yes

<input type="radio" name="featured" value="No" checked> No

</td>

</tr>

<tr>

<td>Active: </td>

<td>

<input type="radio" name="active" value="Yes"> Yes

<input type="radio" name="active" value="No" checked> No

</td>

</tr>

<tr>

<td colspan="2">

<input type="submit" name="submit" value="Add Category" class="btn-secondary">

</td>

</tr>

</table>

</form>

<!-- Add category form ends here -->

</div>

</div>

<!-- Include footer.php at the bottom -->

<?php include('partials/footer.php'); ?>



<?php

session\_start(); // Start session to use session variables

include('partials/menu.php');

?>

<div class="main-content">

<div class="wrapper">

<h1>Add Food</h1>

<br><br>

<?php

if (isset($\_SESSION['upload'])) {

echo $\_SESSION['upload'];

unset($\_SESSION['upload']); // Clear session after displaying

}

if (isset($\_SESSION['add'])) {

echo $\_SESSION['add'];

unset($\_SESSION['add']); // Clear session after displaying

}

?>

<form action="" method="post" enctype="multipart/form-data">

<table class="tbl-30">

<tr>

<td>Title:</td>

<td>

<input type="text" name="title" placeholder="Title of the food" required>

</td>

</tr>

<tr>

<td>Description:</td>

<td>

<textarea name="description" cols="30" rows="5" placeholder="Description of the food" required></textarea>

</td>

</tr>

<tr>

<td>Price:</td>

<td>

<input type="number" name="price" step="0.01" required>

</td>

</tr>

<tr>

<td>Select Image:</td>

<td>

<input type="file" name="image" required>

</td>

</tr>

<tr>

<td>Category:</td>

<td>

<select name="category" required>

<option value="" selected disabled>Select Category</option>

<?php

$sql = "SELECT \* FROM tbl\_category WHERE active='Yes'";

$res = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($res) > 0) {

while ($row = mysqli\_fetch\_assoc($res)) {

$id = $row['id'];

$title = $row['title'];

echo "<option value='$id'>$title</option>";

}

} else {

echo "<option value='0'>No Category Found</option>";

}

?>

</select>

</td>

</tr>

<tr>

<td>Featured:</td>

<td>

<input type="radio" name="featured" value="Yes" required> Yes

<input type="radio" name="featured" value="No" required> No

</td>

</tr>

<tr>

<td>Active:</td>

<td>

<input type="radio" name="active" value="Yes" required> Yes

<input type="radio" name="active" value="No" required> No

</td>

</tr>

<tr>

<td colspan="2">

<input type="submit" name="submit" value="Add Food" class="btn-secondary">

</td>

</tr>

</table>

</form>

<?php

if (isset($\_POST['submit'])) {

// Function to sanitize and validate inputs

function sanitize\_input($data)

{

global $conn; // Access the global connection variable

$data = mysqli\_real\_escape\_string($conn, trim($data));

return $data;

}

// Validate and sanitize inputs

$title = sanitize\_input($\_POST['title']);

$description = sanitize\_input($\_POST['description']);

$price = floatval($\_POST['price']); // Convert to float

$category = intval($\_POST['category']); // Convert to integer

$featured = isset($\_POST['featured']) ? sanitize\_input($\_POST['featured']) : "No";

$active = isset($\_POST['active']) ? sanitize\_input($\_POST['active']) : "No";

// Validate image upload

if ($\_FILES['image']['error'] === UPLOAD\_ERR\_OK) {

$file\_name = $\_FILES['image']['name'];

$file\_tmp = $\_FILES['image']['tmp\_name'];

$file\_size = $\_FILES['image']['size'];

$file\_type = $\_FILES['image']['type'];

// Check file size (max 5MB)

$max\_file\_size = 5 \* 1024 \* 1024; // 5 MB in bytes

if ($file\_size > $max\_file\_size) {

$\_SESSION['upload'] = "<div class='error'>File size exceeds maximum limit (5MB).</div>";

header('location:' . SITEURL . 'admin/add-food.php');

exit();

}

// Ensure uploaded file is an image

$allowed\_types = array('jpg', 'jpeg', 'png', 'gif');

$file\_ext = strtolower(pathinfo($file\_name, PATHINFO\_EXTENSION));

if (!in\_array($file\_ext, $allowed\_types)) {

$\_SESSION['upload'] = "<div class='error'>Only JPG, JPEG, PNG, GIF files are allowed.</div>";

header('location:' . SITEURL . 'admin/add-food.php');

exit();

}

// Generate a unique file name to prevent conflicts

$new\_file\_name = "Food\_" . uniqid() . "." . $file\_ext;

// Destination path for the uploaded file

$upload\_path = "../images/food/" . $new\_file\_name;

// Move uploaded file to destination directory

if (!move\_uploaded\_file($file\_tmp, $upload\_path)) {

$\_SESSION['upload'] = "<div class='error'>Failed to upload image. Please try again.</div>";

header('location:' . SITEURL . 'admin/add-food.php');

exit();

}

} else {

$\_SESSION['upload'] = "<div class='error'>Error uploading image: " . $\_FILES['image']['error'] . "</div>";

header('location:' . SITEURL . 'admin/add-food.php');

exit();

}

// Insert into database using prepared statement to prevent SQL injection

$sql = "INSERT INTO tbl\_food (title, description, price, image\_name, category\_id, featured, active)

VALUES (?, ?, ?, ?, ?, ?, ?)";

$stmt = mysqli\_stmt\_init($conn);

if (!mysqli\_stmt\_prepare($stmt, $sql)) {

$\_SESSION['add'] = "<div class='error'>SQL Error: " . mysqli\_error($conn) . "</div>";

header('location:' . SITEURL . 'admin/add-food.php');

exit();

} else {

mysqli\_stmt\_bind\_param($stmt, "ssdssss", $title, $description, $price, $new\_file\_name, $category, $featured, $active);

mysqli\_stmt\_execute($stmt);

// Check whether data inserted or not

if (mysqli\_stmt\_affected\_rows($stmt) > 0) {

$\_SESSION['add'] = "<div class='success'>Food added successfully.</div>";

header('location:' . SITEURL . 'admin/manage-food.php');

exit();

} else {

$\_SESSION['add'] = "<div class='error'>Failed to add food.</div>";

header('location:' . SITEURL . 'admin/add-food.php');

exit();

}

}

}

?>

</div>

</div>

<?php include('partials/footer.php'); ?>



<?php

include('../config/constants.php');

if (isset($\_GET['id']) && isset($\_GET['status'])) {

$id = $\_GET['id'];

$status = $\_GET['status'];

// Validate status to avoid any unauthorized or incorrect input

$validStatuses = array('Ordered', 'Processing', 'Delivered', 'Cancelled');

if (!in\_array($status, $validStatuses)) {

$\_SESSION['error'] = "Invalid status.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

// Update order status in the database

$sql = "UPDATE tbl\_order SET status = '$status' WHERE id = $id";

$res = mysqli\_query($conn, $sql);

if ($res) {

$\_SESSION['update'] = "Order status updated successfully.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

} else {

$\_SESSION['error'] = "Failed to update order status.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

} else {

$\_SESSION['error'] = "Invalid request.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

?>



<?php

// Check if a session is not already started

if (session\_status() == PHP\_SESSION\_NONE) {

// If not, start the session

session\_start();

}

// Include constants.php file here

include('../config/constants.php');

// Check if the 'id' parameter is set in the URL

if(isset($\_GET['id'])) {

// Get the id of admin to be deleted

$id = $\_GET['id'];

// Validate input: Ensure that $id is a non-empty integer

if(!empty($id) && is\_numeric($id)) {

// Create SQL query to delete admin

$sql = "DELETE FROM tbl\_admin WHERE id=$id";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check whether the query executed successfully or not

if($res) {

// Query executed successfully and admin deleted

$\_SESSION['delete'] = "<div class='success'>Admin deleted successfully.</div>";

} else {

// Failed to delete admin

$\_SESSION['delete'] = "<div class='error'>Failed to delete Admin. Try again later</div>";

}

} else {

// Invalid or missing 'id' parameter

$\_SESSION['delete'] = "<div class='error'>Invalid or missing 'id' parameter</div>";

}

} else {

// 'id' parameter is not set in the URL

$\_SESSION['delete'] = "<div class='error'>ID parameter missing</div>";

}

// Redirect to manage admin page with message (success/error)

header('Location: '.SITEURL.'admin/manage-admin.php');

exit(); // added exit() to prevent further execution

?>



<?php

// Include constants file

include('../config/constants.php');

// Delete category

// Check whether the id and image\_name values are set or not

if (isset($\_GET['id']) and isset($\_GET['image\_name'])) {

// Get the value and delete

$id = $\_GET['id'];

$image\_name = $\_GET['image\_name'];

// Remove the physical image file if available

if ($image\_name != "") {

// Image is available, so remove it

$path = "../images/Category/" . $image\_name;

// Remove the image

$remove = unlink($path);

// If failed to remove image then add an error message and stop the process

if ($remove == false) {

// Set the session message

$\_SESSION['remove'] = "<div class='error'>Failed to remove category image. </div>";

// Redirect to manage category page

header('location:' . SITEURL . 'admin/manage-category.php');

// Stop the process

die();

}

}

// Delete data from database

// SQL query to delete data from database

$sql = "DELETE FROM tbl\_category WHERE id=$id";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check whether the data is deleted from database or not

if ($res == true) {

// Set success message and redirect

$\_SESSION['delete'] = "<div class='success'>Category deleted successfully.</div>";

// Redirect to manage category page

header('location:' . SITEURL . 'admin/manage-category.php');

} else {

// Set fail message and redirect

$\_SESSION['delete'] = "<div class='error'>Failed to delete category. Try again later.</div>";

// Redirect to manage category page

header('location:' . SITEURL . 'admin/manage-category.php');

}

} else {

// Redirect to manage category page

header('location:' . SITEURL . 'admin/manage-category.php');

}

?>



<?php

session\_start();

include('partials/menu.php');

// Check if the ID and image\_name are set in the URL

if (isset($\_GET['id']) && isset($\_GET['image\_name'])) {

// Get the ID and image name

$id = $\_GET['id'];

$image\_name = $\_GET['image\_name'];

// Remove the physical image file if it exists

if ($image\_name != "") {

// Image path

$path = "../images/food/" . $image\_name;

// Remove the image

$remove = unlink($path);

// If failed to remove image then add an error message and stop the process

if ($remove == false) {

// Set the session message

$\_SESSION['upload'] = "<div class='error'>Failed to remove image file.</div>";

// Redirect to Manage Food page

header('location:' . SITEURL . 'admin/manage-food.php');

// Stop the process

exit();

}

}

// Delete food from database

$sql = "DELETE FROM tbl\_food WHERE id=$id";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check whether the query executed successfully

if ($res == true) {

// Set success message and redirect

$\_SESSION['delete'] = "<div class='success'>Food deleted successfully.</div>";

header('location:' . SITEURL . 'admin/manage-food.php');

} else {

// Set failure message and redirect

$\_SESSION['delete'] = "<div class='error'>Failed to delete food. Try again later.</div>";

header('location:' . SITEURL . 'admin/manage-food.php');

}

} else {

// Redirect to Manage Food page if ID or image\_name not set

$\_SESSION['unauthorized'] = "<div class='error'>Unauthorized access.</div>";

header('location:' . SITEURL . 'admin/manage-food.php');

}

?>



<?php

include('partials/menu.php');

// Start session if not already started

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

if (isset($\_POST['submit'])) {

// Check if id is set and valid

if (isset($\_POST['id']) && is\_numeric($\_POST['id'])) {

$id = $\_POST['id'];

// Attempt to delete order

$sql = "DELETE FROM tbl\_order WHERE id = $id";

$result = mysqli\_query($conn, $sql);

if ($result) {

// Deletion successful

$\_SESSION['success'] = "Order deleted successfully.";

} else {

// Deletion failed

$\_SESSION['error'] = "Failed to delete order. Please try again.";

}

// Redirect to manage-order.php after deletion attempt

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

} else {

// Redirect to manage-order.php if id is missing or invalid

$\_SESSION['error'] = "Invalid order ID.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

} else if (isset($\_GET['id']) && is\_numeric($\_GET['id'])) {

$id = $\_GET['id'];

} else {

// Redirect to manage-order.php if id is not set or invalid in GET request

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

?>

<!-- HTML content for delete-order.php with confirmation message -->

<div class="main-content">

<div class="wrapper">

<h1>Delete Order</h1>

<br><br>

<?php if (isset($\_SESSION['error'])) : ?>

<div class="error"><?php echo $\_SESSION['error']; ?></div>

<?php unset($\_SESSION['error']); ?>

<?php endif; ?>

<p>Are you sure you want to delete this order?</p>

<!-- Form for confirmation of deletion -->

<form action="<?php echo htmlspecialchars($\_SERVER['PHP\_SELF']); ?>" method="POST">

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Delete Order" class="btn btn-danger">

<a href="<?php echo SITEURL; ?>admin/manage-order.php" class="btn btn-secondary">Cancel</a>

</form>

</div>

</div>

<?php include('partials/footer.php'); ?>



<?php

// Include the menu file which might contain header information or navigation

// Include the constants file for database connection

include('partials/menu.php');

// SQL queries to fetch data

$sqlCategory = "SELECT COUNT(id) AS total\_categories FROM tbl\_category";

$sqlFood = "SELECT COUNT(id) AS total\_foods FROM tbl\_food";

$sqlTotalOrders = "SELECT COUNT(id) AS total\_orders FROM tbl\_order";

$sqlProcessingOrders = "SELECT COUNT(id) AS processing\_orders FROM tbl\_order WHERE status = 'Processing'";

$sqlDeliveredOrders = "SELECT COUNT(id) AS delivered\_orders FROM tbl\_order WHERE status = 'Delivered'";

$sqlOrderedOrders = "SELECT COUNT(id) AS ordered\_orders FROM tbl\_order WHERE status = 'Ordered'";

$sqlCancelledOrders = "SELECT COUNT(id) AS cancelled\_orders FROM tbl\_order WHERE status = 'Cancelled'";

$sqlRevenue = "SELECT SUM(total) AS total\_revenue FROM tbl\_order WHERE status = 'Delivered'";

// Execute SQL queries to fetch data

$resultCategory = mysqli\_query($conn, $sqlCategory);

$resultFood = mysqli\_query($conn, $sqlFood);

$resultTotalOrders = mysqli\_query($conn, $sqlTotalOrders);

$resultProcessingOrders = mysqli\_query($conn, $sqlProcessingOrders);

$resultDeliveredOrders = mysqli\_query($conn, $sqlDeliveredOrders);

$resultOrderedOrders = mysqli\_query($conn, $sqlOrderedOrders);

$resultCancelledOrders = mysqli\_query($conn, $sqlCancelledOrders);

$resultRevenue = mysqli\_query($conn, $sqlRevenue);

// Function to fetch count from query result

function fetchCount($result) {

global $conn;

if ($result && mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_assoc($result);

return $row[array\_key\_first($row)]; // Return the count value

}

return 0; // Return 0 if no rows found or query failed

}

// Function to fetch total revenue from query result

function fetchTotalRevenue($result) {

global $conn;

if ($result && mysqli\_num\_rows($result) > 0) {

$row = mysqli\_fetch\_assoc($result);

return $row['total\_revenue']; // Return the total revenue value

}

return 0; // Return 0 if no rows found or query failed

}

// Initialize variables to hold fetched data

$totalCategories = fetchCount($resultCategory);

$totalFoods = fetchCount($resultFood);

$totalOrders = fetchCount($resultTotalOrders);

$processingOrders = fetchCount($resultProcessingOrders);

$deliveredOrders = fetchCount($resultDeliveredOrders);

$orderedOrders = fetchCount($resultOrderedOrders);

$cancelledOrders = fetchCount($resultCancelledOrders);

$totalRevenue = fetchTotalRevenue($resultRevenue);

// Function to generate dynamic background color based on a string input

function generateColor($str) {

$colors = ['#2980b9', '#c0392b', '#27ae60', '#f39c12', '#8e44ad', '#2c3e50', '#16a085']; // Array of predefined colors

$index = crc32($str) % count($colors); // Generate an index based on the hash of the input string

return $colors[$index]; // Return color from the array based on the index

}

?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Dashboard</h1>

<div class="dashboard-cards">

<!-- Displaying different dashboard cards with dynamic styles -->

<div class="dashboard-card category" style="background-color: <?php echo generateColor('Categories'); ?>">

<h2 style="color: #fff;"><?php echo $totalCategories; ?></h2>

<p style="color: #fff;">Categories</p>

</div>

<div class="dashboard-card foods" style="background-color: <?php echo generateColor('Foods'); ?>">

<h2 style="color: #fff;"><?php echo $totalFoods; ?></h2>

<p style="color: #fff;">Food Added </p>

</div>

<div class="dashboard-card processing-orders" style="background-color: <?php echo generateColor('Processing Orders'); ?>">

<h2 style="color: #fff;"><?php echo $processingOrders; ?></h2>

<p style="color: #fff;">Orders in Processing</p>

</div>

<div class="dashboard-card ordered-orders" style="background-color: <?php echo generateColor('Ordered Orders'); ?>">

<h2 style="color: #fff;"><?php echo $orderedOrders; ?></h2>

<p style="color:cornsilk;">Orders placed and awaiting processing</p>

</div>

<div class="dashboard-card cancelled-orders" style="background-color: <?php echo generateColor('Cancelled Orders'); ?>">

<h2 style="color: #fff;"><?php echo $cancelledOrders; ?></h2>

<p style="color: #fff;">Cancelled Orders</p>

</div>

<div class="dashboard-card total-orders" style="background-color: <?php echo generateColor('Total Orders'); ?>">

<h2 style="color: #fff;"><?php echo $totalOrders; ?></h2>

<p style="color: #fff;">Total Orders</p>

</div>

<div class="dashboard-card delivered-orders" style="background-color: <?php echo generateColor('Delivered Orders'); ?>">

<h2 style="color: #fff;"><?php echo $deliveredOrders; ?></h2>

<p style="color: #fff;">Successfully Delivered</p>

</div>

<div class="dashboard-card revenue" style="background-color: <?php echo generateColor('Total Revenue'); ?>">

<h2 style="color: #fff;">$<?php echo $totalRevenue; ?></h2>

<p style="color: #fff;">Revenue Generated</p>

</div>

</div>

<div class="clearfix"></div> <!-- to fix floating categories in horizontal line -->

</div>

</div>

<!-- Main Content ends here -->

<?php include('partials/footer.php'); ?>

<style>

/\* Basic styling for dashboard cards \*/

.dashboard-cards {

display: flex;

justify-content: space-around;

flex-wrap: wrap;

margin-top: 20px;

}

.dashboard-card {

border-radius: 10px;

padding: 20px;

text-align: center;

width: 200px;

margin: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

transition: transform 0.3s ease-in-out, box-shadow 0.3s ease-in-out;

background-color: #fff;

}

.dashboard-card:hover {

transform: translateY(-8px);

box-shadow: 0 12px 20px rgba(0, 0, 0, 0.3);

}

.dashboard-card h2 {

font-size: 2.5rem;

margin: 0;

font-weight: 600;

}

.dashboard-card p {

font-size: 1.2rem;

margin-top: 10px;

color: #666;

}

.dashboard-card.category {

background-color: #2980b9; /\* Blue color \*/

}

.dashboard-card.foods {

background-color: #c0392b; /\* Red color \*/

}

.dashboard-card.total-orders {

background-color: #27ae60; /\* Green color \*/

}

.dashboard-card.processing-orders {

background-color: #f39c12; /\* Orange color \*/

}

.dashboard-card.delivered-orders {

background-color: #8e44ad; /\* Purple color \*/

}

.dashboard-card.ordered-orders {

background-color: #2c3e50; /\* Dark blue color \*/

}

.dashboard-card.cancelled-orders {

background-color: #16a085; /\* Turquoise color \*/

}

.dashboard-card.revenue {

background-color: #f1c40f; /\* Yellow color \*/

}

</style>



<?php

include('../config/constants.php'); // Adjust the path according to the actual location

// Start the session if not already started

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

// Check if the login button is clicked

if (isset($\_POST['login'])) {

$username = $\_POST['username'];

$password = md5($\_POST['password']);

// SQL query to check if the user with the given username and password exists

$sql = "SELECT \* FROM tbl\_admin WHERE username='$username' AND password='$password'";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check if the query was successful

if ($res) {

// Check if any rows were returned

$count = mysqli\_num\_rows($res);

if ($count == 1) {

// User exists and login is successful

$\_SESSION['login'] = "<div class='success text-center'>Login Successful</div>";

//to check whether the user is logged in or not and logout will unset it

$\_SESSION['user'] = $username;

// Redirect to the dashboard or desired page

header("Location:" . SITEURL . "admin/index.php");

exit(); // Exit to prevent further execution

} else {

// User does not exist, display error message or redirect back to login page

$\_SESSION['login'] = "<div class='error text-center'>Invalid Username or Password</div>";

header("Location:" . SITEURL . "admin/login.php");

exit(); // Exit to prevent further execution

}

} else {

// Query execution failed

echo "Error: " . mysqli\_error($conn);

}

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Login - Taste Awesome Food</title>

<link rel="stylesheet" href="../css/admin.css">

</head>

<body>

<div class="login">

<br><br>

<h1 class="text-center">Login</h1>

<?php

if (isset($\_SESSION['login'])) {

echo $\_SESSION['login'];

unset($\_SESSION['login']);

}

if (isset($\_SESSION['no-login-message'])) {

echo $\_SESSION['no-login-message'];

unset($\_SESSION['no-login-message']);

}

?>

<!-- Login Form Starts Here -->

<form action="" method="POST" class="text-center">

<label for="username">Username:</label> <br>

<input type="text" id="username" name="username" placeholder="Enter Username" required><br><br>

<label for="password">Password:</label> <br>

<input type="password" id="password" name="password" placeholder="Enter Password" required> <br><br>

<input type="submit" name="login" value="Login" class="btn-primary">

<br><br>

</form>

<!-- Login Form Ends Here -->

<p class="text-center">Developed by - <a href="#">Mohammed Shohel</a></p>

</div>

</body>

</html>



<?php

// Include constants.php for SITEURL

include('../config/constants.php');

// 1. Destroy the session

session\_destroy(); //unset $\_session ['user'] and it will logout our system

// 2. Redirect to login page

header('Location: ' . SITEURL . 'admin/login.php');

exit(); // It's a good practice to call exit after a header redirect

?>



<?php

// Check if session is not already started

if (session\_status() == PHP\_SESSION\_NONE) {

// If not, start session

session\_start();

}

// Include the menu

include('partials/menu.php');

?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Manage Admin</h1>

<br><br>

<div>

<?php

// Check if session variables for add, delete, update, user-not-found, pwd-not-match, change-pwd are set and display messages if they are

$session\_vars = ['add', 'delete', 'update', 'user-not-found', 'pwd-not-match', 'change-pwd'];

foreach ($session\_vars as $var) {

if (isset($\_SESSION[$var])) {

// Assign appropriate class based on the session variable

$class = ($var == 'add') ? 'message-info' : (($var == 'update' || $var == 'change-pwd') ? 'success' : 'error');

echo "<div class='$class'>" . $\_SESSION[$var] . "</div>"; // displaying session message with class

unset($\_SESSION[$var]); // Clear session message after displaying it

}

}

?>

</div>

<br><br><br>

<!-- Button to add admin -->

<a href="add-admin.php" class="btn-primary">Add Admin</a>

<br><br><br>

<table class="tbl-full">

<tr>

<th>S. N.</th>

<th>Full Name</th>

<th>Username</th>

<th>Actions</th>

</tr>

<?php

// Database connection

$conn = mysqli\_connect('localhost', 'root', '', 'food-order') or die(mysqli\_error($conn));

// Query to get all admins

$sql = "SELECT \* FROM tbl\_admin";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check whether the query is executed or not

if ($res) {

// Count rows to check whether we have data in the database or not

$count = mysqli\_num\_rows($res); // Function to get all the rows in the database

$sn = 1; // Create a variable and assign the value

// Check the number of rows

if ($count > 0) {

// We have data in the database

while ($rows = mysqli\_fetch\_assoc($res)) {

// Using while loop to get all the data from the database

// and while loop will run as long as we have data in the database

// Get individual data

$id = $rows['id'];

$full\_name = $rows['full\_name'];

$username = $rows['username'];

// Display the values in our table

?>

<tr>

<td><?php echo $sn++; ?></td>

<td><?php echo $full\_name; ?></td>

<td><?php echo $username; ?></td>

<td>

<a href="<?php echo SITEURL; ?>admin/update-password.php?id=<?php echo $id; ?>" class="btn-primary">Update Password</a>

<a href="<?php echo SITEURL; ?>admin/update-admin.php?id=<?php echo $id; ?>" class="btn-secondary">Update Admin</a>

<a href="<?php echo SITEURL; ?>admin/delete-admin.php?id=<?php echo $id; ?>" class="btn-danger">Delete Admin</a>

</td>

</tr>

<?php

}

} else {

// We do not have data in the database

?>

<tr>

<td colspan="4">No Admins Added Yet.</td>

</tr>

<?php

}

} else {

// Query failed

echo "<tr><td colspan='4' class='error'>Failed to Retrieve Admins.</td></tr>";

}

?>

</table>

</div>

</div>

<!-- Main Content ends here -->

<?php include('partials/footer.php'); ?>



<?php include('partials/menu.php'); ?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Manage Category</h1>

<br><br>

<?php

// Display session messages if set and then unset them

$session\_variables = ['add', 'remove', 'delete', 'no-category-found', 'update', 'failed-remove'];

foreach ($session\_variables as $session\_variable) {

if (isset($\_SESSION[$session\_variable])) {

echo $\_SESSION[$session\_variable];

unset($\_SESSION[$session\_variable]);

}

}

?>

<br><br>

<!-- Button to add category -->

<a href="<?php echo SITEURL; ?>admin/add-category.php" class="btn-primary">Add Category</a>

<br><br><br>

<table class="tbl-full">

<tr>

<th>S. N.</th>

<th>Title</th>

<th>Image</th>

<th>Featured</th>

<th>Active</th>

<th>Actions</th>

</tr>

<?php

// Query to get all categories from the database

$sql = "SELECT \* FROM tbl\_category";

// Execute query

$res = mysqli\_query($conn, $sql);

// Count rows

$count = mysqli\_num\_rows($res);

// Check whether we have data in the database or not

if ($count > 0) {

// We have data in the database

// Get the data and display

$sn = 1; // Create a variable to store serial number

while ($row = mysqli\_fetch\_assoc($res)) {

$id = $row['id'];

$title = $row['title'];

$image\_name = $row['image\_name'];

$featured = $row['featured'];

$active = $row['active'];

?>

<tr>

<td><?php echo $sn++; ?>.</td>

<td><?php echo $title; ?></td>

<td>

<?php

// Check whether the image name is available or not

if ($image\_name != "") {

// Check if the file exists

$image\_path = "../images/category/" . $image\_name;

if (file\_exists($image\_path)) {

// Display the image

?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $image\_name; ?>" width="100px">

<?php

} else {

// Display the message

echo "<div class='error'>Image not found.</div>";

}

} else {

// Display the message

echo "<div class='error'>Image not added.</div>";

}

?>

</td>

<td><?php echo $featured; ?></td>

<td><?php echo $active; ?></td>

<td>

<a href="<?php echo SITEURL; ?>admin/update-category.php?id=<?php echo $id; ?>" class="btn-secondary">Update Category</a>

<a href="<?php echo SITEURL; ?>admin/delete-category.php?id=<?php echo $id; ?>&image\_name=<?php echo $image\_name; ?>" class="btn-danger">Delete Category</a>

</td>

</tr>

<?php

}

} else {

// We do not have data in the database

// We will display the message inside the table

?>

<tr>

<td colspan="6">

<div class="error">No Category Added</div>

</td>

</tr>

<?php

}

?>

</table>

</div>

</div>

<!-- Main Content ends here -->

<?php include('partials/footer.php'); ?>



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Manage Food</title>

<style>

.tbl-full {

width: 100%;

border-collapse: collapse;

margin: 20px 0;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

border-radius: 10px;

overflow: hidden;

}

.tbl-full th, .tbl-full td {

padding: 12px 15px;

border: 1px solid #ddd;

text-align: left;

}

.tbl-full th {

background-color: #007bff;

color: #ffffff;

text-transform: uppercase;

}

.tbl-full tr:nth-child(even) {

background-color: #f2f2f2;

}

.tbl-full tr:hover {

background-color: #f1f1f1;

}

.tbl-full td img {

border-radius: 5px;

}

.btn-primary, .btn-secondary, .btn-danger {

padding: 8px 12px;

text-decoration: none;

border-radius: 5px;

color: #fff;

margin-right: 5px;

}

.btn-primary {

background-color: #007bff;

}

.btn-primary:hover {

background-color: #0056b3;

}

.btn-secondary {

background-color: #6c757d;

}

.btn-secondary:hover {

background-color: #5a6268;

}

.btn-danger {

background-color: #dc3545;

}

.btn-danger:hover {

background-color: #c82333;

}

/\* Centered search bar styles \*/

.search-container {

display: flex;

justify-content: center;

margin: 20px 0;

}

.search-container input[type="text"],

.search-container select {

width: 300px;

padding: 10px;

border-radius: 5px 0 0 5px;

border: 1px solid #ddd;

outline: none;

}

.search-container input[type="submit"] {

padding: 10px 20px;

border: none;

border-radius: 0 5px 5px 0;

background-color: #007bff;

color: #fff;

cursor: pointer;

}

.search-container input[type="submit"]:hover {

background-color: #0056b3;

}

</style>

</head>

<body>

<?php include('partials/menu.php'); ?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Manage Food</h1>

<br>

<!-- Button to add food -->

<a href="<?php echo SITEURL; ?>admin/add-food.php" class="btn-primary">Add Food</a>

<br><br><br>

<!-- Search and filter form -->

<div class="search-container">

<form action="" method="POST">

<input type="text" name="search" placeholder="Search for food...">

<select name="featured">

<option value="">Select Featured</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

<select name="active">

<option value="">Select Active</option>

<option value="Yes">Yes</option>

<option value="No">No</option>

</select>

<input type="submit" name="submit" value="Search">

</form>

</div>

<br>

<?php

if (isset($\_SESSION['add'])) {

echo "<div class='success'>" . $\_SESSION['add'] . "</div>";

unset($\_SESSION['add']);

}

if (isset($\_SESSION['delete'])) {

echo "<div class='success'>" . $\_SESSION['delete'] . "</div>";

unset($\_SESSION['delete']);

}

if (isset($\_SESSION['update'])) {

echo "<div class='success'>" . $\_SESSION['update'] . "</div>";

unset($\_SESSION['update']);

}

?>

<table class="tbl-full">

<tr>

<th>S. N.</th>

<th>Title</th>

<th>Description</th>

<th>Price</th>

<th>Image</th>

<th>Category</th>

<th>Featured</th>

<th>Active</th>

<th>Actions</th>

</tr>

<?php

$sql = "SELECT f.\*, c.title AS category\_title FROM tbl\_food f INNER JOIN tbl\_category c ON f.category\_id = c.id";

// Check if the search form was submitted

if (isset($\_POST['submit'])) {

$search = mysqli\_real\_escape\_string($conn, $\_POST['search']);

$featured = mysqli\_real\_escape\_string($conn, $\_POST['featured']);

$active = mysqli\_real\_escape\_string($conn, $\_POST['active']);

$sql .= " WHERE (f.title LIKE '%$search%' OR f.description LIKE '%$search%' OR c.title LIKE '%$search%')";

if ($featured != "") {

$sql .= " AND f.featured='$featured'";

}

if ($active != "") {

$sql .= " AND f.active='$active'";

}

}

$res = mysqli\_query($conn, $sql);

$sn = 1;

if (mysqli\_num\_rows($res) > 0) {

while ($row = mysqli\_fetch\_assoc($res)) {

$id = $row['id'];

$title = $row['title'];

$description = $row['description'];

$price = $row['price'];

$image\_name = $row['image\_name'];

$category = $row['category\_title'];

$featured = $row['featured'];

$active = $row['active'];

?>

<tr>

<td><?php echo $sn++; ?>.</td>

<td><?php echo $title; ?></td>

<td><?php echo $description; ?></td>

<td><?php echo $price; ?></td>

<td>

<?php

if ($image\_name != "") {

echo "<img src='" . SITEURL . "images/food/$image\_name' width='100px'>";

} else {

echo "<div class='error'>Image not added.</div>";

}

?>

</td>

<td><?php echo $category; ?></td>

<td><?php echo $featured; ?></td>

<td><?php echo $active; ?></td>

<td>

<a href="<?php echo SITEURL; ?>admin/update-food.php?id=<?php echo $id; ?>" class="btn-secondary">Update Food</a>

<a href="<?php echo SITEURL; ?>admin/delete-food.php?id=<?php echo $id; ?>&image\_name=<?php echo $image\_name; ?>" class="btn-danger">Delete Food</a>

</td>

</tr>

<?php

}

} else {

echo "<tr><td colspan='9' class='error'>Food not found.</td></tr>";

}

?>

</table>

</div>

</div>

<!-- Main Content ends here -->

<?php include('partials/footer.php'); ?>

</body>

</html>



<?php

ob\_start(); // Start output buffering

include('partials/menu.php');

// Start session if not already started

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

// Function to sort orders by date/time

function sortByDateTime($orders, $sortBy = 'DESC') {

usort($orders, function($a, $b) use ($sortBy) {

$dateA = strtotime($a['order\_date']);

$dateB = strtotime($b['order\_date']);

if ($sortBy == 'DESC') {

return $dateB <=> $dateA;

} else {

return $dateA <=> $dateB;

}

});

return $orders;

}

// Function to filter orders by status

function filterByStatus($orders, $status) {

$filteredOrders = array\_filter($orders, function($order) use ($status) {

return strtolower($order['status']) == strtolower($status);

});

return $filteredOrders;

}

// Fetch all orders from the database

$sql = "SELECT \* FROM tbl\_order ORDER BY id DESC";

$res = mysqli\_query($conn, $sql);

$count = mysqli\_num\_rows($res);

$orders = []; // Array to store orders

if ($count > 0) {

while ($row = mysqli\_fetch\_assoc($res)) {

$orders[] = [

'id' => $row['id'],

'food' => $row['food'], // Replace 'food' with the actual column name from your database

'price' => $row['price'],

'qty' => $row['qty'],

'total' => $row['total'],

'order\_date' => $row['order\_date'],

'status' => $row['status'],

'customer\_name' => $row['customer\_name'],

'customer\_contact' => $row['customer\_contact'],

'customer\_email' => $row['customer\_email'],

'customer\_address' => $row['customer\_address']

];

}

}

// Handle sorting by date/time

if (isset($\_GET['sort']) && $\_GET['sort'] == 'datetime') {

$orders = sortByDateTime($orders);

}

// Handle filtering by status

if (isset($\_GET['status'])) {

$statusFilter = $\_GET['status'];

if ($statusFilter != 'all') {

$orders = filterByStatus($orders, $statusFilter);

}

}

// Handle search functionality

if (isset($\_POST['search'])) {

$searchQuery = mysqli\_real\_escape\_string($conn, $\_POST['search']);

$searchedOrders = [];

foreach ($orders as $order) {

// Customize search criteria based on your needs

if (stripos($order['customer\_name'], $searchQuery) !== false ||

stripos($order['id'], $searchQuery) !== false || // Added search by Order ID

stripos($order['customer\_contact'], $searchQuery) !== false ||

stripos($order['customer\_email'], $searchQuery) !== false ||

stripos($order['customer\_address'], $searchQuery) !== false ||

stripos($order['order\_date'], $searchQuery) !== false || // Added search by Order Date

stripos($order['status'], $searchQuery) !== false || // Added search by Status

stripos($order['food'], $searchQuery) !== false) { // Added search by Food

$searchedOrders[] = $order;

}

}

$orders = $searchedOrders;

}

// Handle actions (mark as processing, mark as delivered, etc.)

if (isset($\_POST['action']) && isset($\_POST['order\_id'])) {

$action = $\_POST['action'];

$orderId = $\_POST['order\_id'];

// Update order status based on action

switch ($action) {

case 'processing':

updateOrderStatus($orderId, 'Processing');

break;

case 'delivered':

updateOrderStatus($orderId, 'Delivered');

break;

case 'cancelled':

updateOrderStatus($orderId, 'Cancelled');

break;

// Add more cases as needed for other actions

default:

// Handle default case

break;

}

}

// Function to update order status

function updateOrderStatus($orderId, $status) {

global $conn;

$sql = "UPDATE tbl\_order SET status = '$status' WHERE id = $orderId";

$res = mysqli\_query($conn, $sql);

if ($res) {

$\_SESSION['update'] = "<div class='success'>Order status updated successfully.</div>";

} else {

$\_SESSION['update'] = "<div class='error'>Failed to update order status.</div>";

}

// Redirect to manage-order.php after updating

header("Location: manage-order.php");

exit();

}

?>

<div class="main-content">

<div class="wrapper">

<h1>Manage Orders</h1>

<br><br>

<?php

if(isset($\_SESSION['update'])) {

echo $\_SESSION['update'];

unset($\_SESSION['update']);

}

?>

<div style="margin-bottom: 10px;">

<a href="?sort=datetime" class="btn">Sort by Date/Time</a>

<form action="" method="GET" style="display: inline;">

<label for="status">Filter by Status:</label>

<select id="status" name="status" onchange="this.form.submit()">

<option value="all" <?php if (!isset($\_GET['status']) || $\_GET['status'] == 'all') echo 'selected'; ?>>All Statuses</option>

<option value="Ordered" <?php if (isset($\_GET['status']) && $\_GET['status'] == 'Ordered') echo 'selected'; ?>>Ordered</option>

<option value="Processing" <?php if (isset($\_GET['status']) && $\_GET['status'] == 'Processing') echo 'selected'; ?>>Processing</option>

<option value="Cancelled" <?php if (isset($\_GET['status']) && $\_GET['status'] == 'Cancelled') echo 'selected'; ?>>Cancelled</option>

<option value="Delivered" <?php if (isset($\_GET['status']) && $\_GET['status'] == 'Delivered') echo 'selected'; ?>>Delivered</option>

</select>

</form>

<!-- Search Form -->

<form action="" method="POST" style="display: inline;">

<label for="search" style="margin-right: 10px;">Search:</label>

<input type="text" id="search" name="search" placeholder="Search by Name, Order ID, Contact, Email, Address, Order Date, Status, Food" class="input-search">

<button type="submit" class="btn btn-secondary">Search</button>

</form>

</div>

<table class="tbl-full">

<tr>

<th>S.N.</th>

<th>Order ID</th>

<th>Food</th>

<th>Price</th>

<th>Qty</th>

<th>Total</th>

<th>Order Date</th>

<th>Status</th>

<th>Customer Name</th>

<th>Contact</th>

<th>Email</th>

<th>Address</th>

<th class="actions-column">Actions</th>

</tr>

<?php

$sn = 1; // Serial number initialization

if (!empty($orders)) {

foreach ($orders as $order) {

?>

<tr>

<td><?php echo $sn++; ?></td>

<td><?php echo $order['id']; ?></td>

<td><?php echo $order['food']; ?></td>

<td>$<?php echo $order['price']; ?></td>

<td><?php echo $order['qty']; ?></td>

<td>$<?php echo $order['total']; ?></td>

<td><?php echo $order['order\_date']; ?></td>

<td class="status <?php echo strtolower($order['status']); ?>"><?php echo $order['status']; ?></td>

<td><?php echo $order['customer\_name']; ?></td>

<td><?php echo $order['customer\_contact']; ?></td>

<td><?php echo $order['customer\_email']; ?></td>

<td><?php echo $order['customer\_address']; ?></td>

<td>

<form action="manage-order.php" method="POST">

<input type="hidden" name="order\_id" value="<?php echo $order['id']; ?>">

<button type="submit" name="action" value="processing" class="btn-action processing">Mark as Processing</button>

<button type="submit" name="action" value="delivered" class="btn-action delivered">Mark as Delivered</button>

<button type="submit" name="action" value="cancelled" class="btn-action cancelled">Mark as Cancelled</button>

<a href="<?php echo SITEURL; ?>admin/update-order.php?id=<?php echo $order['id']; ?>" class="btn-action update">Update Order</a>

</form>

</td>

</tr>

<?php

}

} else {

echo "<tr><td colspan='13' class='error'>No matching orders found.</td></tr>";

}

?>

</table>

</div>

</div>

</div>

<?php include('partials/footer.php'); ?>

<style>

.tbl-full th,

.tbl-full td {

border: 1px solid #ddd;

padding: 10px;

text-align: left;

font-size: 14px;

}

.tbl-full th {

background-color: #f2f2f2;

}

.tbl-full tr:nth-child(even) {

background-color: #f9f9f9;

}

.tbl-full tr:hover {

background-color: #f2f2f2;

}

.btn-action {

display: inline-block;

margin-right: 5px;

padding: 8px 12px;

text-decoration: none;

color: #fff;

border-radius: 4px;

font-size: 12px;

transition: background-color 0.3s;

}

.btn-action.processing {

background-color: #f39c12; /\* Orange \*/

}

.btn-action.delivered {

background-color: #2ecc71; /\* Green \*/

}

.btn-action.cancelled {

background-color: #e74c3c; /\* Red \*/

}

.btn-action.update {

background-color: #3498db; /\* Blue \*/

}

.btn-action:hover {

opacity: 0.8;

}

.status {

font-weight: bold;

display: inline-block;

padding: 6px 10px;

border-radius: 4px;

text-transform: capitalize;

}

.status.ordered {

color: #fff;

background-color: #3498db; /\* Blue \*/

}

.status.processing {

color: #fff;

background-color: #f39c12; /\* Orange \*/

}

.status.cancelled {

color: #fff;

background-color: #e74c3c; /\* Red \*/

}

.status.delivered {

color: #fff;

background-color: #2ecc71; /\* Green \*/

}

.error {

color: red;

font-style: italic;

}

.success {

color: green;

font-style: italic;

}

.input-search {

padding: 8px;

margin-right: 6px;

border: 1px solid #ccc;

border-radius: 4px;

width: 300px;

box-sizing: border-box;

font-size: 14px;

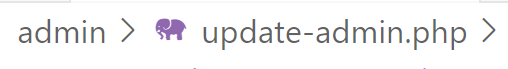
}

.actions-column {

width: 200px;

}

</style>



<?php

// Check if session is not already started

if (session\_status() == PHP\_SESSION\_NONE) {

// If not, start session

session\_start();

}

// Include the menu

include('partials/menu.php');

?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Update Admin</h1>

<br><br>

<?php

//1. get the id of the selected admin

$id = $\_GET['id'];

//2. create SQL query to get the details

$sql = "SELECT \* FROM tbl\_admin WHERE id=$id";

//3. execute the query

$res = mysqli\_query($conn, $sql);

// Check whether the query executed or not

if ($res == true) {

// Check whether the data is available or not

$count = mysqli\_num\_rows($res);

// Check whether we have admin data or not

if ($count == 1) {

// Get the details

$row = mysqli\_fetch\_assoc($res);

$full\_name = $row['full\_name'];

$username = $row['username'];

} else {

// Redirect to manage admin page

header('location: ' . SITEURL . 'admin/manage-admin.php');

}

} else {

// Redirect to manage admin page

header('location: ' . SITEURL . 'admin/manage-admin.php');

}

?>

<form action="" method="POST">

<table class="tbl-30">

<tr>

<td>Full Name:</td>

<td>

<input type="text" name="full\_name" value="<?php echo $full\_name; ?>">

</td>

</tr>

<tr>

<td>Username:</td>

<td>

<input type="text" name="username" value="<?php echo $username; ?>">

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Update Admin" class="btn-secondary">

</td>

</tr>

</table>

</form>

</div>

</div>

<!-- Main Content ends here -->

<?php

// check whether the submit button clicked or not-->

if(isset($\_POST['submit']))

{

//echo "Button clicked";

//Get all the value from form to update

echo $id=$\_POST['id'];

echo $full\_name=$\_POST['full\_name'];

echo $username=$\_POST['username'];

//create a sql queary to update admin

$sql="UPDATE tbl\_admin SET

full\_name='$full\_name',

username='$username'

WHERE id='$id'

";

//execute the query

$res=mysqli\_query($conn, $sql);

//check whether the query executed succuessfully or not

if($res==true)

{

//query executed and admin updated

$\_SESSION['update']="<div class='success'>Admin updated successfully.</div>";

//redirect to manage admin updated

header('location: '.SITEURL.'admin/manage-admin.php');

}

else

{

//failed to update admin

$\_SESSION['update']="<div class='error'>Failed to delete Admin.</div>";

//redirect to manage admin updated

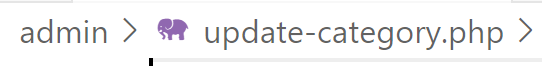
header('location: '.SITEURL.'admin/manage-admin.php');

}

}

?>

<?php include('partials/footer.php'); ?>



<?php include('partials/menu.php'); ?>

<div class="main-content">

<div class="wrapper">

<h1>Update Category</h1>

<br><br>

<?php

// Display session messages if set

if (isset($\_SESSION['update'])) {

echo $\_SESSION['update'];

unset($\_SESSION['update']);

}

if (isset($\_SESSION['upload'])) {

echo $\_SESSION['upload'];

unset($\_SESSION['upload']);

}

// Check whether the ID is set or not

if (isset($\_GET['id'])) {

// Get the ID and all other details

$id = $\_GET['id'];

// Create SQL query to get all other details

$sql = "SELECT \* FROM tbl\_category WHERE id=$id";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Count the rows to check whether the ID is valid or not

$count = mysqli\_num\_rows($res);

if ($count == 1) {

// Get all the data

$row = mysqli\_fetch\_assoc($res);

$title = $row['title'];

$current\_image = $row['image\_name'];

$featured = $row['featured'];

$active = $row['active'];

} else {

// Redirect to manage category with session message

$\_SESSION["no-category-found"] = "<div class='error'>Category not found.</div>";

header('location:' . SITEURL . 'admin/manage-category.php');

die();

}

} else {

// Redirect to manage category

header('location:' . SITEURL . 'admin/manage-category.php');

die();

}

?>

<form action="" method="post" enctype="multipart/form-data">

<table class="tbl-30">

<tr>

<td>Title:</td>

<td>

<input type="text" name="title" value="<?php echo $title; ?>">

</td>

</tr>

<tr>

<td>Current Image:</td>

<td>

<?php

if ($current\_image != "") {

// Display the image

?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $current\_image; ?>" width="150px">

<?php

} else {

echo "<div class='error'>Image not added.</div>";

}

?>

</td>

</tr>

<tr>

<td>New Image:</td>

<td>

<input type="file" name="image">

</td>

</tr>

<tr>

<td>Featured:</td>

<td>

<input type="radio" name="featured" value="Yes" <?php if ($featured == "Yes") { echo "checked"; } ?>> Yes

<input type="radio" name="featured" value="No" <?php if ($featured == "No") { echo "checked"; } ?>> No

</td>

</tr>

<tr>

<td>Active:</td>

<td>

<input type="radio" name="active" value="Yes" <?php if ($active == "Yes") { echo "checked"; } ?>> Yes

<input type="radio" name="active" value="No" <?php if ($active == "No") { echo "checked"; } ?>> No

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="current\_image" value="<?php echo $current\_image; ?>">

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Update Category" class="btn-secondary">

</td>

</tr>

</table>

</form>

<?php

if (isset($\_POST['submit'])) {

// Get all the values from the form

$id = $\_POST['id'];

$title = $\_POST['title'];

$current\_image = $\_POST['current\_image'];

$featured = $\_POST['featured'];

$active = $\_POST['active'];

// Check whether the image is selected or not

if (isset($\_FILES['image']['name']) && $\_FILES['image']['name'] != "") {

// Image is selected

$image\_name = $\_FILES['image']['name'];

// Auto rename the image to avoid duplicate image names

$ext = pathinfo($image\_name, PATHINFO\_EXTENSION);

$image\_name = "Food\_Category\_" . rand(000, 999) . '.' . $ext;

$source\_path = $\_FILES['image']['tmp\_name'];

$destination\_path = "../images/category/" . $image\_name;

// Finally, upload the image

$upload = move\_uploaded\_file($source\_path, $destination\_path);

// Check whether the image is uploaded or not

if ($upload == false) {

// Set message

$\_SESSION['upload'] = "<div class='error'>Failed to upload the image.</div>";

// Redirect to add category page

header('location:' . SITEURL . 'admin/manage-category.php');

// Stop the process

die();

}

// Remove current image if available

if ($current\_image != "") {

$remove\_path = "../images/category/" . $current\_image;

$remove = unlink($remove\_path);

// Check whether the image is removed or not

if ($remove == false) {

// Failed to remove image

$\_SESSION['failed-remove'] = "<div class='error'>Failed to remove the current image.</div>";

// Redirect to manage category page

header('location:' . SITEURL . 'admin/manage-category.php');

// Stop the process

die();

}

}

} else {

$image\_name = $current\_image;

}

// Create SQL query to update category

$sql2 = "UPDATE tbl\_category SET

title='$title',

image\_name='$image\_name',

featured='$featured',

active='$active'

WHERE id=$id";

// Execute the query

$res2 = mysqli\_query($conn, $sql2);

// Check whether the query executed or not

if ($res2 == true) {

// Category updated

$\_SESSION['update'] = "<div class='success'>Category updated successfully.</div>";

// Redirect to manage category page

header('location:' . SITEURL . 'admin/manage-category.php');

} else {

// Failed to update category

$\_SESSION['update'] = "<div class='error'>Failed to update category.</div>";

// Redirect to update category page

header('location:' . SITEURL . 'admin/manage-category.php');

}

}

?>

</div>

</div>

<?php include('partials/footer.php'); ?>



<?php

ob\_start(); // Start output buffering

session\_start();

include('partials/menu.php');

// Check whether the ID is set or not

if (isset($\_GET['id'])) {

$id = $\_GET['id'];

// Get the details of the selected food

$sql = "SELECT \* FROM tbl\_food WHERE id=$id";

$res = mysqli\_query($conn, $sql);

$row = mysqli\_fetch\_assoc($res);

$title = $row['title'];

$description = $row['description'];

$price = $row['price'];

$current\_image = $row['image\_name'];

$current\_category = $row['category\_id'];

$featured = $row['featured'];

$active = $row['active'];

} else {

// Redirect to Manage Food page

header('location:' . SITEURL . 'admin/manage-food.php');

}

?>

<div class="main-content">

<div class="wrapper">

<h1>Update Food</h1>

<br><br>

<?php

if (isset($\_SESSION['upload'])) {

echo $\_SESSION['upload'];

unset($\_SESSION['upload']);

}

?>

<form action="" method="post" enctype="multipart/form-data">

<table class="tbl-30">

<tr>

<td>Title:</td>

<td>

<input type="text" name="title" value="<?php echo $title; ?>" required>

</td>

</tr>

<tr>

<td>Description:</td>

<td>

<textarea name="description" cols="30" rows="5" required><?php echo $description; ?></textarea>

</td>

</tr>

<tr>

<td>Price:</td>

<td>

<input type="number" name="price" value="<?php echo $price; ?>" step="0.01" required>

</td>

</tr>

<tr>

<td>Current Image:</td>

<td>

<?php

if ($current\_image != "") {

echo "<img src='" . SITEURL . "images/food/" . $current\_image . "' width='150px'>";

} else {

echo "<div class='error'>Image not added.</div>";

}

?>

</td>

</tr>

<tr>

<td>New Image:</td>

<td>

<input type="file" name="image">

</td>

</tr>

<tr>

<td>Category:</td>

<td>

<select name="category" required>

<?php

$sql2 = "SELECT \* FROM tbl\_category WHERE active='Yes'";

$res2 = mysqli\_query($conn, $sql2);

if (mysqli\_num\_rows($res2) > 0) {

while ($row2 = mysqli\_fetch\_assoc($res2)) {

$category\_id = $row2['id'];

$category\_title = $row2['title'];

?>

<option <?php if ($current\_category == $category\_id) echo "selected"; ?> value="<?php echo $category\_id; ?>"><?php echo $category\_title; ?></option>

<?php

}

} else {

echo "<option value='0'>No Category Found</option>";

}

?>

</select>

</td>

</tr>

<tr>

<td>Featured:</td>

<td>

<input <?php if ($featured == "Yes") echo "checked"; ?> type="radio" name="featured" value="Yes"> Yes

<input <?php if ($featured == "No") echo "checked"; ?> type="radio" name="featured" value="No"> No

</td>

</tr>

<tr>

<td>Active:</td>

<td>

<input <?php if ($active == "Yes") echo "checked"; ?> type="radio" name="active" value="Yes"> Yes

<input <?php if ($active == "No") echo "checked"; ?> type="radio" name="active" value="No"> No

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="current\_image" value="<?php echo $current\_image; ?>">

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Update Food" class="btn-secondary">

</td>

</tr>

</table>

</form>

<?php

if (isset($\_POST['submit'])) {

// Get all the details from the form

$id = $\_POST['id'];

$title = $\_POST['title'];

$description = $\_POST['description'];

$price = floatval($\_POST['price']);

$current\_image = $\_POST['current\_image'];

$category = $\_POST['category'];

$featured = $\_POST['featured'];

$active = $\_POST['active'];

// Check if the image is selected

if (isset($\_FILES['image']['name']) && $\_FILES['image']['name'] != "") {

$file\_name = $\_FILES['image']['name'];

$file\_tmp = $\_FILES['image']['tmp\_name'];

$file\_size = $\_FILES['image']['size'];

// Check file size (max 5MB)

$max\_file\_size = 5 \* 1024 \* 1024; // 5 MB in bytes

if ($file\_size > $max\_file\_size) {

$\_SESSION['upload'] = "<div class='error'>File size exceeds maximum limit (5MB).</div>";

header('location:' . SITEURL . 'admin/update-food.php?id=' . $id);

exit();

}

// Ensure uploaded file is an image

$allowed\_types = array('jpg', 'jpeg', 'png', 'gif');

$file\_ext = strtolower(pathinfo($file\_name, PATHINFO\_EXTENSION));

if (!in\_array($file\_ext, $allowed\_types)) {

$\_SESSION['upload'] = "<div class='error'>Only JPG, JPEG, PNG, GIF files are allowed.</div>";

header('location:' . SITEURL . 'admin/update-food.php?id=' . $id);

exit();

}

// Generate a unique file name to prevent conflicts

$new\_file\_name = "Food\_" . uniqid() . "." . $file\_ext;

// Destination path for the uploaded file

$upload\_path = "../images/food/" . $new\_file\_name;

// Check if the directory exists and has the correct permissions

if (!is\_dir("../images/food/") || !is\_writable("../images/food/")) {

$\_SESSION['upload'] = "<div class='error'>Upload directory is not writable. Please check permissions.</div>";

header('location:' . SITEURL . 'admin/update-food.php?id=' . $id);

exit();

}

// Move uploaded file to destination directory

if (!move\_uploaded\_file($file\_tmp, $upload\_path)) {

$\_SESSION['upload'] = "<div class='error'>Failed to upload image. Please try again.</div>";

header('location:' . SITEURL . 'admin/update-food.php?id=' . $id);

exit();

}

// Remove the current image if available

if ($current\_image != "") {

$remove\_path = "../images/food/" . $current\_image;

if (!unlink($remove\_path)) {

$\_SESSION['upload'] = "<div class='error'>Failed to remove current image.</div>";

header('location:' . SITEURL . 'admin/update-food.php?id=' . $id);

exit();

}

}

// Update the new image name

$image\_name = $new\_file\_name;

} else {

$image\_name = $current\_image;

}

// Update the database

$sql3 = "UPDATE tbl\_food SET

title = '$title',

description = '$description',

price = $price,

image\_name = '$image\_name',

category\_id = $category,

featured = '$featured',

active = '$active'

WHERE id=$id";

$res3 = mysqli\_query($conn, $sql3);

if ($res3) {

$\_SESSION['update'] = "<div class='success'>Food updated successfully.</div>";

header('location:' . SITEURL . 'admin/manage-food.php');

} else {

$\_SESSION['update'] = "<div class='error'>Failed to update food.</div>";

header('location:' . SITEURL . 'admin/update-food.php?id=' . $id);

}

}

?>

</div>

</div>

<?php include('partials/footer.php'); ?>

<?php

ob\_end\_flush(); // Flush the output from the buffer

?>



<?php

include('partials/menu.php');

// Start session if not already started

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

// Check if ID is provided in URL

if (isset($\_GET['id'])) {

$id = $\_GET['id'];

// Fetch order details from database based on $id

$sql = "SELECT \* FROM tbl\_order WHERE id = $id";

$res = mysqli\_query($conn, $sql);

if ($res && mysqli\_num\_rows($res) == 1) {

// Fetch order details

$order = mysqli\_fetch\_assoc($res);

$food = htmlspecialchars($order['food']);

$qty = $order['qty'];

$status = $order['status'];

$customer\_name = htmlspecialchars($order['customer\_name']);

$customer\_contact = htmlspecialchars($order['customer\_contact']);

$customer\_email = htmlspecialchars($order['customer\_email']);

$customer\_address = htmlspecialchars($order['customer\_address']);

} else {

// Order not found, set session message and redirect

$\_SESSION['error'] = "Order not found.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

} else {

// Redirect to manage-order.php if id is not set

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

}

// Handle form submission

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

if (isset($\_POST['submit'])) {

$id = $\_POST['id'];

$qty = $\_POST['qty'];

$status = $\_POST['status'];

$customer\_name = $\_POST['customer\_name'];

$customer\_contact = $\_POST['customer\_contact'];

$customer\_email = $\_POST['customer\_email'];

$customer\_address = $\_POST['customer\_address'];

// Update order details in database

$sql2 = "UPDATE tbl\_order SET

qty = '$qty',

status = '$status',

customer\_name = '$customer\_name',

customer\_contact = '$customer\_contact',

customer\_email = '$customer\_email',

customer\_address = '$customer\_address'

WHERE id = $id";

$res2 = mysqli\_query($conn, $sql2);

if ($res2) {

$\_SESSION['update'] = "Order updated successfully.";

header('location: ' . SITEURL . 'admin/manage-order.php');

exit;

} else {

$\_SESSION['error'] = "Failed to update order.";

}

}

}

?>

<!-- HTML form for updating order -->

<div class="main-content">

<div class="wrapper">

<h1>Update Order</h1>

<br><br>

<!-- Form with fields for updating order details -->

<form action="" method="POST" class="order">

<fieldset>

<legend>Order Details</legend>

<label for="food">Food:</label>

<input type="text" name="food" id="food" class="input-responsive" value="<?php echo $food; ?>" readonly>

<label for="qty">Quantity:</label>

<input type="number" name="qty" id="qty" class="input-responsive" value="<?php echo $qty; ?>" min="1" required>

<label for="status">Status:</label>

<select name="status" id="status" class="input-responsive" required>

<option value="Ordered" <?php echo ($status == 'Ordered') ? 'selected' : ''; ?> class="status-ordered">Ordered</option>

<option value="Processing" <?php echo ($status == 'Processing') ? 'selected' : ''; ?> class="status-processing">Processing</option>

<option value="Cancelled" <?php echo ($status == 'Cancelled') ? 'selected' : ''; ?> class="status-cancelled">Cancelled</option>

<option value="Delivered" <?php echo ($status == 'Delivered') ? 'selected' : ''; ?> class="status-delivered">Delivered</option>

</select>

<label for="customer\_name">Customer Name:</label>

<input type="text" name="customer\_name" id="customer\_name" class="input-responsive" value="<?php echo $customer\_name; ?>" required>

<label for="customer\_contact">Phone Number:</label>

<input type="tel" name="customer\_contact" id="customer\_contact" class="input-responsive" value="<?php echo $customer\_contact; ?>" required>

<label for="customer\_email">Email:</label>

<input type="email" name="customer\_email" id="customer\_email" class="input-responsive" value="<?php echo $customer\_email; ?>" required>

<label for="customer\_address">Address:</label>

<textarea name="customer\_address" id="customer\_address" rows="5" class="input-responsive" required><?php echo $customer\_address; ?></textarea>

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Update Order" class="btn btn-primary">

</fieldset>

</form>

</div>

</div>

<?php include('partials/footer.php'); ?>

<style>

/\* General Reset and Typography \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

/\* Container and Layout \*/

.container {

width: 90%;

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

/\* Main Content Styling \*/

.main-content {

background-color: #f2f2f2;

padding: 20px;

border-radius: 10px;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

}

.wrapper {

width: 100%;

max-width: 800px;

margin: 0 auto;

}

/\* Form Styling \*/

.order fieldset {

border: 2px solid #ddd;

padding: 20px;

border-radius: 10px;

margin-bottom: 20px;

}

.order legend {

font-size: 1.5rem;

font-weight: bold;

color: #333;

margin-bottom: 10px;

}

.input-group {

margin-bottom: 15px;

}

.input-group label {

display: block;

margin-bottom: 5px;

font-weight: bold;

color: #555;

}

.input-responsive {

width: calc(100% - 20px);

padding: 12px;

font-size: 1rem;

border: 2px solid #ccc;

border-radius: 5px;

transition: border-color 0.3s ease;

}

.input-responsive:focus {

outline: none;

border-color: #ff6b81;

box-shadow: 0 0 8px rgba(255, 107, 129, 0.5);

}

/\* Button Styling \*/

.btn {

padding: 12px 24px;

font-size: 1rem;

background-color: #ff6b81;

color: white;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s ease;

}

.btn:hover {

background-color: #ff4757;

}

/\* Responsive Design \*/

@media only screen and (max-width: 768px) {

.container {

padding: 10px;

}

.wrapper {

width: 100%;

}

.input-responsive {

width: calc(100% - 16px);

padding: 10px;

}

}

</style>



<?php

// Check if session is not already started

if (session\_status() == PHP\_SESSION\_NONE) {

// If not, start session

session\_start();

}

// Include the menu

include('partials/menu.php');

// Check if the ID is set in the GET request

if (isset($\_GET['id'])) {

$id = $\_GET['id'];

} else {

// Redirect to manage admin page if ID is not set

header('location:'.SITEURL.'admin/manage-admin.php');

}

?>

<!-- Main Content starts here -->

<div class="main-content">

<div class="wrapper">

<h1>Change Password</h1>

<br><br>

<form action="" method="POST">

<table class="tbl-30">

<tr>

<td>Current Password: </td>

<td>

<input type="password" name="current\_password" placeholder="Current Password">

</td>

</tr>

<tr>

<td>New Password:</td>

<td>

<input type="password" name="new\_password" placeholder="New Password">

</td>

</tr>

<tr>

<td>Confirm Password:</td>

<td>

<input type="password" name="confirm\_password" placeholder="Confirm Password">

</td>

</tr>

<tr>

<td colspan="2">

<input type="hidden" name="id" value="<?php echo $id; ?>">

<input type="submit" name="submit" value="Change Password" class="btn-secondary">

</td>

</tr>

</table>

</form>

</div>

</div>

<!-- Main Content ends here -->

<?php include('partials/footer.php'); ?>

<?php

// Check whether the submit button is clicked or not

if (isset($\_POST['submit'])) {

// 1. Get the data from form

$id = $\_POST['id'];

$current\_password = md5($\_POST['current\_password']);

$new\_password = md5($\_POST['new\_password']);

$confirm\_password = md5($\_POST['confirm\_password']);

// 2. Check whether the user with current ID and current password exists or not

$sql = "SELECT \* FROM tbl\_admin WHERE id=$id AND password='$current\_password'";

// Execute the query

$res = mysqli\_query($conn, $sql);

if ($res == true) {

// Check whether data is available or not

$count = mysqli\_num\_rows($res);

if ($count == 1) {

// User exists and password can be changed

// Check whether the new password and confirm password match or not

if ($new\_password == $confirm\_password) {

// Update the password

$sql2 = "UPDATE tbl\_admin SET password='$new\_password' WHERE id=$id";

// Execute the query

$res2 = mysqli\_query($conn, $sql2);

// Check whether the query executed or not

if ($res2 == true) {

// Password changed successfully

$\_SESSION['change-pwd'] = "<div class='success'>Password Changed Successfully.</div>";

// Redirect the user

header('location:'.SITEURL.'admin/manage-admin.php');

} else {

// Failed to change password

$\_SESSION['change-pwd'] = "<div class='error'>Failed to Change Password.</div>";

// Redirect the user

header('location:'.SITEURL.'admin/manage-admin.php');

}

} else {

// Password did not match

$\_SESSION['pwd-not-match'] = "<div class='error'>Password Did Not Match.</div>";

// Redirect the user

header('location:'.SITEURL.'admin/manage-admin.php');

}

} else {

// User does not exist, set message and redirect

$\_SESSION['user-not-found'] = "<div class='error'>The password you entered is incorrect.</div>";

// Redirect the user

header('location:'.SITEURL.'admin/manage-admin.php');

}

}

}

?>



/\*

Author: Mohammed Shohel

E-mail: codewithshohel@gmail.com

Theme: Food order website - Admin Panel

Version: 1.0

\*/

/\* Body styles \*/

body {

font-family: Arial, sans-serif;

background-color: #f1f2f6;

}

/\* Reset CSS \*/

\* {

margin: 0;

padding: 0;

font-family: Arial, Helvetica, sans-serif;

box-sizing: border-box; /\* Ensure padding and borders are included in total width and height \*/

}

/\* Wrapper for content \*/

.wrapper {

padding: 20px; /\* Padding around content \*/

width: 80%; /\* Width of the wrapper \*/

margin: 0 auto; /\* Center aligning the wrapper \*/

}

/\* Center align text \*/

.text-center {

text-align: center;

}

/\* Clearfix for floats \*/

.clearfix {

float: none;

clear: both;

}

/\* CSS for menu \*/

.menu {

border-bottom: 2px solid transparent; /\* Bottom border for menu \*/

padding: 3px 0; /\* Padding for menu \*/

background-color: #ffffff; /\* Background color \*/

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /\* Shadow effect for menu \*/

position: sticky; /\* Stick to the top of the page \*/

top: 0; /\* Stick to the top of the viewport \*/

z-index: 1000; /\* Ensure it stays above other elements \*/

transition: all 0.3s ease-in-out; /\* Smooth transition for background \*/

}

.menu ul {

list-style-type: none; /\* Remove default list style \*/

display: flex; /\* Use flexbox for layout \*/

justify-content: center; /\* Center align menu items \*/

margin: 0;

padding: 0;

}

.menu ul li {

margin: 0 15px; /\* Margin around menu items \*/

}

.menu ul li a {

text-decoration: none; /\* Remove underline \*/

font-weight: bold; /\* Bold text \*/

color: #333; /\* Menu item color \*/

transition: color 0.3s, background-color 0.3s, transform 0.3s; /\* Transition for text color, background color, and transform \*/

padding: 10px 20px; /\* Padding for menu items \*/

border-radius: 5px; /\* Rounded corners \*/

position: relative; /\* Position for pseudo-elements \*/

display: inline-block; /\* Set display to inline-block \*/

}

.menu ul li a::before {

content: ''; /\* Empty content for decoration \*/

position: absolute; /\* Absolute positioning \*/

width: 0; /\* Initially no width \*/

height: 3px; /\* Height of underline \*/

bottom: -5px; /\* Position at bottom with slight overlap \*/

left: 50%; /\* Start from center \*/

transform: translateX(-50%); /\* Center align \*/

background-color: transparent; /\* Underline color \*/

transition: width 0.3s ease-in-out, background-color 0.3s ease-in-out; /\* Smooth transition for width and background color \*/

}

.menu ul li a:hover {

color: #fa0000; /\* Hover color \*/

background-color: #f5f5f5; /\* Background color on hover \*/

transform: translateY(-3px); /\* Slightly move up on hover \*/

}

.menu ul li a:hover::before {

width: calc(100% - 10px); /\* Full width on hover with slight padding \*/

background-color: #fa0000; /\* Underline color on hover \*/

}

/\* CSS for main-content \*/

.main-content {

background-color: #f1f2f6; /\* Background color \*/

padding: 3% 2%; /\* Padding for content \*/

transition: padding 0.3s ease-in-out; /\* Smooth transition for padding \*/

}

.col-4 {

width: 18%; /\* Width of columns \*/

background-color: white; /\* Background color \*/

margin: 1%; /\* Margin between columns \*/

padding: 2% 0; /\* Padding inside columns \*/

float: left; /\* Floating columns \*/

border-radius: 10px; /\* Rounded corners \*/

box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1); /\* Box shadow \*/

transition: transform 0.3s ease, box-shadow 0.3s ease; /\* Smooth transition for transform and box shadow \*/

}

.col-4:hover {

transform: translateY(-5px); /\* Move up slightly on hover \*/

box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15); /\* Enhanced shadow on hover \*/

}

.tbl-full, .tbl-30 {

width: 100%; /\* Full width table \*/

border-collapse: collapse; /\* Collapse borders \*/

margin-bottom: 20px; /\* Space below table \*/

}

.tbl-30 {

width: 30%; /\* 30% width table \*/

}

table tr th, table tr td {

border-bottom: 1px solid #ddd; /\* Bottom border for table header \*/

padding: 1%; /\* Padding for table header \*/

text-align: left; /\* Text alignment \*/

transition: background-color 0.3s ease-in-out; /\* Smooth transition for background color \*/

}

table tr th {

background-color: #f9f9f9; /\* Light background for header \*/

}

table tr:hover td {

background-color: #f5f5f5; /\* Light background on row hover \*/

}

/\* Buttons \*/

.btn-primary, .btn-secondary, .btn-danger {

display: inline-block; /\* Inline block for padding and margins \*/

padding: 10px 20px; /\* Padding for buttons \*/

color: white; /\* Text color \*/

text-decoration: none; /\* Remove underline \*/

font-weight: bold; /\* Bold text \*/

border-radius: 5px; /\* Rounded corners \*/

transition: background-color 0.3s, color 0.3s, transform 0.3s; /\* Smooth transition for background color, text color, and transform \*/

}

.btn-primary {

background-color: #0343f3; /\* Button background color \*/

}

.btn-primary:hover {

background-color: #11b4ff; /\* Hover background color \*/

color: black; /\* Hover text color \*/

transform: translateY(-2px); /\* Slightly move up on hover \*/

text-decoration: underline; /\* Underline on hover \*/

}

.btn-secondary {

background-color: #389324; /\* Button background color \*/

}

.btn-secondary:hover {

background-color: #ccf358; /\* Hover background color \*/

color: black; /\* Hover text color \*/

transform: translateY(-2px); /\* Slightly move up on hover \*/

text-decoration: underline; /\* Underline on hover \*/

}

.btn-danger {

background-color: #ff2424; /\* Button background color \*/

}

.btn-danger:hover {

background-color: #ff0000; /\* Hover background color \*/

color: black; /\* Hover text color \*/

transform: translateY(-2px); /\* Slightly move up on hover \*/

text-decoration: underline; /\* Underline on hover \*/

}

/\*CSS for Message\*/

/\* Define variables for messages \*/

:root {

--success-text-color: #00ab66; /\* Green text color for success \*/

--success-bg-color: #f0fff4; /\* Light green background color for success \*/

--success-border-color: #8beac2; /\* Border color for success \*/

--error-text-color: #ff3d71; /\* Red text color for error \*/

--error-bg-color: #ffe6e6; /\* Light red background color for error \*/

--error-border-color: #ffb3b3; /\* Border color for error \*/

--info-text-color: #31708f; /\* Text color for info message \*/

--info-bg-color: #d9edf7; /\* Background color for info message \*/

--info-border-color: #bce8f1; /\* Border color for info message \*/

--font-family: 'Roboto', Arial, sans-serif; /\* Modern font stack \*/

--font-size: 16px; /\* Base font size \*/

--border-radius: 10px; /\* Increased border radius for rounded corners \*/

--box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /\* Soft shadow effect \*/

--padding: 10px; /\* Reduced padding for messages \*/

--padding-error: 15px 20px; /\* Adjusted padding for error messages \*/

--margin-bottom: 1px; /\* Spacing between messages \*/

--container-padding: 5px; /\* Reduced padding for message container \*/

}

/\* Success message \*/

.success {

color: var(--success-text-color);

background-color: var(--success-bg-color);

border: 1px solid var(--success-border-color);

padding: var(--padding);

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

margin-bottom: var(--margin-bottom);

font-size: var(--font-size);

font-family: var(--font-family);

transition: transform 0.3s ease, box-shadow 0.3s ease;

display: inline-block; /\* Fit background to text width \*/

}

/\* Error message \*/

.error {

color: var(--error-text-color);

background-color: var(--error-bg-color);

border: 1px solid var(--error-border-color);

padding: var(--padding-error); /\* Use the adjusted padding \*/

border-radius: var(--border-radius);

box-shadow: var(--box-shadow);

margin-bottom: var(--margin-bottom);

font-size: var(--font-size);

font-family: var(--font-family);

transition: transform 0.3s ease, box-shadow 0.3s ease;

display: inline-block; /\* Fit background to text width \*/

}

/\* Message container \*/

.message-container {

max-width: 600px; /\* Maximum width of the message container \*/

width: 90%; /\* Set width to 90% of its parent \*/

margin: 20px auto; /\* Center align and spacing from top \*/

padding: var(--container-padding); /\* Adjusted padding for message container \*/

box-sizing: border-box; /\* Include padding and border in the element's total width and height \*/

}

/\* Add hover effect for interaction \*/

.success:hover, .error:hover {

transform: translateY(-2px);

box-shadow: 0 6px 8px rgba(0, 0, 0, 0.1);

}

/\* Info message \*/

.message-info {

color: var(--info-text-color);

background-color: var(--info-bg-color);

border: 1px solid var(--info-border-color);

padding: var(--padding); /\* Use the defined padding \*/

border-radius: var(--border-radius); /\* Use the defined border radius \*/

box-shadow: var(--box-shadow); /\* Use the defined box shadow \*/

margin-bottom: var(--margin-bottom); /\* Use the defined margin bottom \*/

font-size: var(--font-size); /\* Use the defined font size \*/

font-family: var(--font-family); /\* Use the defined font family \*/

transition: transform 0.3s ease, box-shadow 0.3s ease; /\* Use the defined transition \*/

}

/\* Add hover effect for interaction \*/

.message-info:hover {

transform: translateY(-2px);

box-shadow: 0 6px 8px rgba(0, 0, 0, 0.1); /\* Hover shadow \*/

}

/\* Add animations \*/

@keyframes fadeIn {

from { opacity: 0; }

to { opacity: 1; }

}

@keyframes slideIn {

from { transform: translateY(20px); opacity: 0; }

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

}

.main-content, .message-container {

animation: fadeIn 0.5s ease-in-out;

}

.col-4 {

animation: slideIn 0.5s ease-in-out;

}

/\*login\*/

/\* Login container \*/

.login {

max-width: 400px;

margin: 50px auto;

background-color: #fff;

padding: 30px;

border-radius: 10px;

box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

}

/\* Login header \*/

.login h1 {

font-size: 24px;

text-align: center;

margin-bottom: 20px;

}

/\* Form styles \*/

.login form {

text-align: center;

}

/\* Form inputs \*/

.login input[type="text"],

.login input[type="password"] {

width: 100%;

padding: 10px;

margin-bottom: 20px;

border: 1px solid #ddd;

border-radius: 5px;

outline: none;

}

/\* Submit button \*/

.login input[type="submit"] {

width: 100%;

padding: 10px;

background-color: #fa0000;

color: #fff;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s ease;

}

/\* Submit button hover effect \*/

.login input[type="submit"]:hover {

background-color: #dd0000;

}

/\* Footer \*/

.login p {

font-size: 14px;

text-align: center;

}

/\* Footer link \*/

.login a {

color: #fa0000;

text-decoration: none;

}

/\* Footer link hover effect \*/

.login a:hover {

text-decoration: underline;

}

/\* CSS for footer \*/

.footer {

background-color: rgb(226, 55, 55); /\* Footer background color \*/

color: white; /\* Text color \*/

padding: 20px; /\* Padding for footer \*/

text-align: center; /\* Center align text \*/

}



/\*

Author: Vijay Thapa;

Theme: Restaurant Food Order;

version: 1.0;

\*/

/\* CSS for All \*/

\* {

margin: 0;

padding: 0;

box-sizing: border-box; /\* Ensures padding and border are included in element's total width and height \*/

font-family: Arial, Helvetica, sans-serif;

}

.container {

width: 80%;

margin: 0 auto;

padding: 1%;

}

.img-responsive {

width: 100%;

}

.img-curve {

border-radius: 15px;

transition: transform 0.3s ease; /\* Smooth transition for zoom effect \*/

}

.img-curve:hover {

transform: scale(1.05); /\* Zoom effect on hover \*/

}

.text-right {

text-align: right;

}

.text-center {

text-align: center;

}

.text-left {

text-align: left;

}

.text-white {

color: white;

}

.clearfix {

clear: both;

float: none;

}

a {

color: #ff6b81;

text-decoration: none;

}

a:hover {

color: #ff4757;

}

.btn {

padding: 1%;

border: none;

font-size: 1rem;

border-radius: 5px;

}

.btn-primary {

background-color: #ff6b81;

color: white;

cursor: pointer;

}

.btn-primary:hover {

background-color: #ff4757;

}

h2 {

color: #2f3542;

font-size: 2rem;

margin-bottom: 2%;

}

h3 {

font-size: 1.5rem;

}

.float-container {

position: relative;

}

.float-text {

position: absolute;

bottom: 50px;

left: 40%;

}

fieldset {

border: 1px solid white;

margin: 5%;

padding: 3%;

border-radius: 5px;

}

/\* CSSS for navbar section \*/

.logo {

width: 10%;

float: left;

}

.menu {

line-height: 60px;

}

.menu ul {

list-style-type: none;

}

.menu ul li {

display: inline;

padding: 1%;

font-weight: bold;

align-items: right;

}

/\* CSS for Food SEarch Section \*/

.food-search {

background-image: url(../images/bg.jpg);

background-size: cover;

background-repeat: no-repeat;

background-position: center;

padding: 7% 0;

}

.food-search input[type="search"] {

width: 50%;

padding: 1%;

font-size: 1rem;

border: none;

border-radius: 5px;

}

/\* CSS for Categories \*/

.categories {

padding: 4% 0;

}

.box-3 {

width: 28%;

float: left;

margin: 2%;

}

/\* CSS for Food Menu \*/

.food-menu {

background-color: #ececec;

padding: 4% 0;

}

.food-menu-box {

width: 43%;

margin: 1%;

padding: 2%;

float: left;

background-color: white;

border-radius: 15px;

transition: transform 0.3s ease; /\* Smooth transition for zoom effect \*/

}

/\* Zoom effect on double-click \*/

.food-menu-box.zoomed {

transform: scale(1.1); /\* Example: Zoom effect on double-click \*/

}

.food-menu-box:hover {

transform: scale(1.05); /\* Zoom effect on hover \*/

}

.food-menu-img {

width: 20%;

float: left;

}

.food-menu-desc {

width: 70%;

float: left;

margin-left: 8%;

}

.food-price {

font-size: 1.2rem;

margin: 2% 0;

}

.food-detail {

font-size: 1rem;

color: #747d8c;

}

/\* CSS for Social \*/

.social ul {

list-style-type: none;

}

.social ul li {

display: inline;

padding: 1%;

}

/\* for Order Section \*/

.order {

width: 50%;

margin: 0 auto;

}

.input-responsive {

width: 96%;

padding: 1%;

margin-bottom: 3%;

border: none;

border-radius: 5px;

font-size: 1rem;

}

.order-label {

margin-bottom: 1%;

font-weight: bold;

}

/\* CSS for Mobile Size or Smaller Screen \*/

@media only screen and (max-width: 768px) {

.logo {

width: 80%;

float: none;

margin: 1% auto;

}

.menu ul {

text-align: center;

}

.food-search input[type="search"] {

width: 90%;

padding: 2%;

margin-bottom: 3%;

}

.btn {

width: 91%;

padding: 2%;

}

.food-search {

padding: 10% 0;

}

.categories {

padding: 20% 0;

}

h2 {

margin-bottom: 10%;

}

.box-3 {

width: 100%;

margin: 4% auto;

}

.food-menu {

padding: 20% 0;

}

.food-menu-box {

width: 90%;

padding: 5%;

margin-bottom: 5%;

}

.social {

padding: 5% 0;

}

.order {

width: 100%;

}

}



<?php

// Include constants

include('config/constants.php');

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<!-- Important to make website responsive -->

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Restaurant Website</title>

<!-- Link our CSS file -->

<link rel="stylesheet" href="<?php echo SITEURL; ?>css/style.css">

</head>

<body>

<!-- Navbar Section Starts Here -->

<section class="navbar">

<div class="container">

<div class="navbar-content">

<div class="logo">

<a href="<?php echo SITEURL; ?>" title="Logo">

<img src="<?php echo SITEURL; ?>images/logo1.png" alt="Restaurant Logo" class="img-responsive">

</a>

</div>

<div class="menu">

<ul>

<li>

TASTE AWESOME FOOD

</li>

<li>

<a href="<?php echo SITEURL; ?>">Home</a>

</li>

<li>

<a href="<?php echo SITEURL; ?>categories.php">Categories</a>

</li>

<li>

<a href="<?php echo SITEURL; ?>foods.php">Foods</a>

</li>

<li>

<a href="#">Contact</a>

</li>

</ul>

</div>

</div>

</div>

</section>

<!-- Navbar Section Ends Here -->

</body>

</html>



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<!-- Important to make website responsive -->

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Restaurant Website</title>

<!-- Link our CSS file -->

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<!-- social Section Starts Here -->

<section class="social">

<div class="container text-center">

<ul>

<li>

<a href="#"><img src="https://img.icons8.com/fluent/50/000000/facebook-new.png"/></a>

</li>

<li>

<a href="#"><img src="https://img.icons8.com/fluent/48/000000/instagram-new.png"/></a>

</li>

<li>

<a href="#"><img src="https://img.icons8.com/fluent/48/000000/twitter.png"/></a>

</li>

</ul>

</div>

</section>

<!-- social Section Ends Here -->

<!-- footer Section Starts Here -->

<section class="footer">

<div class="container text-center">

<p>All rights reserved. Designed By <a href="#">Mohammed Shohel</a></p>

<p>&copy; <?php echo date('Y'); ?> &ndash; <strong>TASTE AWESOME FOOD</strong></p>

</div>

</section>

<!-- footer Section Ends Here -->

</body>

</html>



<?php include\_once 'partials-front/menu.php'; ?>

<!-- Categories Section Starts Here -->

<section class="categories">

<div class="container">

<h2 class="text-center">Explore Foods</h2>

<?php

// Include database connection and constants

include\_once 'config/constants.php';

// Establish database connection

$conn = mysqli\_connect(LOCALHOST, DB\_USERNAME, DB\_PASSWORD, DB\_NAME);

if (!$conn) {

die("Connection failed: " . mysqli\_connect\_error());

}

// SQL query to fetch active categories

$sql = "SELECT \* FROM tbl\_category WHERE active='Yes'";

$res = mysqli\_query($conn, $sql);

// Check if query executed successfully

if ($res) {

// Check if categories are available

if (mysqli\_num\_rows($res) > 0) {

// Iterate through categories

while ($row = mysqli\_fetch\_assoc($res)) {

$id = $row['id'];

$title = $row['title'];

$image\_name = $row['image\_name'];

?>

<a href="<?php echo SITEURL; ?>category-foods.php?category\_id=<?php echo $id; ?>">

<div class="box-3 float-container">

<?php if (!empty($image\_name)) { ?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $image\_name; ?>" alt="<?php echo $title; ?>" class="img-responsive img-curve">

<?php } else { ?>

<div class='error'>Image not found.</div>

<?php } ?>

<h3 class="float-text text-white"><?php echo $title; ?></h3>

</div>

</a>

<?php

}

} else {

// No categories found

echo "<div class='error'>No categories found.</div>";

}

} else {

// Failed to execute query

echo "<div class='error'>Failed to retrieve categories.</div>";

}

// Close database connection

mysqli\_close($conn);

?>

<div class="clearfix"></div>

</div>

</section>

<!-- Categories Section Ends Here -->

<?php include 'partials-front/footer.php'; ?>



<?php

include\_once('partials-front/menu.php');

// Check if category\_id is set in the URL

if(isset($\_GET['category\_id'])) {

$category\_id = $\_GET['category\_id'];

// Fetch the category title based on category\_id using prepared statement

$sql = "SELECT title FROM tbl\_category WHERE id = ? AND active = 'Yes'";

$stmt = mysqli\_prepare($conn, $sql);

mysqli\_stmt\_bind\_param($stmt, 'i', $category\_id);

mysqli\_stmt\_execute($stmt);

$res = mysqli\_stmt\_get\_result($stmt);

if($res) {

if(mysqli\_num\_rows($res) > 0) {

$row = mysqli\_fetch\_assoc($res);

$category\_title = $row['title'];

// Display the category title in the heading

echo "<section class='food-search text-center'>

<div class='container'>

<h2>Foods on <a href='#' class='text-white'>\"$category\_title\"</a> Category</h2>

</div>

</section>";

// SQL query to fetch foods based on category\_id using prepared statement

$sql\_foods = "SELECT \* FROM tbl\_food WHERE category\_id = ? AND active = 'Yes'";

$stmt\_foods = mysqli\_prepare($conn, $sql\_foods);

mysqli\_stmt\_bind\_param($stmt\_foods, 'i', $category\_id);

mysqli\_stmt\_execute($stmt\_foods);

$res\_foods = mysqli\_stmt\_get\_result($stmt\_foods);

if(mysqli\_num\_rows($res\_foods) > 0) {

// Foods available

echo "<section class='food-menu'>

<div class='container'>

<h2 class='text-center'>Food Menu</h2>";

while($row\_food = mysqli\_fetch\_assoc($res\_foods)) {

$food\_id = $row\_food['id'];

$food\_title = $row\_food['title'];

$food\_price = $row\_food['price'];

$food\_description = $row\_food['description'];

$food\_image\_name = $row\_food['image\_name'];

?>

<div class="food-menu-box">

<div class="food-menu-img">

<img src="<?php echo SITEURL; ?>images/food/<?php echo $food\_image\_name; ?>" alt="<?php echo htmlspecialchars($food\_title); ?>" class="img-responsive img-curve">

</div>

<div class="food-menu-desc">

<h4><?php echo htmlspecialchars($food\_title); ?></h4>

<p class="food-price">$<?php echo $food\_price; ?></p>

<p class="food-detail">

<?php echo htmlspecialchars($food\_description); ?>

</p>

<br>

<a href="<?php echo SITEURL; ?>order.php?food\_id=<?php echo $food\_id; ?>" class="btn btn-primary">Order Now</a>

</div>

</div>

<?php

}

echo "<div class='clearfix'></div>

</div>

</section>";

} else {

// No foods found in the category

echo "<div class='container'>

<p class='error'>No foods found in this category.</p>

</div>";

}

} else {

// Category not found or inactive

echo "<div class='container'>

<p class='error'>Category not found or inactive.</p>

</div>";

}

} else {

// Error executing query for category title

echo "<div class='container'>

<p class='error'>Error fetching category title.</p>

</div>";

}

// Close prepared statements

mysqli\_stmt\_close($stmt);

mysqli\_stmt\_close($stmt\_foods);

} else {

// If category\_id is not set, redirect to home page

header('location: ' . SITEURL);

exit; // Ensure no further execution after redirection

}

include\_once('partials-front/footer.php');

?>



<?php include\_once('partials-front/menu.php'); ?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Food Order Website</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<!-- FOOD SEARCH Section Starts Here -->

<section class="food-search text-center">

<div class="container">

<h2>Foods on Your Search "<a href="#" class="text-white"><?php echo isset($\_POST['search']) ? htmlspecialchars($\_POST['search']) : ''; ?></a>"</h2>

</div>

</section>

<!-- FOOD SEARCH Section Ends Here -->

<!-- FOOD MENU Section Starts Here -->

<section class="food-menu">

<div class="container">

<h2 class="text-center">Food Menu</h2>

<?php

// Check if $\_POST['search'] is set

if (isset($\_POST['search'])) {

// Get the search keyword and sanitize it

$search = mysqli\_real\_escape\_string($conn, $\_POST['search']);

// SQL query to get food based on search keyword

$sql = "SELECT \* FROM tbl\_food WHERE title LIKE '%$search%' OR description LIKE '%$search%'";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check whether the foods are available or not

if ($res) {

$count = mysqli\_num\_rows($res);

if ($count > 0) {

// Foods available

while ($row = mysqli\_fetch\_assoc($res)) {

// Get the values

$id = $row['id'];

$title = $row['title'];

$price = $row['price'];

$description = $row['description'];

$image\_name = $row['image\_name'];

?>

<div class="food-menu-box">

<div class="food-menu-img">

<?php if ($image\_name == "") : ?>

<div class='error'>Image not available.</div>

<?php else : ?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name; ?>" alt="<?php echo htmlspecialchars($title); ?>" class="img-responsive img-curve">

<?php endif; ?>

</div>

<div class="food-menu-desc">

<h4><?php echo htmlspecialchars($title); ?></h4>

<p class="food-price">$<?php echo $price; ?></p>

<p class="food-detail">

<?php echo htmlspecialchars($description); ?>

</p>

<br>

<a href="<?php echo SITEURL; ?>order.php?food\_id=<?php echo $id; ?>" class="btn btn-primary">Order Now</a>

</div>

</div>

<?php

}

} else {

// Food not available

echo "<div class='error'>Food not found.</div>";

}

} else {

// Error executing query

echo "<div class='error'>Error fetching food details.</div>";

}

// Free result set

mysqli\_free\_result($res);

} else {

// $\_POST['search'] is not set

echo "<div class='error'>No search keyword provided.</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!-- FOOD MENU Section Ends Here -->

<?php include('partials-front/footer.php'); ?>

</body>

</html>



<?php include\_once('partials-front/menu.php'); ?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Food Order Website</title>

<link rel="stylesheet" href="css/style.css">

</head>

<body>

<!-- FOOD SEARCH Section Starts Here -->

<section class="food-search text-center">

<div class="container">

<form action="<?php echo SITEURL; ?>food-search.php" method="POST">

<input type="search" name="search" placeholder="Search for Food.." required>

<input type="submit" name="submit" value="Search" class="btn btn-primary">

</form>

</div>

</section>

<!-- FOOD SEARCH Section Ends Here -->

<!-- FOOD MENU Section Starts Here -->

<section class="food-menu">

<div class="container">

<h2 class="text-center">Food Menu</h2>

<?php

// SQL query to select active foods

$sql = "SELECT \* FROM tbl\_food WHERE active='Yes'";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check if foods are available

if ($res) {

$count = mysqli\_num\_rows($res);

if ($count > 0) {

// Foods available

while ($row = mysqli\_fetch\_assoc($res)) {

// Get food details

$id = $row['id'];

$title = $row['title'];

$price = $row['price'];

$description = $row['description'];

$image\_name = $row['image\_name'];

?>

<div class="food-menu-box">

<div class="food-menu-img">

<?php if (!empty($image\_name)) : ?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name; ?>" alt="<?php echo htmlspecialchars($title); ?>" class="img-responsive img-curve">

<?php else : ?>

<div class='error'>Image not available.</div>

<?php endif; ?>

</div>

<div class="food-menu-desc">

<h4><?php echo htmlspecialchars($title); ?></h4>

<p class="food-price">$<?php echo $price; ?></p>

<p class="food-detail"><?php echo htmlspecialchars($description); ?></p>

<br>

<a href="<?php echo SITEURL; ?>order.php?food\_id=<?php echo $id; ?>" class="btn btn-primary">Order Now</a>

</div>

</div>

<?php

}

} else {

// No foods found

echo "<div class='error'>No foods found.</div>";

}

} else {

// Query execution failed

echo "<div class='error'>Failed to retrieve foods.</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!-- FOOD MENU Section Ends Here -->

<?php include('partials-front/footer.php'); ?>

</body>

</html>



<?php

include\_once('partials-front/menu.php');

// Start session if not already started

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>TASTE AWESOME FOOD</title>

<link rel="stylesheet" href="<?php echo SITEURL; ?>css/style.css">

</head>

<body>

<!-- FOOD SEARCH Section Starts Here -->

<section class="food-search text-center">

<div class="container">

<form action="<?php echo SITEURL; ?>food-search.php" method="POST">

<input type="search" name="search" placeholder="Search for Food.." required>

<input type="submit" name="submit" value="Search" class="btn btn-primary">

</form>

</div>

</section>

<!-- FOOD SEARCH Section Ends Here -->

<?php

// Display order message if set in session

if(isset($\_SESSION['order'])) {

echo "<div class='success'>" . $\_SESSION['order'] . "</div>";

unset($\_SESSION['order']);

}

?>

<!-- CATEGORIES Section Starts Here -->

<section class="categories">

<div class="container">

<h2 class="text-center">Explore Foods</h2>

<?php

// SQL query to select active and featured categories

$sql = "SELECT \* FROM tbl\_category WHERE active='Yes' AND featured='Yes' LIMIT 6";

// Execute the query

$res = mysqli\_query($conn, $sql);

// Check if categories are available

$count = mysqli\_num\_rows($res);

if ($count > 0) {

// Categories available

while ($row = mysqli\_fetch\_assoc($res)) {

// Get category details

$id = $row['id'];

$title = $row['title'];

$image\_name = $row['image\_name'];

?>

<a href="<?php echo SITEURL; ?>category-foods.php?category\_id=<?php echo $id; ?>">

<div class="box-3 float-container">

<?php if (!empty($image\_name)) : ?>

<img src="<?php echo SITEURL; ?>images/category/<?php echo $image\_name; ?>" alt="<?php echo htmlspecialchars($title); ?>" class="img-responsive img-curve">

<?php else : ?>

<div class='error'>Image not available</div>

<?php endif; ?>

<h3 class="float-text text-white"><?php echo htmlspecialchars($title); ?></h3>

</div>

</a>

<?php

}

} else {

// No categories found

echo "<div class='error'>No categories found.</div>";

}

?>

<div class="clearfix"></div>

</div>

</section>

<!-- CATEGORIES Section Ends Here -->

<!-- FOOD MENU Section Starts Here -->

<section class="food-menu">

<div class="container">

<h2 class="text-center">Food Menu</h2>

<?php

// SQL query to select active and featured foods

$sql2 = "SELECT \* FROM tbl\_food WHERE active='Yes' AND featured='Yes' LIMIT 6";

// Execute the query

$res2 = mysqli\_query($conn, $sql2);

// Check if foods are available

$count2 = mysqli\_num\_rows($res2);

if ($count2 > 0) {

// Foods available

while ($row2 = mysqli\_fetch\_assoc($res2)) {

// Get food details

$id = $row2['id'];

$title = $row2['title'];

$price = $row2['price'];

$description = $row2['description'];

$image\_name = $row2['image\_name'];

?>

<div class="food-menu-box">

<div class="food-menu-img">

<?php if (!empty($image\_name)) : ?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $image\_name; ?>" alt="<?php echo htmlspecialchars($title); ?>" class="img-responsive img-curve">

<?php else : ?>

<div class='error'>Image not available</div>

<?php endif; ?>

</div>

<div class="food-menu-desc">

<h4><?php echo htmlspecialchars($title); ?></h4>

<p class="food-price">$<?php echo $price; ?></p>

<p class="food-detail"><?php echo htmlspecialchars($description); ?></p>

<br>

<a href="<?php echo SITEURL; ?>order.php?food\_id=<?php echo $id; ?>" class="btn btn-primary">Order Now</a>

</div>

</div>

<?php

}

} else {

// No foods found

echo "<div class='error'>No foods found.</div>";

}

?>

<div class="clearfix"></div>

</div>

<p class="text-center">

<a href="<?php echo SITEURL; ?>foods.php">See All Foods</a>

</p>

</section>

<!-- FOOD MENU Section Ends Here -->

<?php include('partials-front/footer.php'); ?>

</body>

</html>



<?php

// Check if session is not already started

if (session\_status() == PHP\_SESSION\_NONE) {

// If not, start session

session\_start();

}

// Include menu.php to access constants and database connection

include\_once('partials-front/menu.php');

// Redirect to homepage if success message is not set

if (!isset($\_SESSION['success'])) {

header('location: ' . SITEURL);

exit;

}

// Clear success message from session after displaying

$success\_message = $\_SESSION['success'];

unset($\_SESSION['success']);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Order Successful - Food Order</title>

<link rel="stylesheet" href="<?php echo SITEURL; ?>css/style.css">

<style>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

.order-success {

padding: 50px 0;

text-align: center;

}

.success-message {

background-color: #fff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

padding: 30px;

max-width: 600px;

margin: 0 auto;

}

.success-message h2 {

color: #4CAF50;

font-size: 36px;

margin-bottom: 20px;

}

.success-message p {

font-size: 18px;

margin-bottom: 30px;

}

.btn-primary {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s;

text-decoration: none;

display: inline-block;

}

.btn-primary:hover {

background-color: #45a049;

}

.btn-primary:focus {

outline: none;

}

</style>

</head>

<body>

<!-- Success Message Section -->

<section class="order-success">

<div class="container">

<div class="success-message">

<h2>Order Placed Successfully!</h2>

<p><?php echo $success\_message; ?></p>

<a href="<?php echo SITEURL; ?>" class="btn btn-primary">Go to Home</a>

</div>

</div>

</section>

<!-- End Success Message Section -->

<?php include\_once('partials-front/footer.php'); ?>

</body>

</html>



<?php

// Start session if not already started

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

// Include menu.php to access constants and database connection

include\_once('partials-front/menu.php');

// Check if food\_id is set in GET parameters

if (isset($\_GET['food\_id'])) {

// Get food\_id from GET parameter and sanitize it

$food\_id = mysqli\_real\_escape\_string($conn, $\_GET['food\_id']);

// Fetch food details from the database

$sql = "SELECT \* FROM tbl\_food WHERE id = '$food\_id' AND active = 'Yes'";

$res = mysqli\_query($conn, $sql);

// Check if data is available

if ($res) {

$count = mysqli\_num\_rows($res);

if ($count == 1) {

// Food found, fetch details

$row = mysqli\_fetch\_assoc($res);

$food\_title = htmlspecialchars($row['title']);

$food\_price = $row['price'];

$food\_image = $row['image\_name'];

} else {

// Redirect to homepage if food not found or inactive

$\_SESSION['error'] = "The selected food item is not available.";

header('location: ' . SITEURL);

exit;

}

} else {

// Error fetching food details

$\_SESSION['error'] = "Failed to fetch food details. Please try again.";

header('location: ' . SITEURL);

exit;

}

} else {

// Redirect to homepage if food\_id is not set

$\_SESSION['error'] = "No food item selected.";

header('location: ' . SITEURL);

exit;

}

// Handle form submission

if (isset($\_POST['submit'])) {

// Sanitize and validate form data

$qty = isset($\_POST['qty']) ? (int)$\_POST['qty'] : 1;

$full\_name = mysqli\_real\_escape\_string($conn, $\_POST['full-name']);

$contact = mysqli\_real\_escape\_string($conn, $\_POST['contact']);

$email = mysqli\_real\_escape\_string($conn, $\_POST['email']);

$address = mysqli\_real\_escape\_string($conn, $\_POST['address']);

// Calculate total price

$total\_price = $food\_price \* $qty;

// Insert order into database

$sql\_order = "INSERT INTO `tbl\_order` (`food`, `price`, `qty`, `total`, `order\_date`, `status`, `customer\_name`, `customer\_contact`, `customer\_email`, `customer\_address`)

VALUES ('$food\_title', '$food\_price', '$qty', '$total\_price', NOW(), 'Ordered', '$full\_name', '$contact', '$email', '$address')";

$res\_order = mysqli\_query($conn, $sql\_order);

if ($res\_order) {

// Order placed successfully

$\_SESSION['success'] = "Order placed successfully. Your order will be delivered soon.";

header('location: ' . SITEURL . 'order-success.php');

exit;

} else {

// Failed to place order

$\_SESSION['error'] = "Failed to place order. Please try again.";

header('location: ' . SITEURL);

exit;

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Order - <?php echo htmlspecialchars($food\_title); ?></title>

<link rel="stylesheet" href="<?php echo SITEURL; ?>css/style.css">

<style>

/\* Additional styles can be added here specific to this page \*/

.food-search {

padding: 50px 0;

}

.food-menu-img {

text-align: center;

}

.food-menu-img img {

max-width: 100%;

height: auto;

}

.food-menu-desc {

margin-top: 20px;

}

.food-menu-desc h3 {

font-size: 24px;

margin-bottom: 10px;

}

.food-menu-desc .food-price {

font-size: 18px;

font-weight: bold;

color: #333;

margin-bottom: 10px;

}

.order-label {

font-weight: bold;

margin-top: 10px;

}

.input-responsive {

width: 100%;

padding: 8px;

margin-top: 6px;

margin-bottom: 12px;

border: 1px solid #ccc;

border-radius: 4px;

}

.btn-primary {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

transition: background-color 0.3s;

}

.btn-primary:hover {

background-color: #45a049;

}

.error {

color: red;

font-style: italic;

margin-top: 10px;

}

</style>

</head>

<body>

<!-- FOOD SEARCH Section Starts Here -->

<section class="food-search">

<div class="container">

<?php

// Display error messages if set in session

if (isset($\_SESSION['error'])) {

echo "<div class='error'>" . $\_SESSION['error'] . "</div>";

unset($\_SESSION['error']);

}

?>

<h2 class="text-center text-white">Fill this form to confirm your order for <?php echo htmlspecialchars($food\_title); ?>.</h2>

<form action="<?php echo htmlspecialchars($\_SERVER['PHP\_SELF']) . '?food\_id=' . $food\_id; ?>" class="order" method="POST">

<fieldset>

<legend>Selected Food</legend>

<div class="food-menu-img">

<?php if (!empty($food\_image)) : ?>

<img src="<?php echo SITEURL; ?>images/food/<?php echo $food\_image; ?>" alt="<?php echo htmlspecialchars($food\_title); ?>" class="img-responsive img-curve">

<?php else : ?>

<div class='error'>Image not found.</div>

<?php endif; ?>

</div>

<div class="food-menu-desc">

<h3><?php echo htmlspecialchars($food\_title); ?></h3>

<input type="hidden" name="food\_id" value="<?php echo $food\_id; ?>">

<input type="hidden" name="food\_title" value="<?php echo htmlspecialchars($food\_title); ?>">

<input type="hidden" name="food\_price" value="<?php echo $food\_price; ?>">

<p class="food-price">$<?php echo $food\_price; ?></p>

<div class="order-label">Quantity</div>

<input type="number" name="qty" class="input-responsive" value="1" min="1" required>

</div>

</fieldset>

<fieldset>

<legend>Delivery Details</legend>

<div class="order-label">Full Name</div>

<input type="text" name="full-name" placeholder="Enter your full name" class="input-responsive" required>

<div class="order-label">Phone Number</div>

<input type="tel" name="contact" placeholder="Enter your mobile number" class="input-responsive" required>

<div class="order-label">Email</div>

<input type="email" name="email" placeholder="Enter your email address" class="input-responsive" required>

<div class="order-label">Address</div>

<textarea name="address" rows="5" placeholder="Enter your delivery address" class="input-responsive" required></textarea>

<input type="submit" name="submit" value="Confirm Order" class="btn btn-primary">

</fieldset>

</form>

</div>

</section>

<!-- FOOD SEARCH Section Ends Here -->

<?php include\_once('partials-front/footer.php'); ?>

</body>

</html>

* **9.4 Detailed System Architecture**
  + Explanation of system components and their interactions.
  + Deployment architecture (server setup, web server configuration).

### **Software and Hardware Requirements**

* **Programming Language**: PHP 7.x
  + Utilizes PHP 7.x for server-side scripting, providing a robust and efficient backend for handling requests, sessions, and interactions with the database.
* **Database**: MySQL 5.x
  + MySQL 5.x is used for database management, ensuring reliable storage and retrieval of data, including user information, orders, food items, and categories.
* **Web Server**: Apache/Nginx
  + The application can be deployed on either Apache or Nginx web servers, which are both well-known for their stability, performance, and security.

**Hardware Requirements:**

* **Memory (RAM)**: Minimum 4GB RAM
  + Adequate memory ensures smooth operation, allowing the server to handle multiple concurrent requests and manage the application's processes efficiently.
* **Storage**: Minimum 20GB HDD
  + Sufficient storage is necessary to store the application files, database, and media assets such as food images and category icons.

**Operating System:**

* **Windows**
  + The application is designed to run on a Windows operating system, providing compatibility with a wide range of hardware and software environments.

**Deployment Details:**

* **Web Server Deployment**
  + The application should be deployed on a reliable web server that supports either Apache or Nginx. This ensures high availability, performance, and scalability.
* **SSL Encryption**
  + SSL (Secure Sockets Layer) encryption should be implemented to secure all transactions and data exchanges between the server and clients. This protects sensitive information such as user credentials, payment details, and order information from potential threats and unauthorized access.

**Benefits of Detailed Specifications:**

1. **Enhanced Security**: SSL encryption safeguards user data, ensuring that all interactions and transactions are secure.
2. **Optimized Performance**: Adequate hardware resources (4GB RAM and 20GB HDD) guarantee that the application runs smoothly, even under heavy load.
3. **Reliability**: Using established software such as PHP 7.x, MySQL 5.x, and Apache/Nginx ensures a stable and reliable environment for the application.
4. **Compatibility**: Running on Windows OS allows for broad compatibility with various deployment environments and simplifies integration with existing systems.

## **Testing and Quality Assurance**

## **To ensure the application is robust, reliable, and user-friendly, a comprehensive testing and quality assurance (QA) process is essential. This involves several types of testing to validate functionality, integration, user experience, and performance.**

### **1. Unit Testing**

## **Purpose:**

## **Validate individual functions and components to ensure they work as expected.**

## **Description:**

## **Each unit or component of the application, such as functions, methods, and classes, is tested independently.**

## **Developers write and run test cases to check that the code behaves correctly for a variety of inputs and conditions.**

## **This helps in identifying and fixing bugs at an early stage.**

## **Tools and Techniques:**

## **PHPUnit for PHP to automate and manage unit tests.**

## **Mock objects to simulate complex dependencies and ensure isolation of units.**

### **2. Integration Testing**

## **Purpose:**

## **Ensure different modules and components work together seamlessly.**

## **Description:**

## **After unit testing, individual modules are combined and tested as a group.**

## **This phase checks for issues in the interactions between integrated units, such as data flow and communication.**

## **Ensures that the integration of modules does not introduce new errors.**

## **Tools and Techniques:**

## **Automated scripts to test end-to-end scenarios involving multiple modules.**

## **Use of tools like Selenium to automate web application testing.**

### **3. User Acceptance Testing (UAT)**

## **Purpose:**

## **Solicit feedback from real users to improve usability and ensure the application meets their needs.**

## **Description:**

## **Actual users test the application in a real-world environment.**

## **This phase validates that the application performs correctly in the hands of end-users and meets business requirements.**

## **Feedback is gathered on the user experience, functionality, and any issues encountered.**

## **Tools and Techniques:**

## **Beta testing with a group of end-users.**

## **Surveys and feedback forms to collect user insights.**

## **Real-time observation and recording of user interactions.**

### **4. Performance Testing**

## **Purpose:**

## **Assess system response times, stability, and scalability under various load conditions.**

## **Description:**

## **Tests are conducted to measure how the application performs under expected and peak load conditions.**

## **Identifies potential bottlenecks, resource limits, and performance issues.**

## **Ensures that the application can handle high traffic volumes and large data processing efficiently.**

## **Tools and Techniques:**

## **Load testing tools like JMeter or LoadRunner to simulate multiple users and measure response times.**

## **Stress testing to push the system beyond its limits and identify breaking points.**

## **Monitoring tools to track resource usage (CPU, memory, disk I/O) during tests.**

### **Summary of Benefits**

## **Reliability: Ensures the application works correctly and consistently under different conditions.**

## **Usability: Validates that the application is user-friendly and meets user expectations.**

## **Performance: Guarantees that the application can handle the expected load without degradation of performance.**

## **Quality: Provides a comprehensive assessment of the application, ensuring high-quality deliverables.**

## 

## **12. Conclusion**

In conclusion, the Food Order Website stands as a testament to innovation and efficiency in the food service industry. By providing a secure, reliable, and user-centric platform, the website aims to redefine the food ordering experience, catering to the needs and expectations of modern consumers. With a commitment to quality and continuous improvement, the website is poised to deliver exceptional service and satisfaction to its users.

This concludes the overview of the Food Order Website, highlighting its technological foundation, user benefits, and future aspirations for ongoing development and enhancement.

## **13. References**

* Online resources used for PHP, MySQL, and web development best practices.
* Documentation and APIs of payment gateways integrated into the system.

This project documentation provides a comprehensive overview of the Food Order Website, covering its objectives, system architecture, database schema, features, implementation details, testing strategies, future enhancements, and references used during development. Adjust the details as per your specific project requirements and further developments.