**Online Food Ordering System (PHP + MySQL + JSON + AJAX)**

**Scenario**: Build an app called **“FoodieHub”** for online food ordering.

**Functional Requirements:**

* Menu fetched dynamically via **AJAX** from server and shown to user.
* Add to cart (with item count and total cost).
* Place an order and receive order ID confirmation.
* Admin Panel:
  + Add/edit/delete food items.
  + View live orders and update status ("Preparing", "Out for Delivery", "Delivered").
* Orders saved in **MySQL**, order status updated live with **AJAX polling**.
* Order details exportable as **JSON** file.
* Business Logic:
  + Delivery fee: ₹40 below ₹500, free otherwise.
  + Apply 15% GST on total food price.
* Create a folder FoodieHub inside xampp/htdocs (C://xampp/htdocs/FoodieHub)
* Add all codes inside the FoodieHub folder
* Create a database named foodiehub in the xampp phpmyadmin
* Inside the FoodieHub database, create tables menu and orders:

create table menu (

id int auto\_increment primary key,

name varchar(100) not null,

description text,

price decimal(8,2) not null

);

create table orders (

id int auto\_increment primary key,

items text not null,

total\_price double not null,

delivery\_fee double not null,

gst double not null,

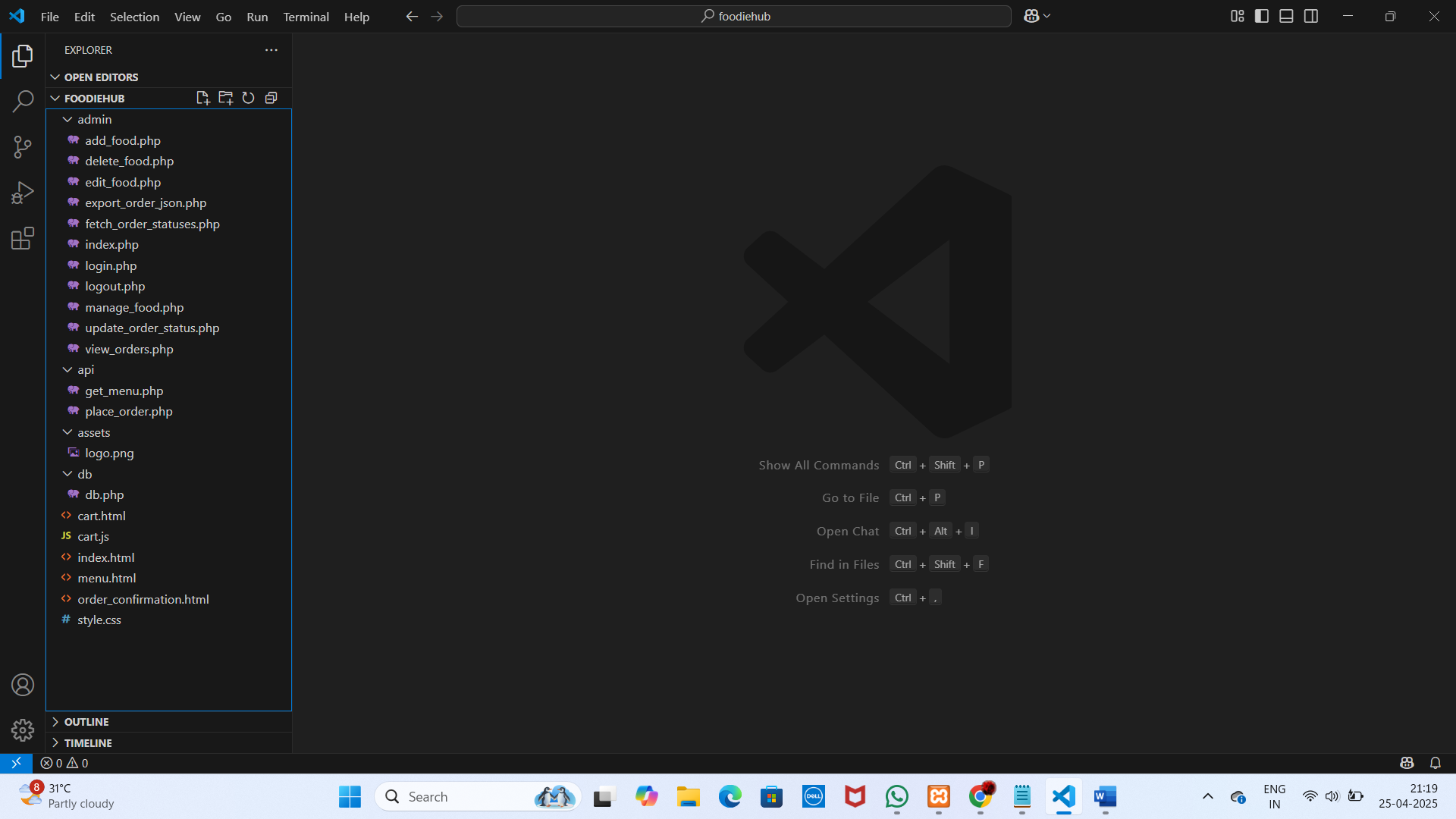
final\_amount double not null,

status varchar(50) default 'preparing',

order\_time datetime default current\_timestamp

);

Here is the FoodieHub Project Structure:



admin/

Files related to admin features (food management, orders)

* add\_food.php – Admin can add new food items to the menu.
* delete\_food.php – Deletes a food item from the menu.
* edit\_food.php – Edits/updates details of existing food items.
* export\_order\_json.php – Exports orders as a downloadable JSON file (admin side).
* fetch\_order\_statuses.php – Gets current statuses of orders.
* index.php – Admin dashboard/homepage
* manage\_food.php – Interface/page for managing all food items.
* update\_order\_status.php – Changes the status of an order
* view\_orders.php – Displays list of all customer orders to the admin.

api/

Files for API endpoints used by customer side (frontend):

* get\_menu.php – Sends the food menu (JSON format) to frontend (e.g., menu.html).
* place\_order.php – Receives and stores customer order data in the database.

db/

Database connection file:

* db.php – Contains database connection logic used by PHP files.

cart.html

Customer cart page where selected food items are shown before placing the order.

cart.js

JavaScript code that handles cart operations (adding/removing items, calculating total, etc.).

index.html

Main landing page for customers (e.g., welcome page or menu shortcut).

menu.html

Displays all available food items to the user with “Add to cart” options.

order\_confirmation.html

Shows confirmation message and order details after customer places an order.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Flow:

menu ➝ Add to cart ➝ Place order ➝ See confirmation.

admin ➝ View/manage orders ➝ Add/edit/delete food.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

db/db.php:

<?php

$host = "localhost";

$username = "root";

$password = "";

$database = "foodiehub";

$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect\_error) {

    die(" Connection failed: " . $conn->connect\_error);

}

?>

index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>FoodieHub - Online Food Ordering</title>

</head>

<body>

<header>

<h1>Welcome to FoodieHub</h1>

<a href="admin/index.php">Admin</a>

</header>

<main>

<h2>Explore Our Menu</h2>

<a href="menu.html">View Menu</a>

<a href="cart.html">Go to Cart</a>

</main>

<footer>

<p>&copy; 2025 FoodieHub</p>

</footer>

</body>

</html>

menu.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>FoodieHub - Menu</title>

</head>

<body>

<header>

<h1>FoodieHub - Menu</h1>

<a href="index.html">Home</a>

<a href="cart.html">Go to Cart</a>

</header>

<main>

<h2>Our Menu</h2>

<div id="menu"></div>

</main>

<footer>

<p>&copy; 2025 FoodieHub</p>

</footer>

<script>

fetch('api/get\_menu.php')

.then(response => response.json())

.then(data => {

let html = '';

data.forEach(item => {

html += `

<div>

<h3>${item.name}</h3>

<p>${item.description}</p>

<p>₹${item.price}</p>

<button onclick="updateQuantity(${item.id}, 'decrement')">-</button>

<span id="quantity-${item.id}">1</span>

<button onclick="updateQuantity(${item.id}, 'increment')">+</button>

<button onclick='addToCart(${JSON.stringify(item)})'>Add to Cart</button>

</div>

`;

});

document.getElementById('menu').innerHTML = html;

});

function updateQuantity(itemId, action) {

const quantityElement = document.getElementById(`quantity-${itemId}`);

let quantity = parseInt(quantityElement.innerText);

if (action === 'increment') {

quantity++;

} else if (action === 'decrement' && quantity > 1) {

quantity--;

}

quantityElement.innerText = quantity;

}

function addToCart(item) {

const quantity = parseInt(document.getElementById(`quantity-${item.id}`).innerText);

const cartItem = { ...item, quantity };

let cart = JSON.parse(localStorage.getItem('cart')) || [];

const existingItem = cart.find(cartItem => cartItem.id === item.id);

if (existingItem) {

existingItem.quantity += cartItem.quantity;

} else {

cart.push(cartItem);

}

localStorage.setItem('cart', JSON.stringify(cart));

alert(`${item.name} added to cart!`);

}

</script>

</body>

</html>

api/get\_menu.php:

<?php

header('Content-Type: application/json');

include('../db/db.php');

$result = $conn->query("SELECT \* FROM menu");

$menu = [];

while ($row = $result->fetch\_assoc()) {

    $menu[] = $row;

}

echo json\_encode($menu);

$conn->close();

?>

Cart.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>🛒 Your Cart - FoodieHub</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<h1>FoodieHub - 🛒 Your Cart</h1>

<div class="btn-container">

<a href="index.html" class="btn">Home</a>

<a href="menu.html" class="btn">Back to Menu</a>

</div>

</header>

<main class="container">

<div id="cart">

<!-- Cart content will be dynamically injected here -->

</div>

</main>

<footer>

<p>&copy; 2025 FoodieHub. Craving solved, one bite at a time 🍕</p>

</footer>

<script src="cart.js"></script>

</body>

</html>

Cart.js

let cart = JSON.parse(localStorage.getItem('cart')) || [];

function updateCart() {

    const cartContainer = document.getElementById('cart');

    let total\_price = 0;

    let html = '';

    if (cart.length > 0) {

        cart.forEach((item, index) => {

            const itemTotal = parseFloat(item.price) \* item.quantity;

            total\_price += itemTotal;

            html += `

                <div class="cart-item">

                    <h3>${item.name}</h3>

                    <p>Quantity: ${item.quantity}</p>

                    <p>Price: ₹${item.price} x ${item.quantity} = ₹${itemTotal.toFixed(2)}</p>

                    <button onclick="removeItem(${index})">Remove</button>

                </div>

            `;

        });

        const delivery\_fee = total\_price < 500 ? 40 : 0;

        const gst = Math.round(total\_price \* 0.15); // 15% GST

        const final\_amount = total\_price + gst + delivery\_fee;

        html += `

            <div class="cart-summary">

                <p>Total Price: ₹${total\_price.toFixed(2)}</p>

                <p>Delivery Fee: ₹${delivery\_fee}</p>

                <p>GST (15%): ₹${gst}</p>

                <p><strong>Final Amount: ₹${final\_amount.toFixed(2)}</strong></p>

            </div>

            <button class="place-order-btn" onclick="placeOrder()">Place Order</button>

        `;

    } else {

        html = `<p>Your cart is empty.</p>`;

    }

    cartContainer.innerHTML = html;

    localStorage.setItem('cart', JSON.stringify(cart));

}

function removeItem(index) {

    cart.splice(index, 1);

    updateCart();

}

function placeOrder() {

    if (cart.length === 0) {

        alert("Cart is empty!");

        return;

    }

    let total\_price = 0;

    cart.forEach(item => {

        total\_price += parseFloat(item.price) \* item.quantity;

    });

    const delivery\_fee = total\_price < 500 ? 40 : 0;

    const gst = Math.round(total\_price \* 0.15);

    const final\_amount = total\_price + gst + delivery\_fee;

    const itemsList = cart.map(item => `${item.name} (x${item.quantity})`).join(", ");

    const orderData = {

        items: itemsList,

        total\_price,

        delivery\_fee,

        gst,

        final\_amount

    };

    fetch('api/place\_order.php', {

        method: 'POST',

        headers: { 'Content-Type': 'application/json' },

        body: JSON.stringify(orderData)

    })

    .then(res => res.json())

    .then(response => {

        if (response.success) {

            // Save order details in localStorage for confirmation page

            localStorage.setItem('last\_order\_items', itemsList);

            localStorage.setItem('last\_order\_total', total\_price);

            localStorage.setItem('last\_order\_delivery', delivery\_fee);

            localStorage.setItem('last\_order\_gst', gst);

            localStorage.setItem('last\_order\_final', final\_amount);

            // Clear cart and redirect

            cart = [];

            localStorage.removeItem('cart');

            updateCart();

            window.location.href = `order\_confirmation.html?order\_id=${response.order\_id}`;

        }

         else {

            alert("Failed to place order: " + response.error);

        }

    })

    .catch(error => {

        console.error("Error placing order:", error);

        alert("Failed to place order.");

    });

}

updateCart();

api/place\_order.php:

<?php

header('Content-Type: application/json');

header('Access-Control-Allow-Origin: \*');

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

include('../db/db.php');

$data = json\_decode(file\_get\_contents("php://input"), true);

if (!$data) {

    echo json\_encode(["success" => false, "error" => "Invalid JSON"]);

    exit;

}

$items = $data['items'];

$total\_price = floatval($data['total\_price']);

$delivery\_fee = floatval($data['delivery\_fee']);

$gst = floatval($data['gst']);

$final\_amount = floatval($data['final\_amount']);

$status = isset($data['status']) ? $conn->real\_escape\_string($data['status']) : null;

if (empty($items)) {

    echo json\_encode(["success" => false, "error" => "'items' cannot be empty"]);

    exit;

}

$items = $conn->real\_escape\_string($items);

if ($status) {

    $sql = "INSERT INTO `orders` (items, total\_price, delivery\_fee, gst, final\_amount, status)

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

    $stmt = $conn->prepare($sql);

    $stmt->bind\_param("sdddds", $items, $total\_price, $delivery\_fee, $gst, $final\_amount, $status);

} else {

    $sql = "INSERT INTO `orders` (items, total\_price, delivery\_fee, gst, final\_amount)

            VALUES (?, ?, ?, ?, ?)";

    $stmt = $conn->prepare($sql);

    $stmt->bind\_param("sdddd", $items, $total\_price, $delivery\_fee, $gst, $final\_amount);

}

if ($stmt === false) {

    echo json\_encode(["success" => false, "error" => "Failed to prepare statement"]);

    exit;

}

if ($stmt->execute()) {

    echo json\_encode(["success" => true, "order\_id" => $stmt->insert\_id]);

} else {

    echo json\_encode(["success" => false, "error" => $stmt->error]);

}

$stmt->close();

?>

Order\_confirmation.php

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title> FoodieHub - Order Confirmation</title>

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

<link rel="stylesheet" href="style.css">

</head>

<body>

<header>

<h1> FoodieHub - Order Confirmation</h1>

</header>

<main>

<div class="confirmation">

<h2>Order placed successfully!</h2>

<div class="order-info" id="order-details">

<!-- Dynamic content here -->

</div>

<button id="export-json">Download Order Summary</button>

<div class="home-link">

<a href="index.html">← Back to Home</a>

</div>

</div>

</main>

<footer>

<p>&copy; 2025 FoodieHub. </p>

</footer>

<script>

const params = new URLSearchParams(window.location.search);

const orderId = params.get('order\_id');

const orderItems = localStorage.getItem('last\_order\_items');

const totalPrice = localStorage.getItem('last\_order\_total');

const deliveryFee = localStorage.getItem('last\_order\_delivery');

const gst = localStorage.getItem('last\_order\_gst');

const finalAmount = localStorage.getItem('last\_order\_final');

const details = document.getElementById('order-details');

if (orderId && orderItems && totalPrice && gst && finalAmount) {

details.innerHTML = `

<p><strong>Order ID:</strong> #${orderId}</p>

<p><strong>Items:</strong> ${orderItems}</p>

<p><strong>Total Price:</strong> ₹${parseFloat(totalPrice).toFixed(2)}</p>

<p><strong>Delivery Fee:</strong> ₹${parseFloat(deliveryFee).toFixed(2)}</p>

<p><strong>GST (15%):</strong> ₹${parseFloat(gst).toFixed(2)}</p>

<p><strong>Final Amount:</strong> ₹${parseFloat(finalAmount).toFixed(2)}</p>

`;

} else {

details.innerHTML = `<p>Order details not found.</p>`;

}

localStorage.removeItem('last\_order\_items');

localStorage.removeItem('last\_order\_total');

localStorage.removeItem('last\_order\_delivery');

localStorage.removeItem('last\_order\_gst');

localStorage.removeItem('last\_order\_final');

document.getElementById('export-json').addEventListener('click', function() {

const orderData = {

orderId: orderId,

items: orderItems,

totalPrice: parseFloat(totalPrice),

deliveryFee: parseFloat(deliveryFee),

gst: parseFloat(gst),

finalAmount: parseFloat(finalAmount)

};

const jsonBlob = new Blob([JSON.stringify(orderData, null, 2)], { type: 'application/json' });

const link = document.createElement('a');

link.href = URL.createObjectURL(jsonBlob);

link.download = `order\_${orderId}.json`;

link.click();

});

</script>

</body>

</html>

Admin Dashboard (index.php)

<?php

session\_start();

require\_once '../db/db.php';

?>

<!DOCTYPE html>

<html>

<head>

<title>Admin Dashboard - FoodieHub</title>

</head>

<body>

<h1>Admin Dashboard - FoodieHub</h1>

<p>Welcome Admin!</p>

<ul>

<li><a href="manage\_food.php">Manage Food Items</a></li>

<li><a href="view\_orders.php">View Orders</a></li>

</ul>

<p>&copy; 2025 FoodieHub</p>

</body>

</html>

<?php

$conn->close();

?>

manage\_food.php (Admin - Add, Edit, Delete):

<?php

session\_start();

require\_once '../db/db.php';

$sql = "SELECT \* FROM menu";

$result = $conn->query($sql);

?>

<!DOCTYPE html>

<html>

<head>

<title>Manage Food Items</title>

</head>

<body>

<h1>Manage Food Items</h1>

<a href="add\_food.php">Add New Food Item</a>

<table border="1" cellpadding="8" cellspacing="0">

<tr>

<th>Food ID</th>

<th>Name</th>

<th>Description</th>

<th>Price</th>

<th>Actions</th>

</tr>

<?php

if ($result->num\_rows > 0) {

while ($row = $result->fetch\_assoc()) {

echo "<tr>

<td>{$row['id']}</td>

<td>{$row['name']}</td>

<td>{$row['description']}</td>

<td>{$row['price']}</td>

<td>

<a href='edit\_food.php?id={$row['id']}'>Edit</a> |

<a href='delete\_food.php?id={$row['id']}'>Delete</a>

</td>

</tr>";

}

} else {

echo "<tr><td colspan='5'>No food items found.</td></tr>";

}

?>

</table>

</body>

</html>

<?php

$conn->close();

?>

add\_food.php:

<?php

require\_once '../db/db.php';

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

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

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

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

$sql = "INSERT INTO menu (name, description, price) VALUES (?, ?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("ssd", $name, $description, $price);

if ($stmt->execute()) {

header("Location: manage\_food.php");

exit();

} else {

echo "Error adding food item: " . $conn->error;

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Add Food Item - FoodieHub</title>

</head>

<body>

<h1>Add New Food Item</h1>

<form method="POST">

<label for="name">Food Name:</label>

<input type="text" name="name" required><br><br>

<label for="description">Description:</label>

<textarea name="description" required></textarea><br><br>

<label for="price">Price:</label>

<input type="number" name="price" required><br><br>

<button type="submit">Add Food Item</button>

</form>

</body>

</html>

<?php

$conn->close();

?>

edit\_food.php

<?php

require\_once '../db/db.php';

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

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

$sql = "SELECT \* FROM menu WHERE id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $id);

$stmt->execute();

$result = $stmt->get\_result();

$food\_item = $result->fetch\_assoc();

}

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

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

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

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

$update\_sql = "UPDATE menu SET name = ?, description = ?, price = ? WHERE id = ?";

$update\_stmt = $conn->prepare($update\_sql);

$update\_stmt->bind\_param("ssdi", $name, $description, $price, $id);

$update\_stmt->execute();

header("Location: manage\_food.php");

exit();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Edit Food Item - FoodieHub</title>

</head>

<body>

<h1>Edit Food Item</h1>

<h2>Edit Food Item: <?php echo htmlspecialchars($food\_item['name']); ?></h2>

<form method="POST">

<label for="name">Food Name:</label>

<input type="text" name="name" value="<?php echo htmlspecialchars($food\_item['name']); ?>" required><br><br>

<label for="description">Description:</label>

<textarea name="description" required><?php echo htmlspecialchars($food\_item['description']); ?></textarea><br><br>

<label for="price">Price:</label>

<input type="number" name="price" value="<?php echo htmlspecialchars($food\_item['price']); ?>" required><br><br>

<button type="submit">Update Food Item</button>

</form>

</body>

</html>

<?php

$conn->close();

?>

delete\_food.php

<?php

session\_start();

require\_once '../db/db.php';

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

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

$sql = "DELETE FROM menu WHERE id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $id);

if ($stmt->execute()) {

header("Location: manage\_food.php");

exit();

} else {

echo "Error deleting food item: " . $conn->error;

}

} else {

echo "Food item ID is required!";

}

$conn->close();

?>

View Orders (Admin – update statuses)

view\_orders.php

<?php

session\_start();

require\_once '../db/db.php';

$status\_filter = isset($\_GET['status']) ? $\_GET['status'] : '';

$sql = "SELECT \* FROM orders";

if ($status\_filter) {

$sql .= " WHERE status = ?";

}

$stmt = $conn->prepare($sql);

if ($status\_filter) {

$stmt->bind\_param("s", $status\_filter);

}

$stmt->execute();

$result = $stmt->get\_result();

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>View Orders</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script>

$(document).ready(function() {

$('.update-status').on('click', function() {

var orderId = $(this).data('order-id');

var status = $(this).data('status');

// Send an AJAX request to update the status

$.ajax({

url: 'update\_order\_status.php',

type: 'POST',

data: { order\_id: orderId, status: status },

success: function(response) {

// Update the status on the page without refreshing

$('#status\_' + orderId).text(response);

},

error: function() {

alert('Error updating status.');

}

});

});

// AJAX polling to fetch live order statuses every 10 seconds

setInterval(function() {

$.ajax({

url: 'fetch\_order\_statuses.php',

method: 'GET',

dataType: 'json',

success: function(data) {

data.forEach(function(order) {

const statusCell = $('#status\_' + order.id);

if (statusCell.length && statusCell.text().trim() !== order.status) {

statusCell.text(order.status);

}

});

},

error: function() {

console.error('Failed to fetch live order statuses.');

}

});

}, 10000);

});

</script>

</head>

<body>

<div>

<h2>All Orders</h2>

<div>

<form method="get" action="view\_orders.php">

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

<select name="status" id="status">

<option value="">All</option>

<option value="Preparing" <?php echo ($status\_filter == 'Preparing') ? 'selected' : ''; ?>>Preparing</option>

<option value="Out for Delivery" <?php echo ($status\_filter == 'Out for Delivery') ? 'selected' : ''; ?>>Out for Delivery</option>

<option value="Delivered" <?php echo ($status\_filter == 'Delivered') ? 'selected' : ''; ?>>Delivered</option>

</select>

<button type="submit">Filter</button>

</form>

</div>

<table>

<thead>

<tr>

<th>Order ID</th>

<th>Items</th>

<th>Amount</th>

<th>Status</th>

<th>Update / Export</th>

</tr>

</thead>

<tbody>

<?php

if ($result->num\_rows > 0) {

while($row = $result->fetch\_assoc()) {

echo "<tr>

<td>" . $row['id'] . "</td>

<td>" . $row['items'] . "</td>

<td>" . $row['final\_amount'] . "</td>

<td id='status\_" . $row['id'] . "'>" . $row['status'] . "</td>

<td>

<div>

<button data-order-id='" . $row['id'] . "' data-status='Preparing' class='update-status'>Preparing</button>

<button data-order-id='" . $row['id'] . "' data-status='Out for Delivery' class='update-status'>Out for Delivery</button>

<button data-order-id='" . $row['id'] . "' data-status='Delivered' class='update-status'>Delivered</button>

</div>

<div>

<a href='export\_order\_json.php?order\_id=" . $row['id'] . "'>Download Order Summary</a>

</div>

</td>

</tr>";

}

} else {

echo "<tr><td colspan='5'>No orders found.</td></tr>";

}

?>

</tbody>

</table>

</div>

</body>

</html>

<?php

$conn->close();

?>

update\_order\_status.php

<?php

require\_once '../db/db.php';

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

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

    $status = $\_POST['status'];

    $order\_id = intval($order\_id);

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

    $sql = "UPDATE orders SET status = ? WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bind\_param("si", $status, $order\_id);

    if ($stmt->execute()) {

        echo $status;

    } else {

        echo "Error updating status";

    }

    $stmt->close();

} else {

    echo "Invalid request.";

}

$conn->close();

?>

fetch\_order\_statuses.php

<?php

require\_once '../db/db.php';

$sql = "SELECT id, status FROM orders";

$result = $conn->query($sql);

$orders = [];

if ($result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        $orders[] = [

            'id' => $row['id'],

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

        ];

    }

}

header('Content-Type: application/json');

echo json\_encode($orders);

?>

export\_order\_json.php

<?php

require\_once '../db/db.php';

if (isset($\_GET['order\_id'])) {

    $order\_id = $\_GET['order\_id'];

    $sql = "SELECT \* FROM orders WHERE id = ?";

    $stmt = $conn->prepare($sql);

    $stmt->bind\_param("i", $order\_id);

    $stmt->execute();

    $result = $stmt->get\_result();

    if ($result->num\_rows > 0) {

        $order = $result->fetch\_assoc();

        $order\_data = array(

            'order\_id' => $order['id'],

            'items' => $order['items'],

            'amount' => $order['final\_amount'],

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

            'timestamp' => $order['timestamp'],

        );

        header('Content-Type: application/json');

        header('Content-Disposition: attachment; filename="order\_' . $order\_id . '.json"');

        // Output the order data as JSON

        echo json\_encode($order\_data, JSON\_PRETTY\_PRINT);

    } else {

        echo json\_encode(array("error" => "Order not found."));

    }

    $stmt->close();

} else {

    echo json\_encode(array("error" => "No order ID specified."));

}

$conn->close();

?>