Restaurant Management System

CC223 – Application Development and Emerging Technologies

Instructor

Prof. Darwin Vargas

Group Members

Algara, Andrea Mikaela A. (TUPM-23-2133)

Laurente, Althea Marie S. (TUPM-23-2082)

Lesigues, Ian Lester D. (TUPM-23-2131)

Ramilo, Kirby D. (TUPM-23-1691)

BSIT – 2A

07/04/2025

Introduction

Overview of the Project

Main purpose of this project is to allow application of PHP and MySQL in creating a full-stack CRUD system to enable project-based learning. This project mainly focuses on creating a restaurant management system which aims to help a restaurant handle CRUD application in their establishment. This system is useful especially in tracking orders for customers and helping businesses handle management and customer experience much better.

Objectives

- Develop a CRUD-based restaurant management system where admins can add, edit, delete, and view menu and order records by dashboards
- Develop a website to handle customer transactions
- Develop a menu dashboard and order dashboard to handle menu items and order status
- Track transactions and display them using graphs

System Features

List of Functionalities

Admin

- Admin can create new menu items
- Admin can view records in a table
- Admin can update/edit existing menu items
- Admin can delete menu items from the database
- Admin needs to login in landing page to access dashboards
- Admin can view order details of customers
- Admin can update order status of customers

Customer

- Customers can select dine in or take out in landing page

- Customers can select menu items for their order
- Customers can access sidebar for specific categories of menu items
- Customers can check out their order and be showed a receipt
- Customers can download receipt to show to counter

Target Users

Who will use the system?

Admin: Track menu and orders using dashboards. Can add, edit, delete, and display menu items. Can view order items of customers and update their order status. Can track numerous data for statistical applications.

Customer: Ordering food in the restaurant. Pick between dine in or take out and selects their wanted food. Following checkout, they are prompted a receipt to show to counter.

Technology Stack

Frontend:

- **HTML** (Forms for data input)
- **CSS** (Custom styling)
- **PHP** (Dynamic page regeneration)
- **Bootstrap 5** (Styling and responsive design)
- **SweatAlert2 (SWAL)** (User-friendly modals)

Backend:

- **PHP** and **MySQL** (Handling database operations)

Database:

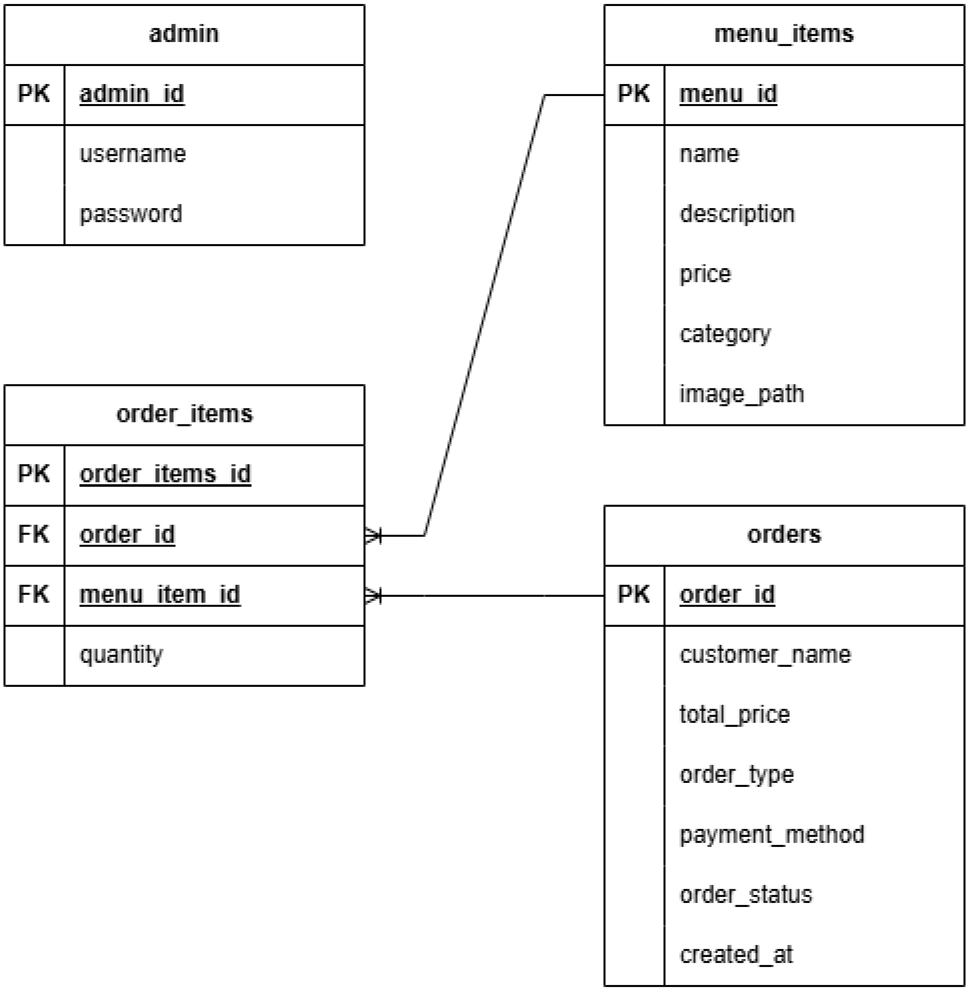
- **MySQL** (Data storage)

Tools:

- **XAMPP** (Local development)

- **phpMyAdmin** Database management)
- **VS Code** (Coding environment)

Database Design



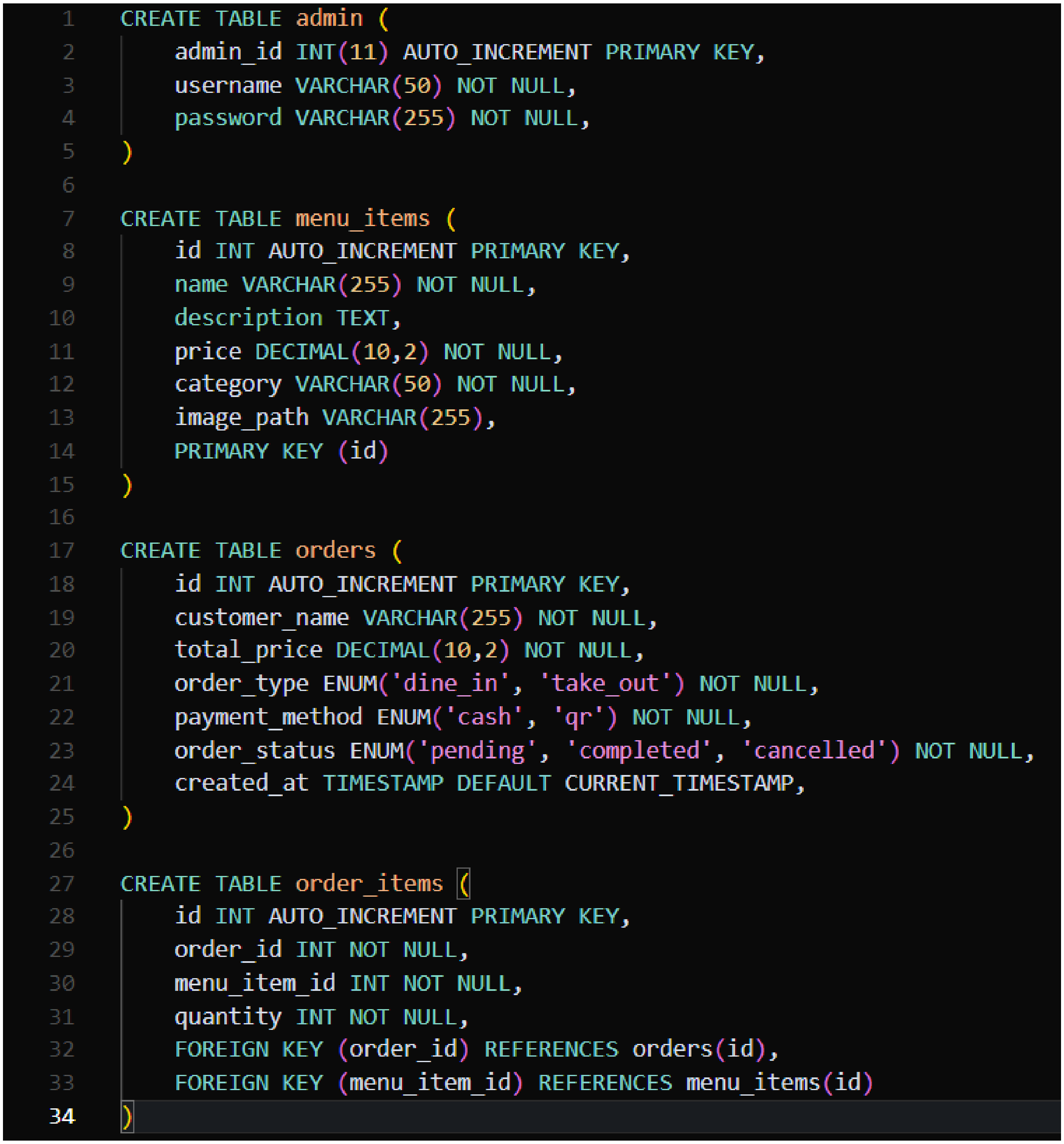
	admin
PK	admin_id
	username
	password

	menu_items
PK	menu_id
	name
	description
	price
	category
	image_path

	order_items
PK	order_items_id
FK	order_id
FK	menu_item_id
	quantity

	orders
PK	order_id
	customer_name
	total_price
	order_type
	payment_method
	order_status
	created_at

Database Schema



```
CREATE TABLE admin (
    admin_id INT(11) AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL,
)

CREATE TABLE menu_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    description TEXT,
    price DECIMAL(10,2) NOT NULL,
    category VARCHAR(50) NOT NULL,
    image_path VARCHAR(255),
    PRIMARY KEY (id)
)

CREATE TABLE orders (
    id INT AUTO_INCREMENT PRIMARY KEY,
    customer_name VARCHAR(255) NOT NULL,
    total_price DECIMAL(10,2) NOT NULL,
    order_type ENUM('dine_in', 'take_out') NOT NULL,
    payment_method ENUM('cash', 'qr') NOT NULL,
    order_status ENUM('pending', 'completed', 'cancelled') NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
)

CREATE TABLE order_items (
    id INT AUTO_INCREMENT PRIMARY KEY,
    order_id INT NOT NULL,
    menu_item_id INT NOT NULL,
    quantity INT NOT NULL,
    FOREIGN KEY (order_id) REFERENCES orders(id),
    FOREIGN KEY (menu_item_id) REFERENCES menu_items(id)
)
```

System Implementation

/restaurant_management

|-- /addMenu.php -> Form for adding new menu in database and menu.php.

|-- /check_admin.php –> Checks the admin credentials for admin login.

|-- /config.php -> Database connection.

|-- /delete_menu.php -> Deletes a menu item from the database.

|-- /edit.php -> Edits a menu item in the database. Used along with get_menu_item.php for editing.

|-- /get_menu_item.php -> Gets a menu item from the database. Used in menu_dashboard.php.

|-- /get_order_details.php -> Like receipt.php but on admin side, using order_dashboard.php.

|-- /index.php -> Main dashboard for the admin. Manages menu and order dashboards.

|-- /mainPage.php -> Main landing page of customers. Can be used to login admin.

|-- /menu_dashboard.php -> Menu dashboard. Admin can update, edit, or delete and add item.

|-- /menu.php -> Main menu page for the customer. View menu, order food, categories, checkout,

|-- /order_dashboard.php -> Order dashboard. Admin can view order details, update order status.

|-- /processOrder.php -> Used by menu.php to process the order of the customer.

|-- /receipt.php -> Receipt of the order of the customer. Can download receipt.

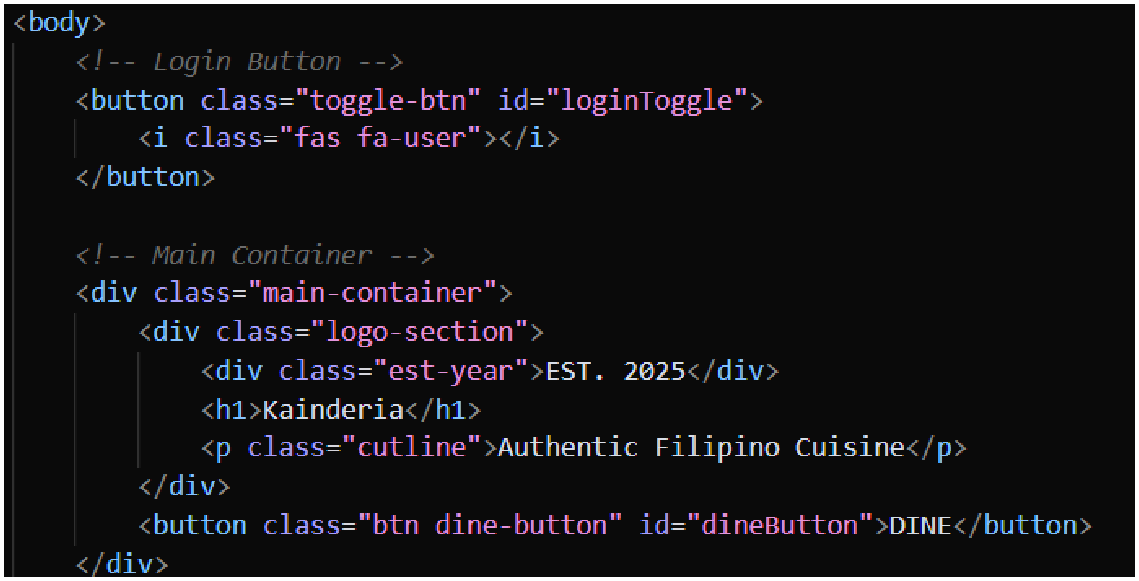
|-- /store.php -> Stores the menu item to the database. Used by addMenu.php to add a menu item.

|-- /update_order_status.php -> Used by order_dashboard.php to update order status of order.

Code Snippets and Explanation

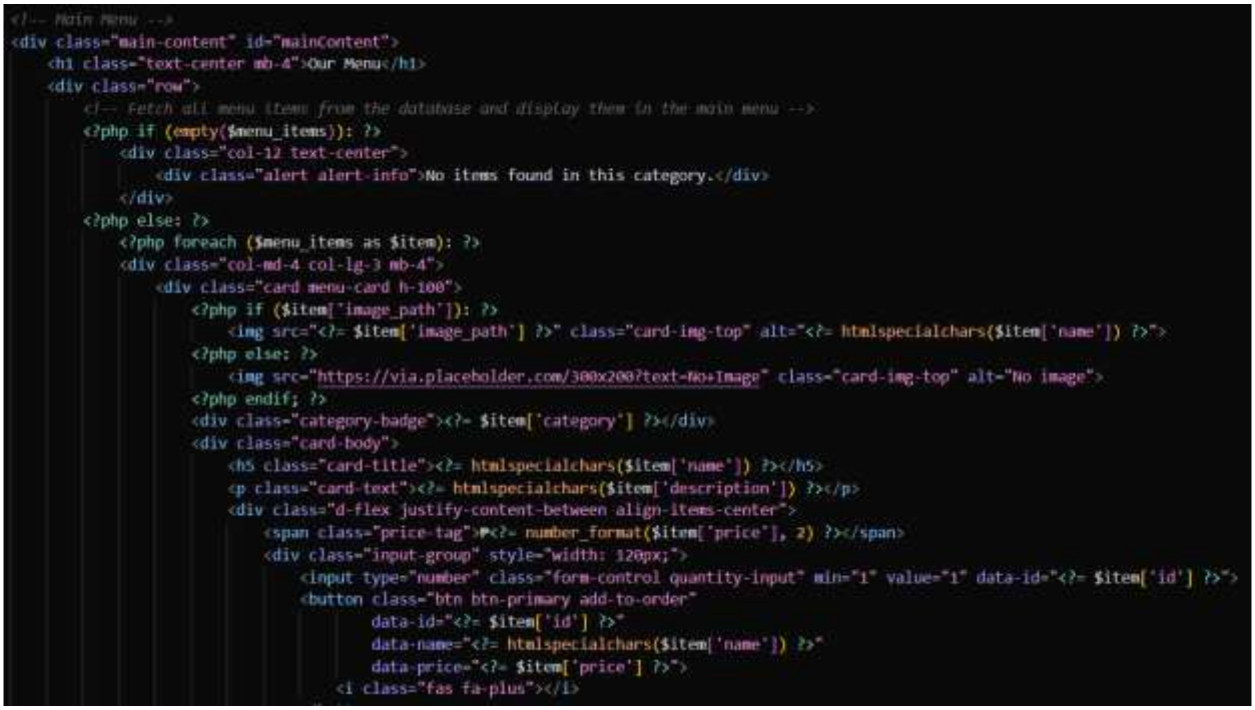
Frontend

mainPage.php -> Main landing page for the user. Used to login admin by login button.



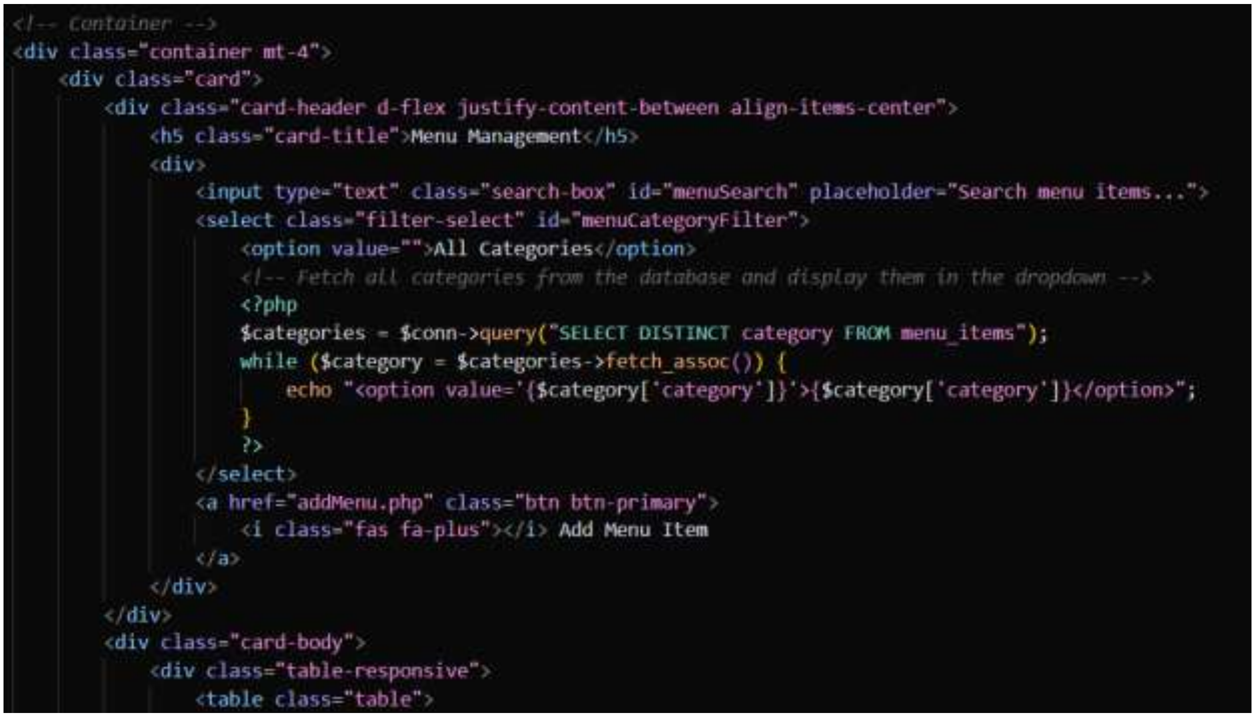
```
<body>
    <!-- Login Button -->
    <button class="toggle-btn" id="loginToggle">
        <i class="fas fa-user"></i>
    </button>

    <!-- Main Container -->
    <div class="main-container">
        <div class="logo-section">
            <div class="est-year">EST. 2025</div>
            <h1>Kainderia</h1>
            <p class="cutline">Authentic Filipino Cuisine</p>
        </div>
        <button class="btn dine-button" id="dineButton">DINE</button>
    </div>
```

menu.php -> Shows menu items. Has sidebar for categories, cart for order and checkout button.



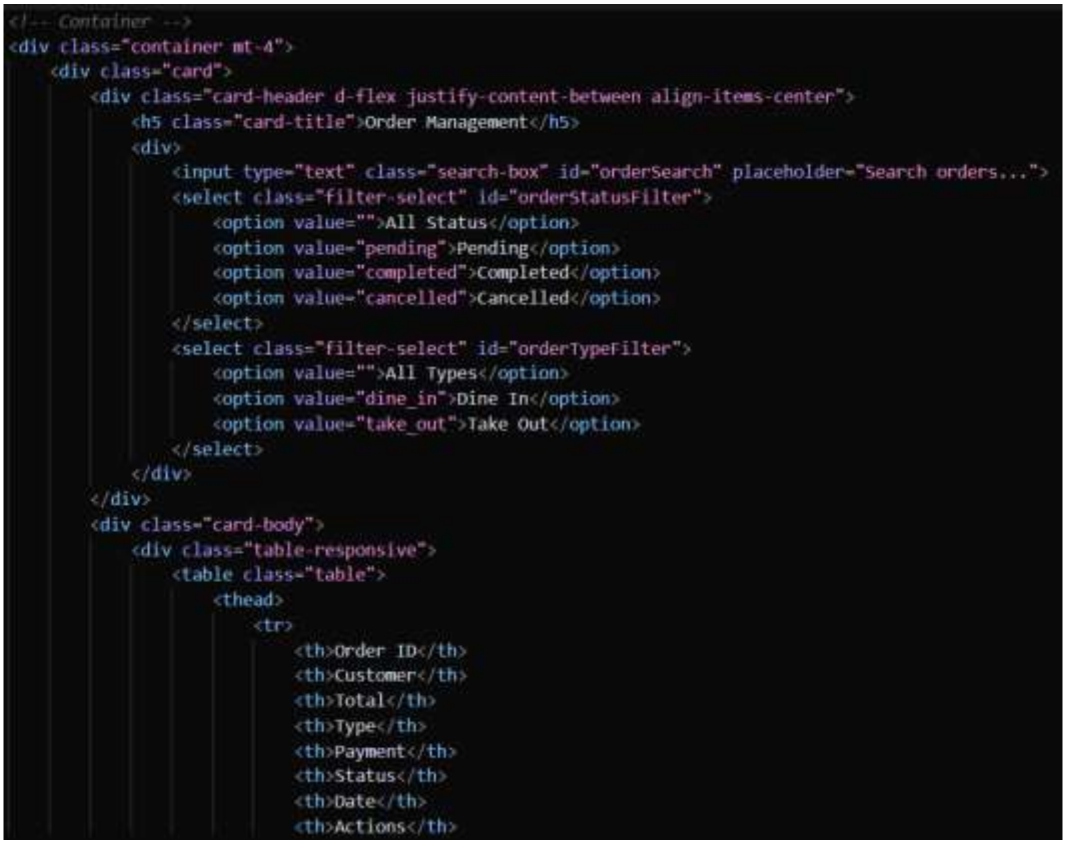
```
<!-- Main Menu -->
<div class="main-content" id="mainContent">
    <h1 class="text-center mb-4">Our Menu</h1>
    <div class="row">
        <!-- Fetch all menu items from the database and display them in the main menu -->
        <?php if (empty($menu_items)): ?>
            <div class="col-12 text-center">
                <div class="alert alert-info">No items found in this category.</div>
            </div>
        <?php else: ?>
            <?php foreach ($menu_items as $item): ?>
            <div class="col-md-4 col-lg-3 mb-4">
                <div class="card menu-card h-100">
                    <?php if ($item['image_path']): ?>
                        <img src="<?= $item['image_path'] ?>" class="card-img-top" alt="<?= htmlspecialchars($item['name']) ?>">
                    <?php else: ?>
                        <img src="https://via.placeholder.com/300x200?text=No+Image" class="card-img-top" alt="No image">
                    <?php endif; ?>
                    <div class="category-badge"><?= $item['category'] ?></div>
                    <div class="card-body">
                        <h5 class="card-title"><?= htmlspecialchars($item['name']) ?></h5>
                        <p class="card-text"><?= htmlspecialchars($item['description']) ?></p>
                        <div class="d-flex justify-content-between align-items-center">
                            <span class="price-tag">₱<?= number_format($item['price'], 2) ?></span>
                            <div class="input-group" style="width: 120px;">
                                <input type="number" class="form-control quantity-input" min="1" value="1" data-id="<?= $item['id'] ?>">
                                <button class="btn btn-primary add-to-order"
                                        data-id="<?= $item['id'] ?>"
                                        data-name="<?= htmlspecialchars($item['name']) ?>"
                                        data-price="<?= $item['price'] ?>">
                                    <i class="fas fa-plus"></i>
```

menu_dashboard.php -> Menu dashboard for admin. Shows menu_items table, add menu item option, search bar, and filtering options



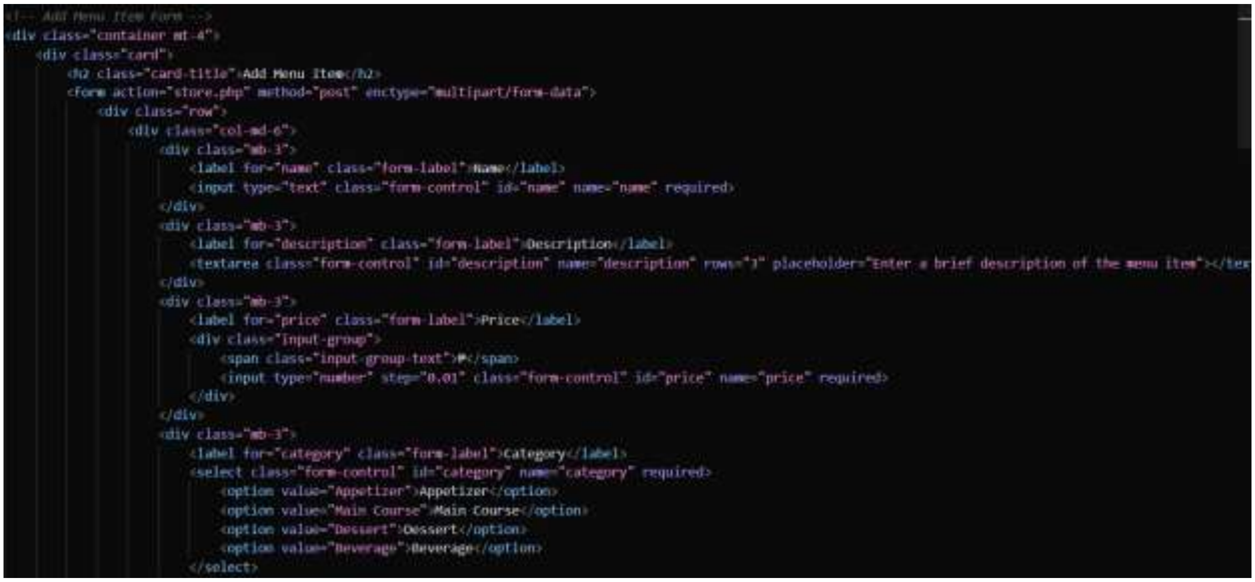
```
<!-- Container -->
<div class="container mt-4">
    <div class="card">
        <div class="card-header d-flex justify-content-between align-items-center">
            <h5 class="card-title">Menu Management</h5>
            <div>
                <input type="text" class="search-box" id="menuSearch" placeholder="Search menu items...">
                <select class="filter-select" id="menuCategoryFilter">
                    <option value="">All Categories</option>
                    <!-- Fetch all categories from the database and display them in the dropdown -->
                    <?php
                    $categories = $conn->query("SELECT DISTINCT category FROM menu_items");
                    while ($category = $categories->fetch_assoc()) {
                        echo "<option value='{$category['category']}'>{$category['category']}</option>";
                    }
                    ?>
                </select>
                <a href="addMenu.php" class="btn btn-primary">
                    <i class="fas fa-plus"></i> Add Menu Item
                </a>
            </div>
        </div>
        <div class="card-body">
            <div class="table-responsive">
                <table class="table">
```

order_dashboard.php -> Order dashboard for admin. Shows orders table, search bar, filtering options. Can view order details and update order status.



```
<!-- Container -->
<div class="container mt-4">
    <div class="card">
        <div class="card-header d-flex justify-content-between align-items-center">
            <h5 class="card-title">Order Management</h5>
            <div>
                <input type="text" class="search-box" id="orderSearch" placeholder="Search orders...">
                <select class="filter-select" id="orderStatusFilter">
                    <option value="">All Status</option>
                    <option value="pending">Pending</option>
                    <option value="completed">Completed</option>
                    <option value="cancelled">Cancelled</option>
                </select>
                <select class="filter-select" id="orderTypeFilter">
                    <option value="">All Types</option>
                    <option value="dine_in">Dine In</option>
                    <option value="take_out">Take Out</option>
                </select>
            </div>
        </div>
        <div class="card-body">
            <div class="table-responsive">
                <table class="table">
                    <thead>
                        <tr>
                            <th>Order ID</th>
                            <th>Customer</th>
                            <th>Total</th>
                            <th>Type</th>
                            <th>Payment</th>
                            <th>Status</th>
                            <th>Date</th>
                            <th>Actions</th>
```

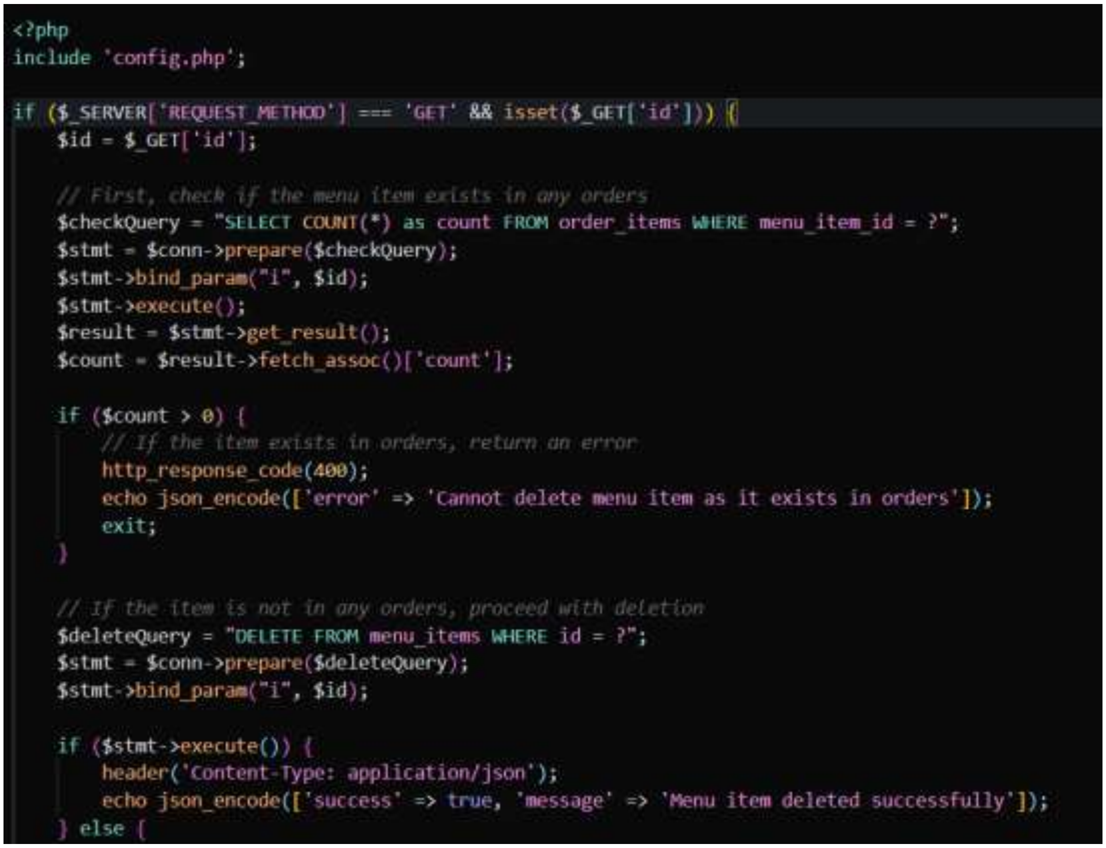
addMenu.php -> Add menu form. Requires menu item name, description, price, category, and image. Added menu item is reflected on the database and menu.



```html
<!-- Add Menu Item Form -->
<div class="container mt-4">
    <div class="card">
        <h2 class="card-title">Add Menu Item</h2>
        <form action="store.php" method="post" enctype="multipart/form-data">
            <div class="row">
                <div class="col-md-6">
                    <div class="mb-3">
                        <label for="name" class="form-label">Name</label>
                        <input type="text" class="form-control" id="name" name="name" required>
                    </div>
                    <div class="mb-3">
                        <label for="description" class="form-label">Description</label>
                        <textarea class="form-control" id="description" name="description" rows="3" placeholder="Enter a brief description of the menu item"></tex
                    </div>
                    <div class="mb-3">
                        <label for="price" class="form-label">Price</label>
                        <div class="input-group">
                            <span class="input-group-text">₱</span>
                            <input type="number" step="0.01" class="form-control" id="price" name="price" required>
                        </div>
                    </div>
                    <div class="mb-3">
                        <label for="category" class="form-label">Category</label>
                        <select class="form-control" id="category" name="category" required>
                            <option value="Appetizer">Appetizer</option>
                            <option value="Main Course">Main Course</option>
                            <option value="Dessert">Dessert</option>
                            <option value="Beverage">Beverage</option>
                        </select>
```

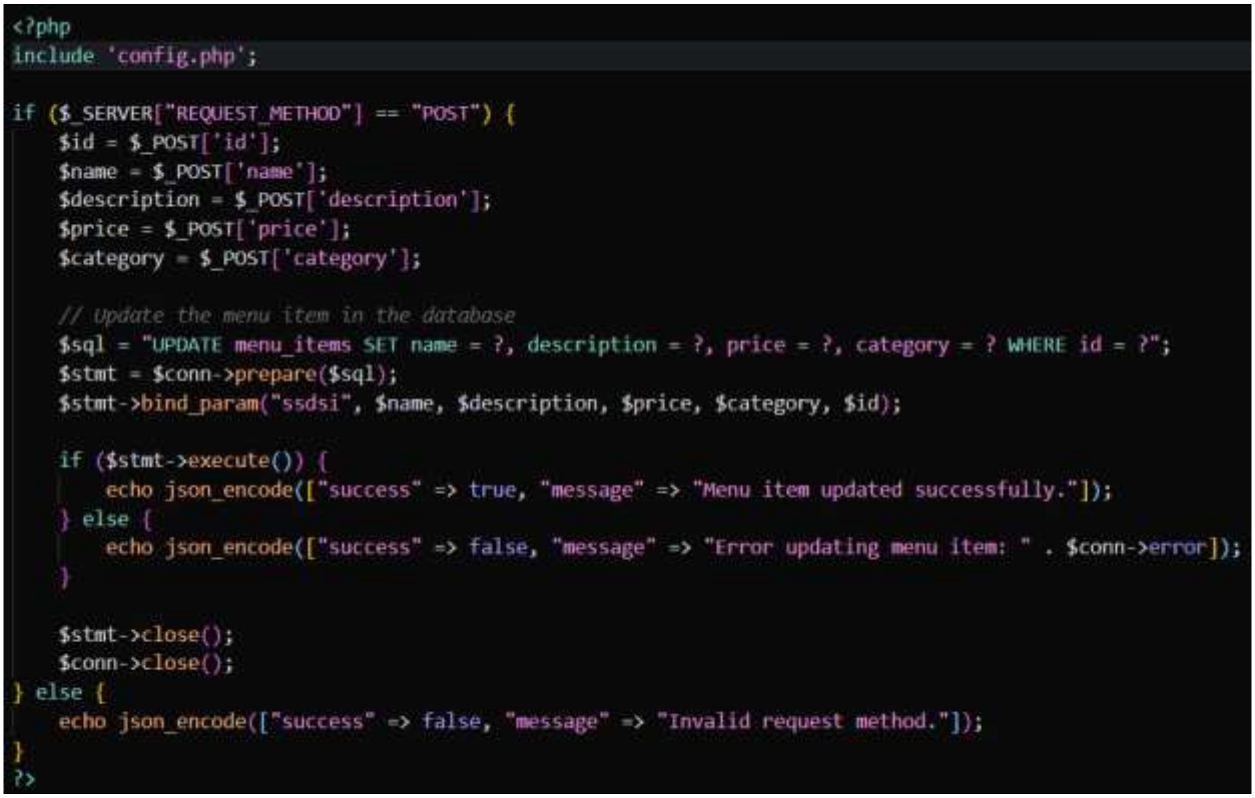
Backend

delete_menu.php -> Deletes a menu item from the database. First checks if menu item exists in any orders to maintain data integrity. If not, proceed with deletion.



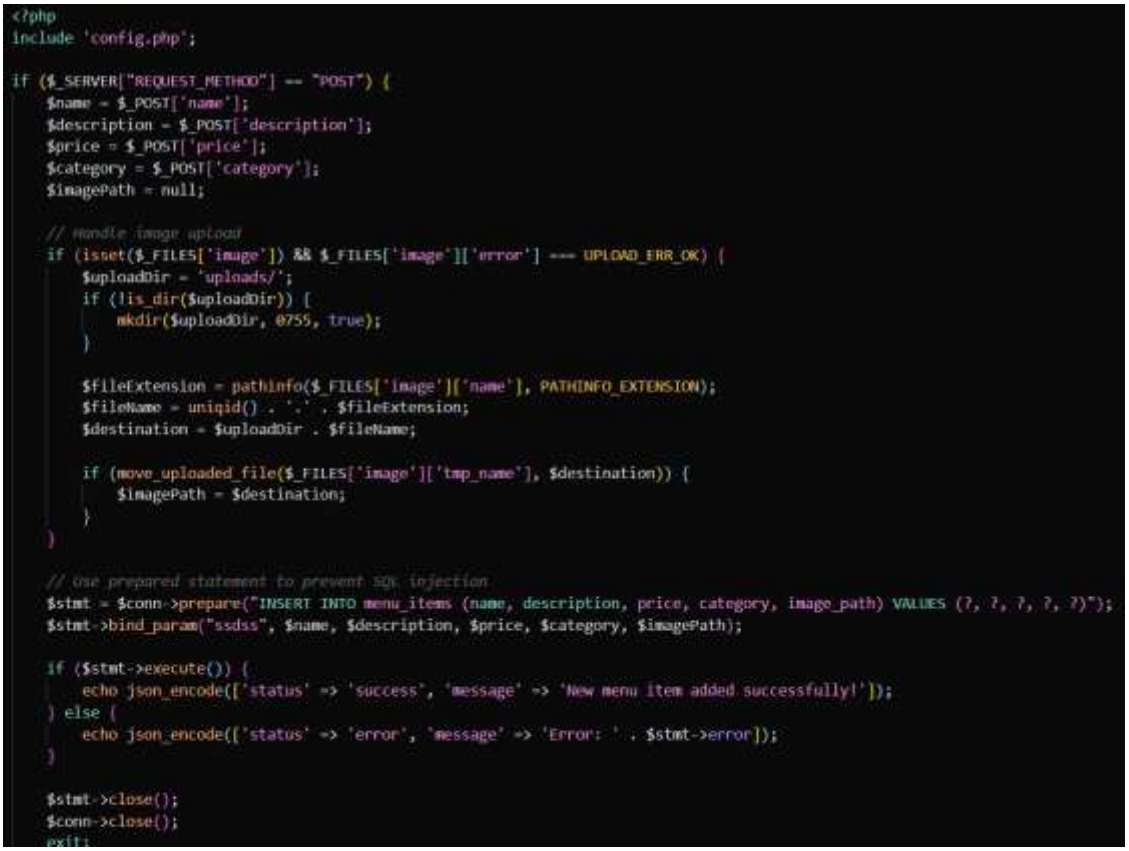
```php
<?php
include 'config.php';

if ($_SERVER['REQUEST_METHOD'] === 'GET' && isset($_GET['id'])) {
    $id = $_GET['id'];

    // First, check if the menu item exists in any orders
    $checkQuery = "SELECT COUNT(*) as count FROM order_items WHERE menu_item_id = ?";
    $stmt = $conn->prepare($checkQuery);
    $stmt->bind_param("i", $id);
    $stmt->execute();
    $result = $stmt->get_result();
    $count = $result->fetch_assoc()['count'];

    if ($count > 0) {
        // If the item exists in orders, return an error
        http_response_code(400);
        echo json_encode(['error' => 'Cannot delete menu item as it exists in orders']);
        exit;
    }

    // If the item is not in any orders, proceed with deletion
    $deleteQuery = "DELETE FROM menu_items WHERE id = ?";
    $stmt = $conn->prepare($deleteQuery);
    $stmt->bind_param("i", $id);

    if ($stmt->execute()) {
        header('Content-Type: application/json');
        echo json_encode(['success' => true, 'message' => 'Menu item deleted successfully']);
    } else {
```

edit.php -> Edits a menu item in the database. After making changes in edit form, it updates the selected menu item with the modified changes.



```php
<?php
include 'config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    $name = $_POST['name'];
    $description = $_POST['description'];
    $price = $_POST['price'];
    $category = $_POST['category'];

    // Update the menu item in the database
    $sql = "UPDATE menu_items SET name = ?, description = ?, price = ?, category = ? WHERE id = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ssdsi", $name, $description, $price, $category, $id);

    if ($stmt->execute()) {
        echo json_encode(["success" => true, "message" => "Menu item updated successfully."]);
    } else {
        echo json_encode(["success" => false, "message" => "Error updating menu item: " . $conn->error]);
    }

    $stmt->close();
    $conn->close();
} else {
    echo json_encode(["success" => false, "message" => "Invalid request method."]);
}
?>
```

store.php -> After adding a menu item, this code executes. It handles storing of the menu item as well as image handling.



```php
<?php
include 'config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    $description = $_POST['description'];
    $price = $_POST['price'];
    $category = $_POST['category'];
    $imagePath = null;

    // Handle image upload
    if (isset($_FILES['image']) && $_FILES['image']['error'] === UPLOAD_ERR_OK) {
        $uploadDir = 'uploads/';
        if (!is_dir($uploadDir)) {
            mkdir($uploadDir, 0755, true);
        }

        $fileExtension = pathinfo($_FILES['image']['name'], PATHINFO_EXTENSION);
        $fileName = uniqid() . '.' . $fileExtension;
        $destination = $uploadDir . $fileName;

        if (move_uploaded_file($_FILES['image']['tmp_name'], $destination)) {
            $imagePath = $destination;
        }
    }

    // Use prepared statement to prevent SQL injection
    $stmt = $conn->prepare("INSERT INTO menu_items (name, description, price, category, image_path) VALUES (?, ?, ?, ?, ?)");
    $stmt->bind_param("ssdss", $name, $description, $price, $category, $imagePath);

    if ($stmt->execute()) {
        echo json_encode(['status' => 'success', 'message' => 'New menu item added successfully!']);
    } else {
        echo json_encode(['status' => 'error', 'message' => 'Error: ' . $stmt->error]);
    }

    $stmt->close();
    $conn->close();
    exit;
```

Testing and Debugging

Test Cases

Feature	Test Case	Expected Output	Result
Add menu item	Enter menu details and picture	Menu item is added to the database	✅ Pass
Editing menu item	Modify menu item details	Menu item is modified and updated	✅ Pass
Deleting menu item	Deletes a menu item	Menu item is deleted from the database	✅ Pass
Viewing order details	Views the order details of the customer	Order details of the customer are properly shown	✅ Pass
Updating order status	Updates the status of the order	Status of order is updated and reflected to the database	✅ Pass
Selecting menu items for order	Customers selecting the food from the menu	Order cart would be filled with the selected food of customer	✅ Pass
Downloading receipt	Clicking download button	Receipt is downloaded by user	✅ Pass

Challenges and Solutions

Issue: Order not processing -> **Solution:** Checked database fields names if matched.

Issue: Order type not being passed to the menu -> **Solution:** Maintain order type status across page navigation

Issue: Sidebar and cart in menu.php not properly collapsing the menu item cards -> **Solution:** Fixed CSS transitions for sidebar and cart

Issue: Receipt not being properly shown -> **Solution:** Proper formatting of all data

Issue: Updating order status results in error -> **Solution:** Fixed implementation of update_order_status.php