UNIVERSITY OF THE PUNJAB

BS (SE) Fall 2022 Morning

Web Engineering

Lab 06

Time Duration: 2 hr, Total Marks: 100

Objective:

The purpose of this lab is to develop a food restaurant web application using the **Model-View-Controller (MVC)** structure and a **connected database model**. This lab will provide practical experience in creating an organized and interactive application for managing delivery and dine-in options, user profile information, a cart system, and reservations. You will also apply CSS styling to improve the user interface.

Through this lab, you will learn to:

1. Cart Implementation.
2. Manage sessions for user information and storing cart details.
3. Apply CSS to style the application for a professional look.

Marks Division

- **Database Setup and Model Implementation** – 10 marks
- **Repository implementation** -20 marks
- **Controller Implementation** – 20 marks
- **View Implementation** – 25 marks
- **Session Management** – 10 marks
- **CSS Styling** – 15 marks

Tasks and Implementation Details

1. Database Setup and Model Implementation (10 Marks)

Objective

Set up a database to store data for users, menu items, cart items, and reservations. Implement models to represent these entities.

Marks Breakdown

- **User Model**,**Menu, Cart, and Reservation Models**: 10 Marks

Implementation

- **Database**: Use SQL Server or any other relational database.
  - **Users Table**: Columns - user_id, username, password, billing_address, phone_number.
  - **Menu Table**: Columns - item_id, item_name, description, price, availability.
  - **Cart Table**: Columns - cart_id, user_id, item_id, quantity.
  - **Reservations Table**: Columns - reservation_id, user_id, date, time, guest_count.

- **Models**:
  - Define classes for each table in the `Models` folder, with properties corresponding to the table columns.
  - Implement CRUD methods to interact with the connected database, including methods for fetching menu items, adding to cart, creating reservations, and managing user information.

---

2. Controller Implementation (20 Marks)

Objective

Create controllers to handle page requests, manage user input, and process data interactions with the Model.

Marks Breakdown

- **Welcome Controller**: 5 Marks
- **Menu Controller**: 10 Marks
- **User Details and Cart Controllers**: 10 Marks
- **Reservation Controller**: 5 Marks

Implementation

- **WelcomeController**: Handles the display of the Welcome page with options for Delivery or Dine-in Reservation.
- **MenuController**: Manages product listing with a search bar and "Add to Cart" functionality.
  - Methods:
    - **GET: /menu -** Display all menu items.
    - POST: /menu/search **-** Search for items by name.

- **UserDetailsController**: Handles user data (address and phone number) to store in the session if not already saved.
    - Methods:
        - GET: /userdetails - Show form to input/edit user information.
        - POST: /userdetails - Save or update user details.
- **CartController**: Manages items added to the cart, quantity updates, and checkout.
    - Methods:
        - GET: /cart - Display cart items.
        - POST: /cart/add - Add item to the cart.
        - POST: /cart/update - Update quantity or remove items.

- **ReservationController**: Manages dine-in reservation functionality.
    - Methods:
        - GET: /reservation - Display reservation form.
        - POST: /reservation - Submit reservation details.

---

3. View Implementation (25 Marks)

Objective

Create views to display content, capture user input, and manage responses for each application page.

Marks Breakdown

- Welcome Page: 5 Marks
- Menu Page: 5 Marks
- User Details Page: 5 Marks
- Cart Page: 5 Marks
- Reservation and Confirmation Pages: 5 Marks

Implementation

- Views: Each view is an HTML template displaying data passed from the Controller.
    - Welcome Page (`welcome.html`): Displays a greeting and options for Delivery or Dine-in.
    - Menu Page (`menu.html`): Lists menu items, provides a search bar, and “Add to Cart” button for each item.
    - User Details Page (`user_details.html`): Form for inputting or editing billing address and phone number.

- Cart Page (`cart.html`): Shows items added to the cart, with options to update quantities or remove items.
- Reservation Form and Confirmation Pages (`reservation_form.html` and `reservation_confirmation.html`): Collects reservation details and displays a confirmation.

---

4. Session Management (10 Marks)

Objective

Use sessions to store user data and cart items for a consistent experience across pages.

Marks Breakdown

- Session Usage in `MenuController`: 5 Marks
- Session Usage in `UserDetailsController`: 5 Marks'

Implementation

- Sessions: Store user ID, billing address, and phone number to avoid redundant data entry.
  - Check for session data in `MenuController` and `UserDetailsController.`
  - If user details are not stored in the session, redirect to the `UserDetails` page.

5. CSS Styling (15 Marks)

Objective

Implement CSS styling across pages to improve user experience with various selectors, box model properties, and pseudo-classes.

Marks Breakdown

- Basic Layout and Styling with Selectors: 5 Marks
- Box Model (Margins, Borders, Padding): 5 Marks
- Pseudo-Classes and Elements: 5 Marks

Implementation

- CSS Types:
  - External CSS for global styles (e.g., `style.css`).
  - Internal CSS in the `<head>` of each page for page-specific styles.

    - **Inline CSS** for individual elements if needed.
- **Selectors**:
    - **Type, ID, and Class Selectors**: Apply to specific page sections, such as `#menu-list` or `.product-card`.
- **Box Model**:
    - Define borders, margins, and padding for form inputs, buttons, and product cards to enhance the layout.
- **Pseudo-Classes**:
    - for interactive buttons.
    - for form fields when active.
- **Pseudo-Elements**:
    - Style the **first letter** of confirmation messages with `::first-letter`.

Important Instructions Before You Start

1. **Read Each Task Carefully**: Understand all requirements before implementation.
2. **Set Up Your Environment**: Ensure tools like Visual Studio, SQL Server, and .NET SDK are installed and configured.
3. **Backup Regularly**: Save progress frequently to avoid data loss.
4. **Test Frequently**: Validate functionality after each step to identify and fix issues early.

Sample "Menu" Table Data

INSERT INTO Menu (MenuId, Category, ItemName, Description, Price, ImageURL)
VALUES

(1, 'Appetizers', 'Garlic Bread', 'Toasted bread with garlic and herbs', 3.99,
'/images/garlic_bread.jpg'),

(2, 'Appetizers', 'Mozzarella Sticks', 'Breaded cheese sticks with marinara sauce', 4.99, '/images/mozzarella_sticks.jpg'),

(3, 'Main Course', 'Margherita Pizza', 'Classic pizza with tomato, basil, and mozzarella', 10.99, '/images/margherita_pizza.jpg'),

(4, 'Main Course', 'Spaghetti Carbonara', 'Pasta with creamy sauce, bacon, and parmesan', 12.99, '/images/spaghetti_carbonara.jpg'),

(5, 'Main Course', 'Grilled Chicken Sandwich', 'Grilled chicken with lettuce and tomato', 8.99, '/images/grilled_chicken.jpg'),

(6, 'Desserts', 'Chocolate Lava Cake', 'Warm cake with a molten chocolate center', 5.99, '/images/chocolate_lava_cake.jpg'),

(7, 'Desserts', 'Vanilla Ice Cream', 'Creamy vanilla ice cream', 3.50, '/images/vanilla_ice_cream.jpg'),

(8, 'Beverages', 'Fresh Lemonade', 'Freshly squeezed lemonade', 2.99, '/images/fresh_lemonade.jpg'),

(9, 'Beverages', 'Cappuccino', 'Italian coffee with frothy milk', 3.50, '/images/cappuccino.jpg'),

(10, 'Beverages', 'Iced Tea', 'Cold brewed tea with lemon flavor', 2.50, '/images/iced_tea.jpg');