UNIVERSITY OF THE PUNJAB

BS (SE) Fall 2022 Morning

Web Engineering - Lab 09

Time Duration: 2 hr, Total Marks: 85

Objective:

Develop a professional and responsive Pharmacy Management System with Create, Read, and Search functionality. Use Bootstrap for the frontend and ASP.NET MVC for the backend, incorporating Partial Views, Dependency Injection (DI), and Sessions for handling data.

Concepts Used:

1. **ASP.NET MVC**: Controllers, Views, and Models for separation of concerns.
2. **Dependency Injection (DI)**: Injecting services like IPharmacyService.
3. **Sessions**: Managing pharmacy items in session.
4. **Partial Views**: Reusable components for displaying items and search results.
5. **Bootstrap**: Responsive frontend with forms, cards, navbar, and footer.
6. **Search Functionality**: Filtering items with partial views.

Steps to Complete the Task:

1. Database and Entity Setup

- **Create PharmacyItem Entity** with the following properties:
  - **Id:** Primary Key, auto-generated.
  - **Name:** Name of the medicine.
  - **Description:** Brief description of the medicine.
  - **Category:** Medicine category (e.g., Painkillers, Antibiotics).
  - **Price:** Price of the medicine.
  - **Quantity:** Available stock of the medicine.
  - **Image:** Path or URL of the medicine image.
- **Use Sessions** to store the list of items instead of directly querying SQL.

2. Backend (MVC and DI)

Step 2.1: Create an Interface and Service

- Define the interface **IPharmacyService** with the following methods:
  - **List<PharmacyItem> GetAllItems():** Retrieve all pharmacy items.
  - **PharmacyItem GetItemById(int id):** Retrieve a pharmacy item by its ID.
  - **void AddItem(PharmacyItem item):** Add a new pharmacy item to the session.
- Implement **IPharmacyService** in **PharmacyService** class and use session to manage data.

Step 2.2: Configure Dependency Injection (DI)

- In **Startup.cs** or **Program.cs**, register IPharmacyService and its implementation **PharmacyService** for Dependency Injection.

Step 2.3: Implement Core Functionality in the Controller

- Create a **PharmacyController** with the following actions:
  - **Index:** Display all items stored in session.
  - **Create:** Render a form for adding a new item.
  - **Details:** Display detailed information for a specific pharmacy item.

Step 2.4: Search Functionality

- Add a search bar in the Index view to filter items based on their name or category.
- Implement the search logic in the controller and update the view using **Partial Views** to dynamically display search results.

3. Frontend Design (Bootstrap)

Step 3.1: Responsive Layout

- Use Bootstrap's grid system for responsive design.
- Display pharmacy items in a card layout with:
  - Name, Category, Price, Quantity, and Image.

Step 3.2: Navbar and Footer

- **Navbar**:
  - Include links: Home, Manage Pharmacy, Categories, About, Contact.
- **Footer**:
  - Add sections like About, Quick Links, and Contact Info.

Step 3.3: Pages

- **Index Page**: Display a grid of all items with search functionality.
- **Create Form**:
  - Use Bootstrap-styled forms for input fields.
  - Include an image upload field.
- **Details Page**: Show detailed information for a pharmacy item using a visually appealing card layout.

4. Partial Views

Step 4.1: Partial Views for Individual Items

- Create a Partial View for displaying individual pharmacy items in the grid layout.

Step 4.2: Partial View for Search Results

- Create another Partial View to update the search results dynamically based on user input.

Marks Distribution (Total: 85)

1. **Database and Entity Setup (10 Marks)**
   - PharmacyItem Entity Creation (5 marks): Correct entity properties (Id, Name, etc.).
   - Sessions for Storing Items (5 marks): Correct use of sessions to manage data.
2. **Backend Development (30 Marks)**
   - IPharmacyService Interface (5 marks): Correct methods definition.
   - PharmacyService Implementation (10 marks): Service logic for session-based data management.
   - Dependency Injection (DI) Setup (5 marks): Proper DI registration in Startup.cs/Program.cs.
   - PharmacyController Implementation (10 marks): Correct MVC actions for CRUD operations.
3. **Search Functionality (10 Marks)**
   - Search Bar in the Index View (5 marks): Implementing search functionality.
   - Search Logic in Controller (5 marks): Correct filtering logic based on search input.
4. **Frontend Design (Bootstrap) (25 Marks)**
   - Responsive Layout (10 marks): Correct use of Bootstrap grid for layout and item display.
   - Navbar and Footer (5 marks): Implementing responsive navbar and footer.
   - Pages (Index, Create, and Details) (10 marks): Proper Bootstrap-styled pages.
5. **Partial Views (10 Marks)**
   - Partial View for Individual Items (5 marks): Correct partial view for displaying items.
   - Partial View for Search Results (5 marks): Correct partial view for dynamically updating search results.

Important Instructions Before You Start:

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