UNIVERSITY OF THE PUNJAB

BS (SE) Fall 2022 Morning

Web Engineering - Lab 05

Time Duration: 2.5 hr, Total Marks: 60

Marks Division

- Database Setup in SQL Server: 5 Marks
- User Model Creation: 5 Marks
- User Repository Implementation: 10 Marks
- Account Controller: 10 Marks
- Login and Register Views: 10 Marks
- Cookie-Based Session Management: 10 Marks
- User Information Management (CRUD Operations): 10 Marks

Task: Implement Database Integration for User Login and Registration

---

1. Database Setup in SQL Server (5 Marks)

- **Create Users Table**: Define columns with the following schema:
  - **Id**: Primary Key, INT, auto-increment.
  - **Name**: VARCHAR(100), not null.
  - **Email**: VARCHAR(100), unique, not null.
  - **Password**: VARCHAR(100), not null.
  - **CreatedDate**: DATETIME, default to current date.

2. User Model (5 Marks)

- **Create User Class**: In the Models folder, create a User class with properties:
  - Id
  - Name
  - Email
  - Password
  - CreatedDate
    Ensure that the properties match the Users table structure.

3. User Repository (10 Marks)

- **Create UserRepository**: In the Repositories folder, implement the following methods:
  - GetUserByEmailAndPassword(string email, string password): Retrieves a user by email and password.
  - IsEmailExists(string email): Checks if the email already exists in the database.
  - **CRUD Operations**:
    - AddUser(User user): Adds a new user to the database.
    - GetUserById(int id): Retrieve user details by Id.

        - UpdateUser(User user): Update user information.
        - DeleteUser(int id): Delete user by Id.

4. Account Controller (10 Marks)

- **Create AccountController**: Use UserRepository for user validation and interactions. Implement the following actions:

  Login Actions:

    - GET Login(): Renders the login view.
    - POST Login(string email, string password): Checks if the provided email exists in the database:
        - If registered, validate the password and log in the user.
        - If not registered, redirect to the Register view with a ViewBag message stating that the email does not exist.

  Register Actions:

    - GET Register(): Displays the registration form.
    - POST Register(User user): Manages new user registrations:
        - Ensures the email is unique before proceeding.
        - Adds the new user's details to the database.
        - Displays registration success or failure feedback through ViewBag.

  CRUD Operations:

    - GET UserDetails(int id): Display user details.
    - GET EditUser(int id): Show the edit form for a specific user.
    - POST EditUser(User user): Update user information in the database.
    - POST DeleteUser(int id): Delete user from the database.

5. Feedback Management

- **Use ViewBag**: Display messages for login and registration, including "Invalid credentials" and "User registered successfully." Display messages for update success or failure.

6. Create Login and Register Views (10 Marks)

- **Views/Account/Login.cshtml**:
    - Create form fields for Email and Password, along with a login button.
    - Display login feedback using ViewBag.
- **Views/Account/Register.cshtml**:
    - Include fields for Name, Email, Password, and Confirm Password with client-side validation.
    - Use ViewBag to display outcomes for successful or failed registration.
- **Views/Account/UserDetails.cshtml**:
    - Display user details retrieved from the database.
- **Views/Account/EditUser.cshtml**:
    - Create form fields for editing user information (Name, Email, Password).

    - Use ViewBag to show success or error messages.

7. Cookie-Based Session Management (10 Marks)

- **Set a Login Cookie**: On successful login, store the user's Email or Id in a cookie to manage the login state.
- **Logout Action**: Add a Logout action in AccountController to clear the cookie and redirect to the login view.

8. Testing

- **Run the Application**: Test login and registration by navigating to /Account/Login and /Account/Register.
- **Validation Checks**:
    - Test for valid and invalid login attempts and confirm ViewBag messages display as expected.
    - Verify that duplicate emails are prevented during registration.
    - Confirm that unregistered users attempting to log in are redirected to the Register view.
- **CRUD Operations**:
    - Test reading user information by navigating to /Account/UserDetails/{id}.
    - Test the edit functionality by navigating to /Account/EditUser/{id} and updating user information.
    - Test the delete functionality to ensure users can be removed from the database.
- **Cookie Handling**:
    - Ensure cookies are properly set upon login and removed upon logout, and verify the login state is accurately tracked.

Important Instructions Before You Start

1. **Read Each Task Carefully**: Before beginning any task, ensure you understand what is required. Pay attention to specific instructions, especially regarding file names, locations, and coding practices.
2. **Set Up Your Environment**: Make sure your development environment is properly configured with the necessary tools and software, including:
    - Visual Studio (or another IDE that supports ASP.NET MVC).
    - SQL Server for database management.
    - .NET Framework or .NET Core SDK (depending on your project setup).
3. **Backup Your Work**: Regularly save your progress and back up your code. This can help prevent data loss and allows you to revert to a previous state if needed.
4. **Test Frequently**: After completing each task or section, run your application to test functionality. This will help you identify issues early.