

CST-350 Activity 2 Login and Registration System

Alex M. Frear

College of Science, Engineering, and Technology, Grand Canyon University

Course Number: CST-350

Professor Brandon Bass

10/28/2024

GitHub Link: https://github.com/amfrear/cst350/tree/main/Activity_2

Contents

Part 1: Login and Registration System Overview	3
Screenshots.....	3
Home Page with Login and Register Links	3
Login Page.....	4
Successful Login	5
Login Failure	6
Summary of Key Concepts (Part 1)	7
Part 2: User Registration and Members-Only Access	8
Screenshots.....	8
Members Only Page Access by Existing User (Harry)	8
Register Page	9
New Registered User - Login Success.....	10
Members Only Page Access by New User	11
Summary of Key Concepts (Part 2)	12

Part 1: Login and Registration System Overview

Screenshots

Home Page with Login and Register Links

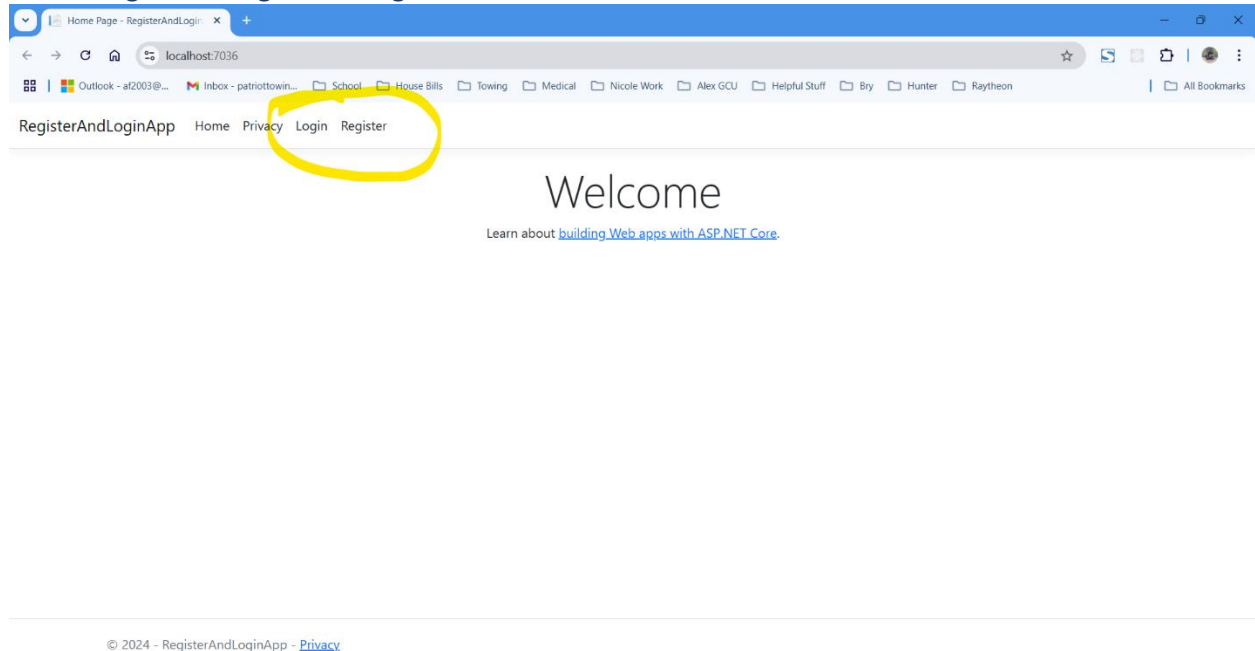


Figure 1 This screenshot shows the home page of the application with Login and Register links in the navigation bar, providing easy access to authentication features.

Login Page

The screenshot shows a web browser window with the following elements:

- Browser Tab:** Login - RegisterAndLoginApp
- Address Bar:** localhost:7036/Login
- Navigation Bar:** RegisterAndLoginApp | Home | Privacy
- Main Content:**
 - Heading:** Login
 - Form:**
 - Username:** Input field
 - Password:** Input field
 - Login:** Blue button
- Footer:** © 2024 - RegisterAndLoginApp - [Privacy](#)

Figure 2 This screenshot displays the login page, prompting the user to enter their Username and Password to access the system.

Successful Login

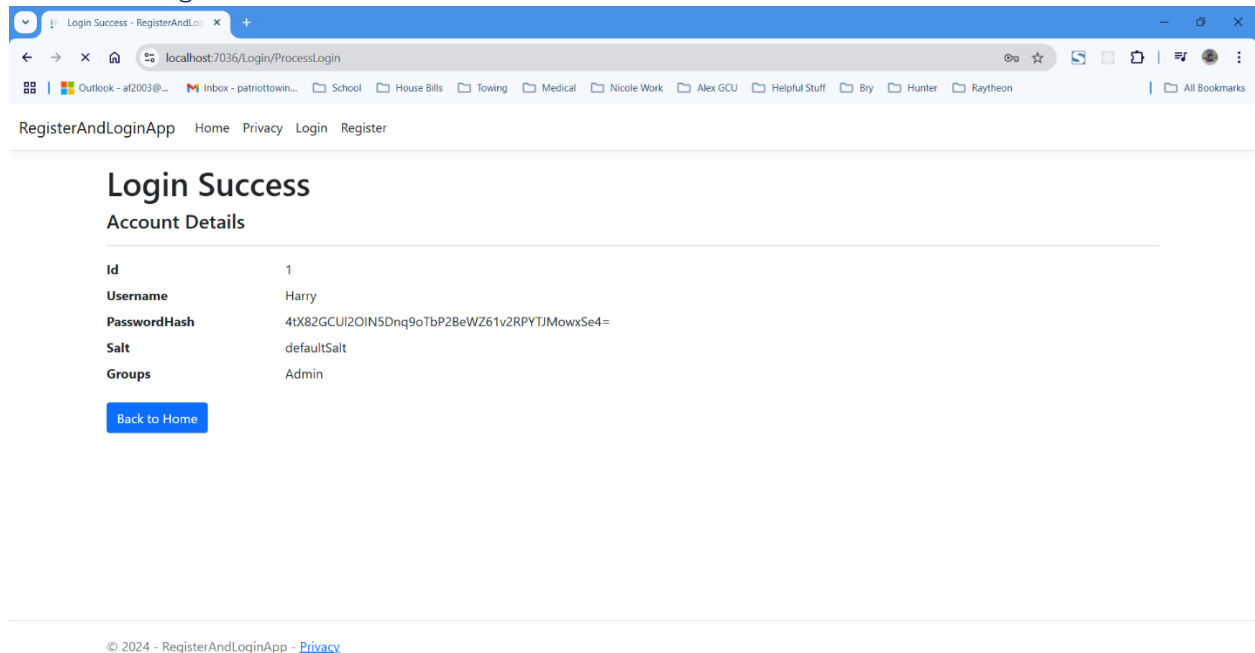


Figure 3 The screenshot shows a successful login attempt, displaying the user's account details, including Username, PasswordHash, Salt, and Groups.

Login Failure

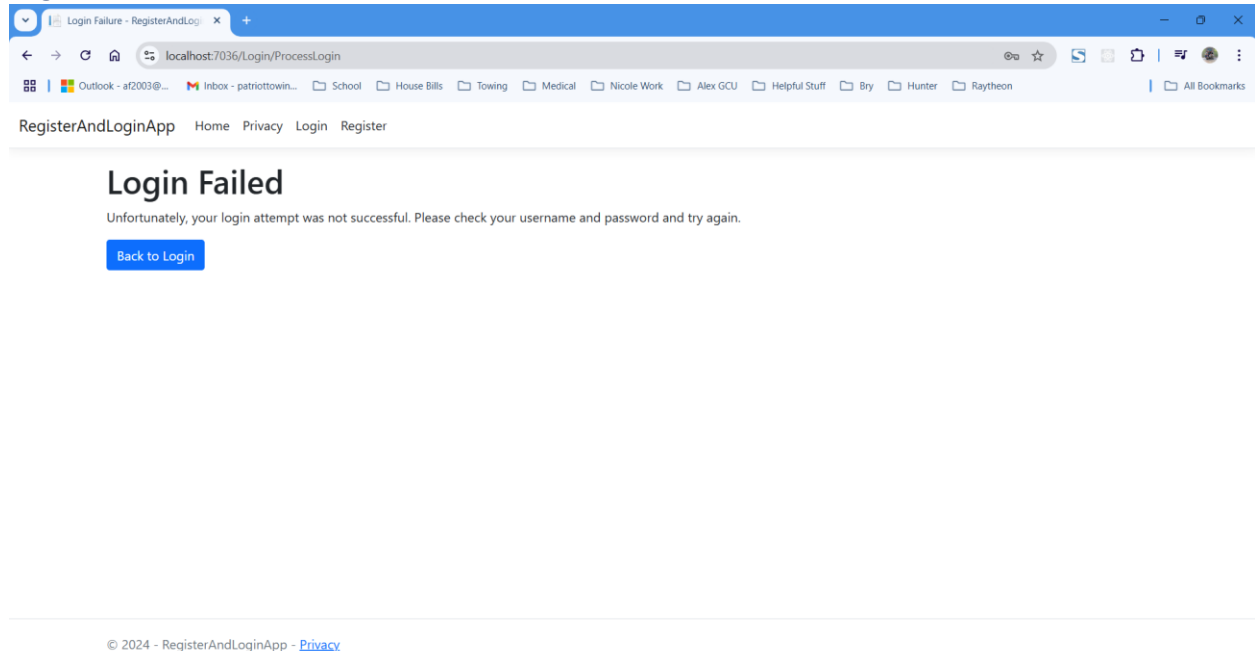


Figure 4 This screenshot shows the login failure page, which appears when the user enters incorrect credentials. The message indicates that the login attempt was not successful and prompts the user to try again.

Summary of Key Concepts (Part 1)

In Part 1 of this activity, I developed a login and registration system in ASP.NET Core using the MVC pattern. Key concepts I covered include setting up the project structure, building user models, creating views for login and registration, implementing controllers, and integrating user management logic. I learned about the importance of hashing and salting passwords, used ASP.NET helper tags for simplifying form creation, and tested my application to ensure proper user authentication functionality.

Part 2: User Registration and Members-Only Access

Screenshots

Members Only Page Access by Existing User (Harry)

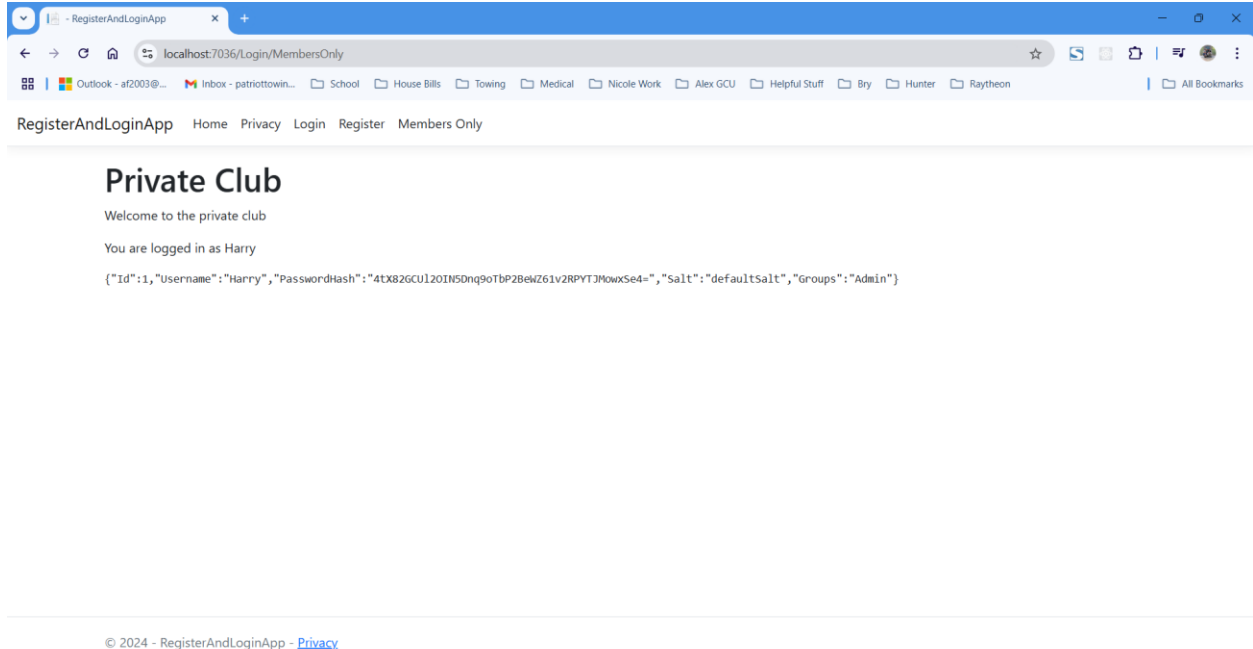


Figure 5 This screenshot shows that the existing user Harry is also able to access the Members-Only page, with session data displayed for verification.

Register Page

Register - RegisterAndLoginApp X

localhost:7036/Login/Register

RegisterAndLoginApp Home Privacy Login Register Members Only Logout

Register

Username

Password

Groups

☐ Admin

☐ Users

☐ Students

Create

[Back to List](#)

© 2024 - RegisterAndLoginApp - [Privacy](#)

Figure 6 This screenshot shows the registration page, where new users can create an account by providing a Username, Password, and selecting one or more Groups to belong to.

New Registered User - Login Success

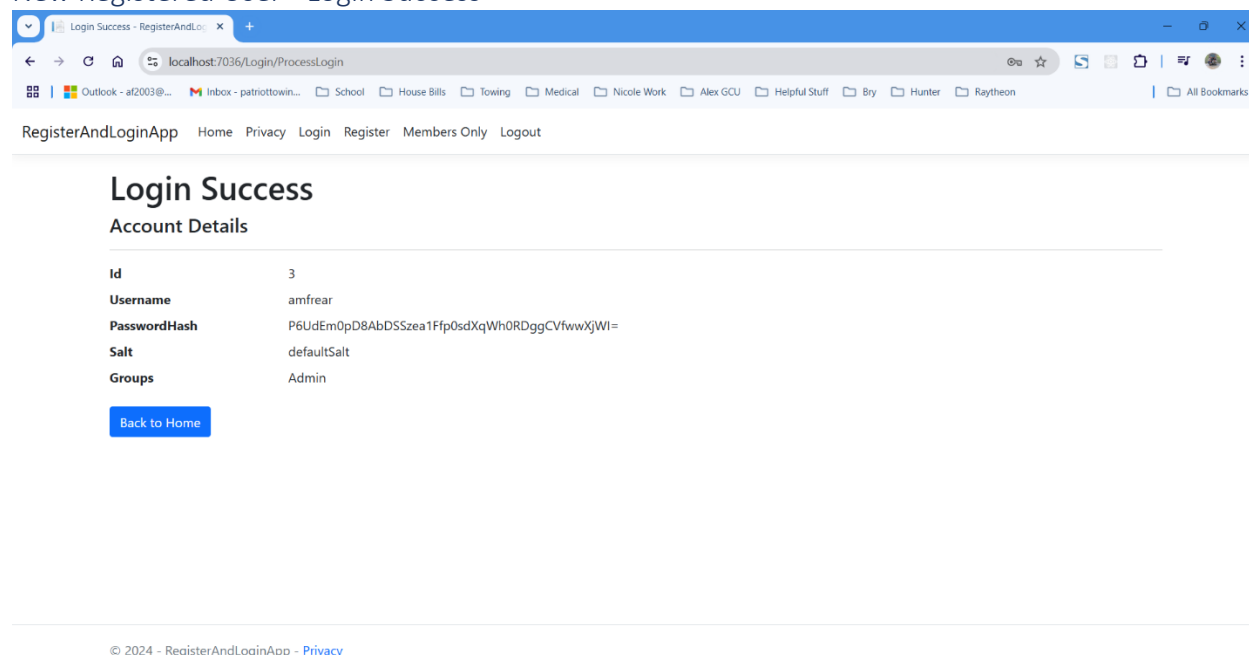


Figure 7 The screenshot shows a successful login by a newly registered user, displaying the account details including Username, PasswordHash, Salt, and Groups.

Members Only Page Access by New User

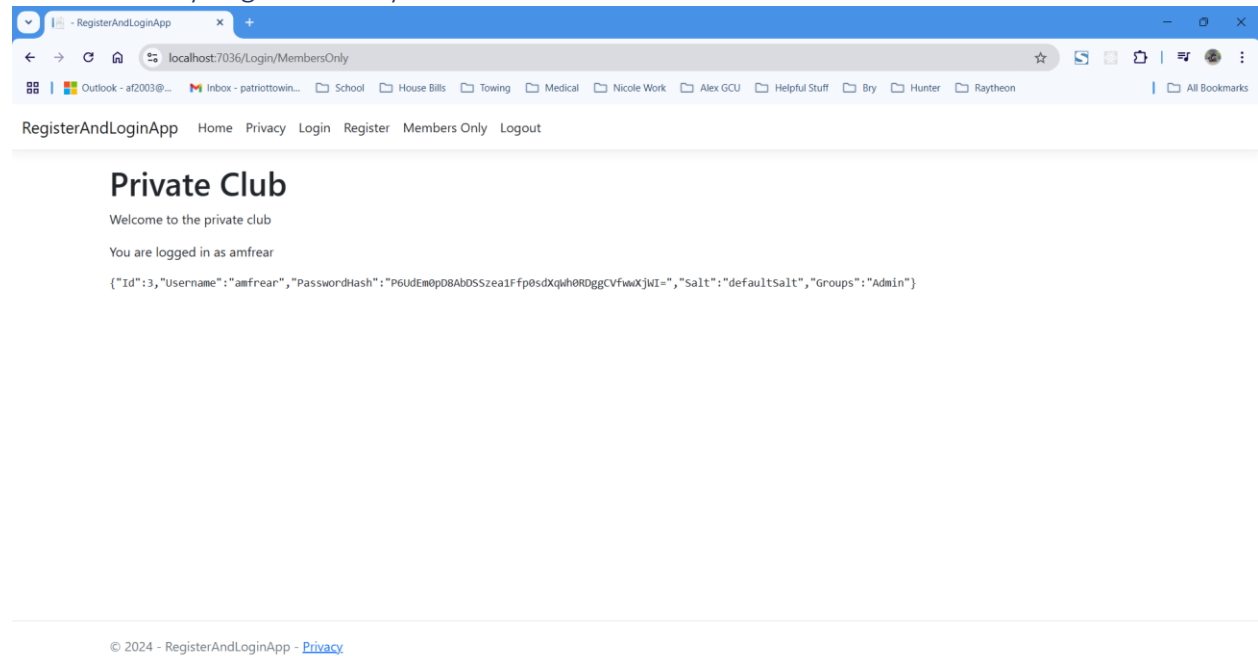


Figure 8 This screenshot demonstrates that the newly registered user can successfully access the Members-Only page, showing the session data for verification.

Summary of Key Concepts (Part 2)

In Part 2 of this activity, I extended the login and registration system to include a user registration feature and restricted access to a Members-Only page. Key concepts I covered include creating a registration form, building a RegisterViewModel to handle user input, and adding new methods in the controller to process registration. I also introduced session management to maintain user state and ensure that only authenticated users can access restricted pages. The Members-Only page verifies session information before granting access, demonstrating effective use of action filters and session variables in ASP.NET Core. By completing Part 2, I learned how to add new users to the system, manage their authentication state, and restrict page access based on login status.