CST-350 Activity 3 Database and Users

Alex M. Frear

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

Course Number: CST-350

Professor Brandon Bass

11/6/2024

GitHub Link: https://github.com/amfrear/cst350/tree/main/Activity_3

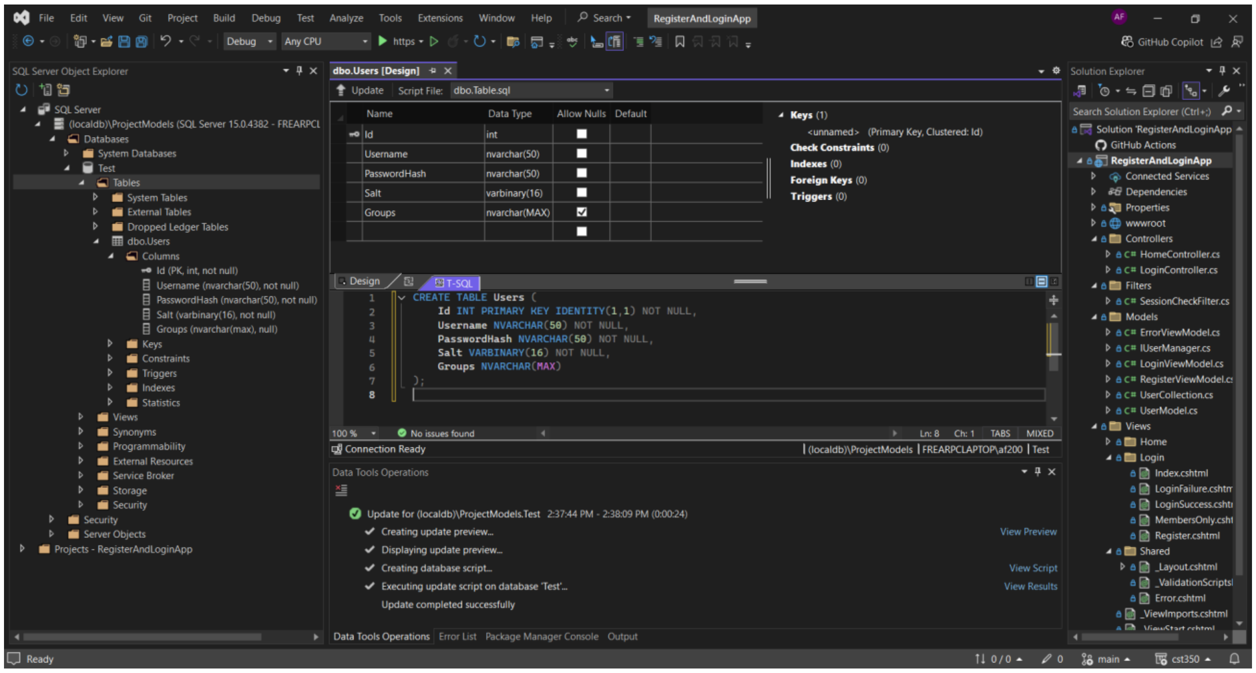
Contents

Part 1: Database Setup and Integration Overview 3
Screenshots 3
Users Table Creation 3
Users Table Data Insertion 4
Users Table Data Validation 5
New Registered User - Login Success 6
New Registered User in Database 7
Members Only Page Access by New User 8
Summary of Key Concepts (Part 1) 9
Part 2: User Group Access 10
Screenshots 10
Admin Only Page 10
User Only Access Restriction 11
Summary of Key Concepts (Part 2) 12

Part 1: Database Setup and Integration Overview

Screenshots

Users Table Creation



Name	Data Type	Allow Nulls	Default
Id	int	☐	
Username	nvarchar(50)	☐	
PasswordHash	nvarchar(50)	☐	
Salt	varbinary(16)	☐	
Groups	nvarchar(MAX)	✓	
		☐	

```
Keys (1)
    <unnamed>   (Primary Key, Clustered: Id)
Check Constraints (0)
Indexes (0)
Foreign Keys (0)
Triggers (0)
```

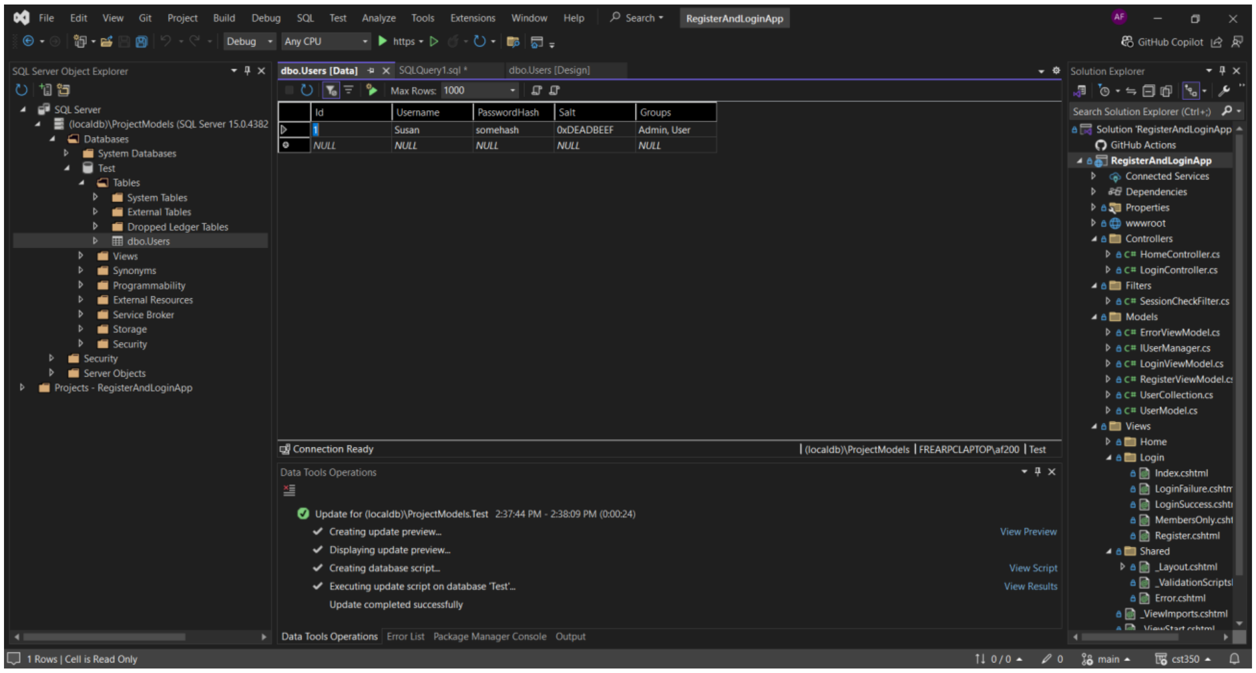
```sql
CREATE TABLE Users (
    Id INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    Username NVARCHAR(50) NOT NULL,
    PasswordHash NVARCHAR(50) NOT NULL,
    Salt VARBINARY(16) NOT NULL,
    Groups NVARCHAR(MAX)
);
```

```
Connection Ready    (localdb)\ProjectModels | FREARPCLAPTOP\af200 | Test

Update for (localdb)\ProjectModels.Test   2:37:44 PM - 2:38:09 PM (0:00:24)
  ✓ Creating update preview...                          View Preview
  ✓ Displaying update preview...
  ✓ Creating database script...                         View Script
  ✓ Executing update script on database 'Test'...       View Results
    Update completed successfully
```

Figure 1 The screenshot shows the Users table created in SQL Server with fields for Id, Username, PasswordHash, Salt, and Groups, as part of the database setup.

Users Table Data Insertion



Id	Username	PasswordHash	Salt	Groups
1	Susan	somehash	0xDEADBEEF	Admin, User
NULL	NULL	NULL	NULL	NULL

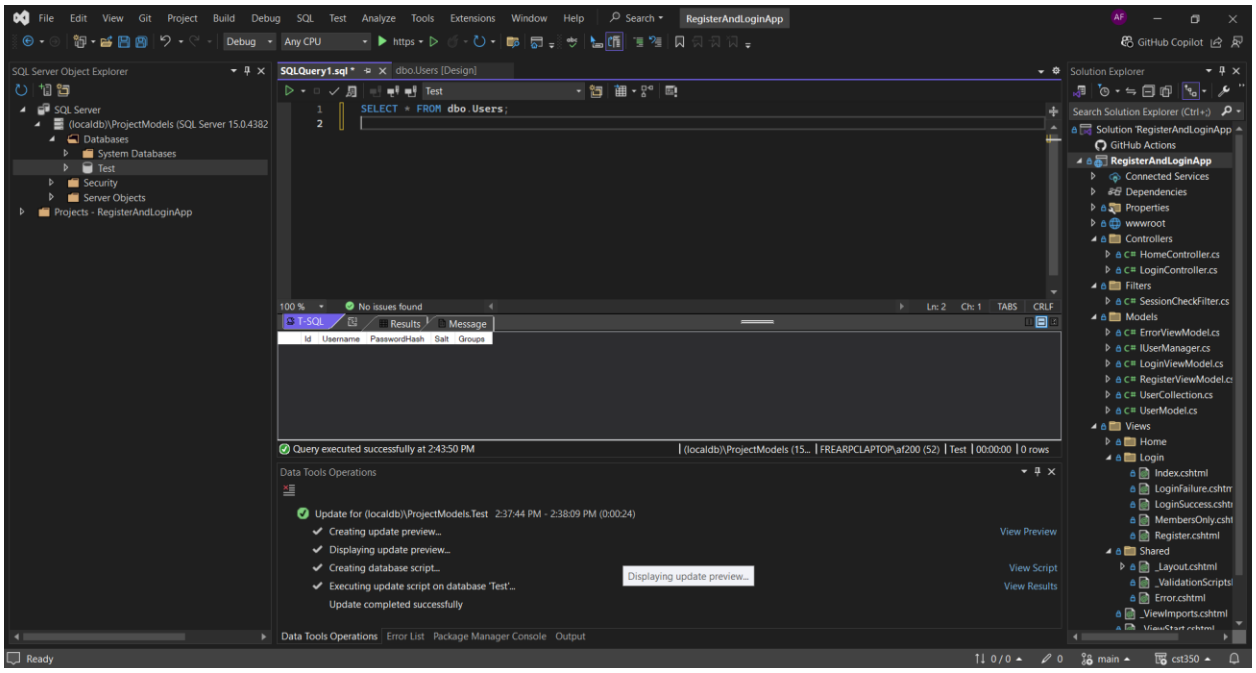
```
Connection Ready                                   (localdb)\ProjectModels | FREARPCLAPTOP\af200 | Test

Update for (localdb)\ProjectModels.Test  2:37:44 PM - 2:38:09 PM (0:00:24)
  ✓ Creating update preview...                                  View Preview
  ✓ Displaying update preview...
  ✓ Creating database script...                                  View Script
  ✓ Executing update script on database 'Test'...                View Results
    Update completed successfully

1 Rows | Cell is Read Only
```

Figure 2 This screenshot displays the insertion of a new user into the Users table, populating fields such as Username, PasswordHash, Salt, and Groups.

Users Table Data Validation



```sql
SELECT * FROM dbo.Users;
```

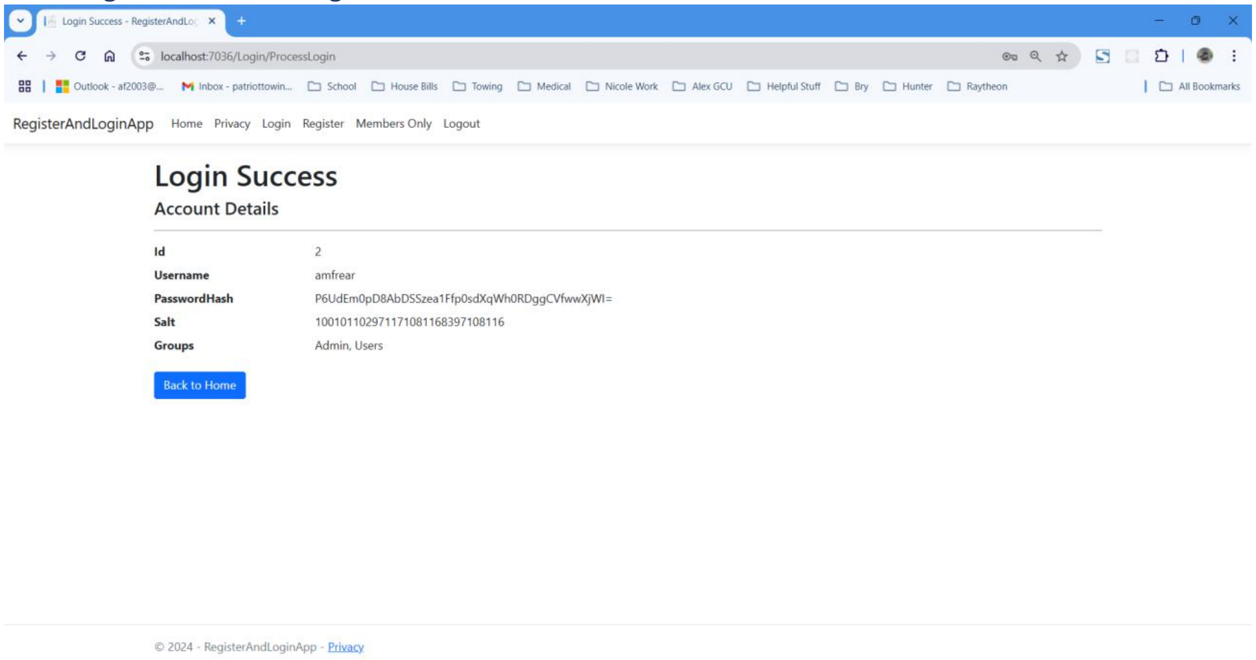
Id	Username	PasswordHash	Salt	Groups

```
Query executed successfully at 2:43:50 PM | (localdb)\ProjectModels (15... | FREARPCLAPTOP\af200 (52) | Test | 00:00:00 | 0 rows
```

```
Update for (localdb)\ProjectModels.Test  2:37:44 PM - 2:38:09 PM (0:00:24)
  ✓ Creating update preview...
  ✓ Displaying update preview...
  ✓ Creating database script...
  ✓ Executing update script on database 'Test'...
    Update completed successfully
```

*Figure 3 Here, we see the validation step where the SELECT * FROM dbo.Users query confirms that the user data is successfully saved in the database.*

New Registered User - Login Success



Login Success

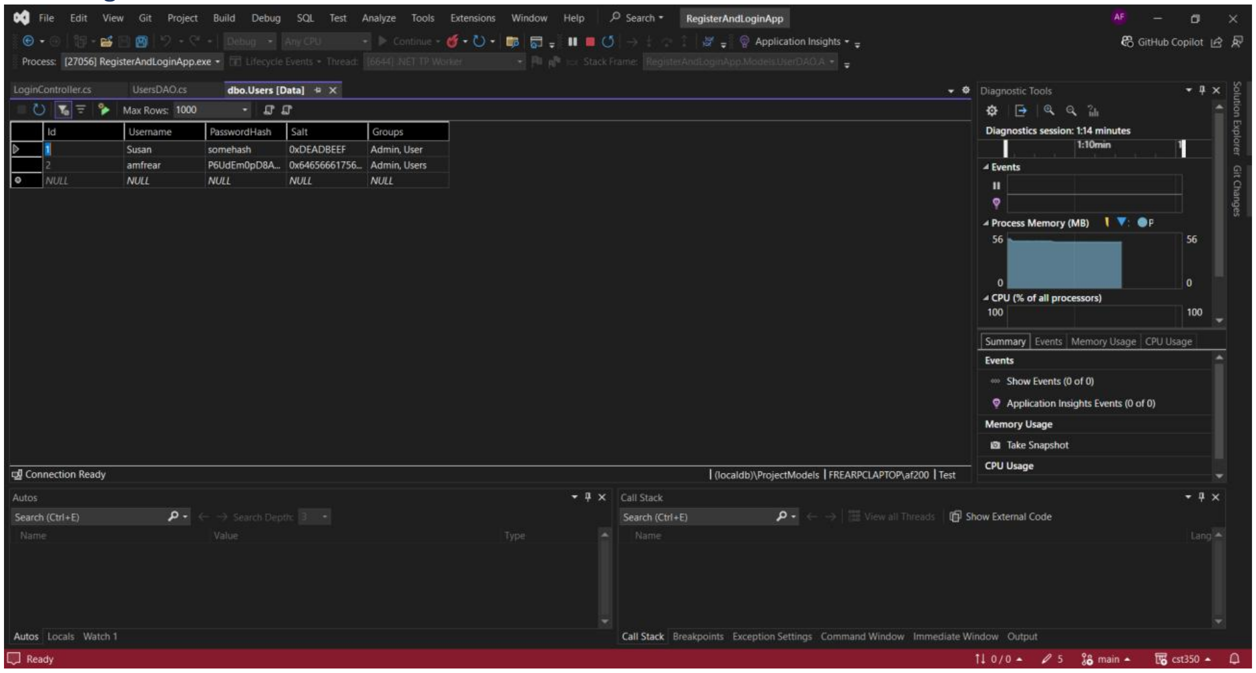
Account Details

Id	2
Username	amfrear
PasswordHash	P6UdEm0pD8AbDSSzea1Ffp0sdXqWh0RDggCVfwwXjWI=
Salt	100101102971171081168397108116
Groups	Admin, Users

© 2024 - RegisterAndLoginApp - Privacy

Figure 4 The screenshot shows a successful login by the newly registered user, displaying account details, including Username, PasswordHash, Salt, and Groups.

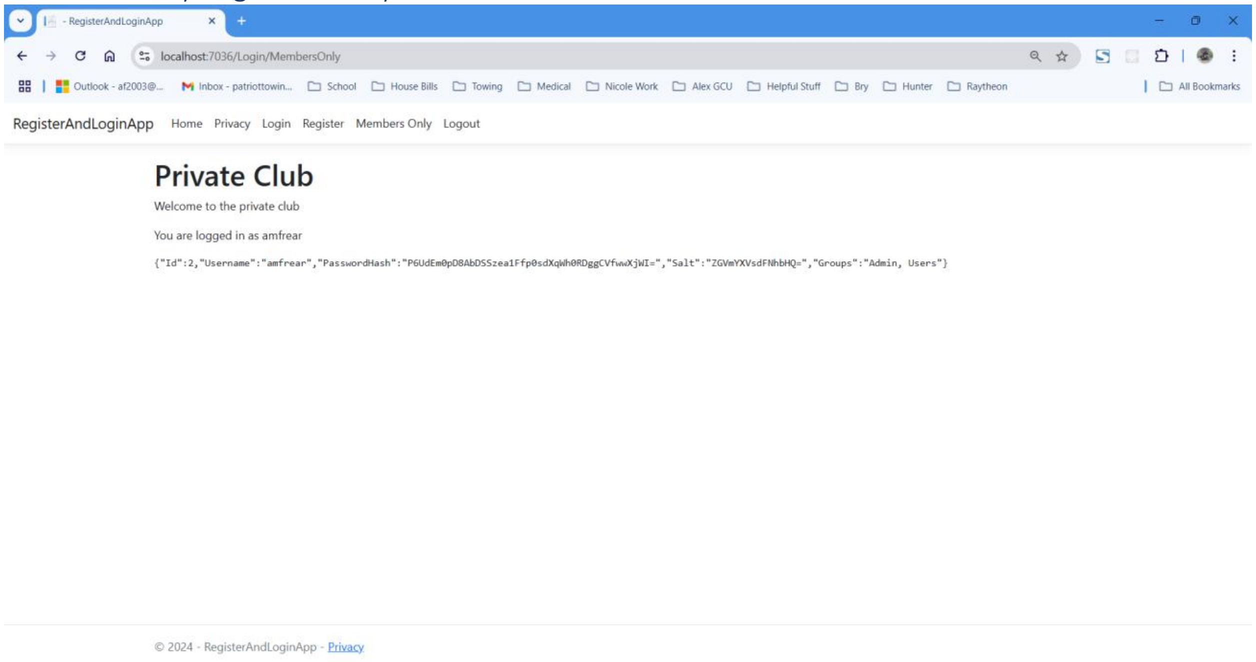
New Registered User in Database



	Id	Username	PasswordHash	Salt	Groups
▷	1	Susan	somehash	0xDEADBEEF	Admin, User
	2	amfrear	P6UdEm0pD8A...	0x64656661756...	Admin, Users
	NULL	NULL	NULL	NULL	NULL

Figure 5 The new user entry is verified within the database, showing that the registration functionality successfully added the user details to the Users table.

Members Only Page Access by New User



RegisterAndLoginApp Home Privacy Login Register Members Only Logout

Private Club

Welcome to the private club

You are logged in as amfrear

{"Id":2,"Username":"amfrear","PasswordHash":"P6UdEm0pD8AbDSSzea1Ffp0sdXqWh0RDggCVfwwXjWI=","Salt":"ZGVmYXVsdFNhbHQ=","Groups":"Admin, Users"}

© 2024 - RegisterAndLoginApp - Privacy

Figure 6 This screenshot demonstrates that the newly registered user has successfully accessed the Members-Only page, confirming the session validation and restricted access.

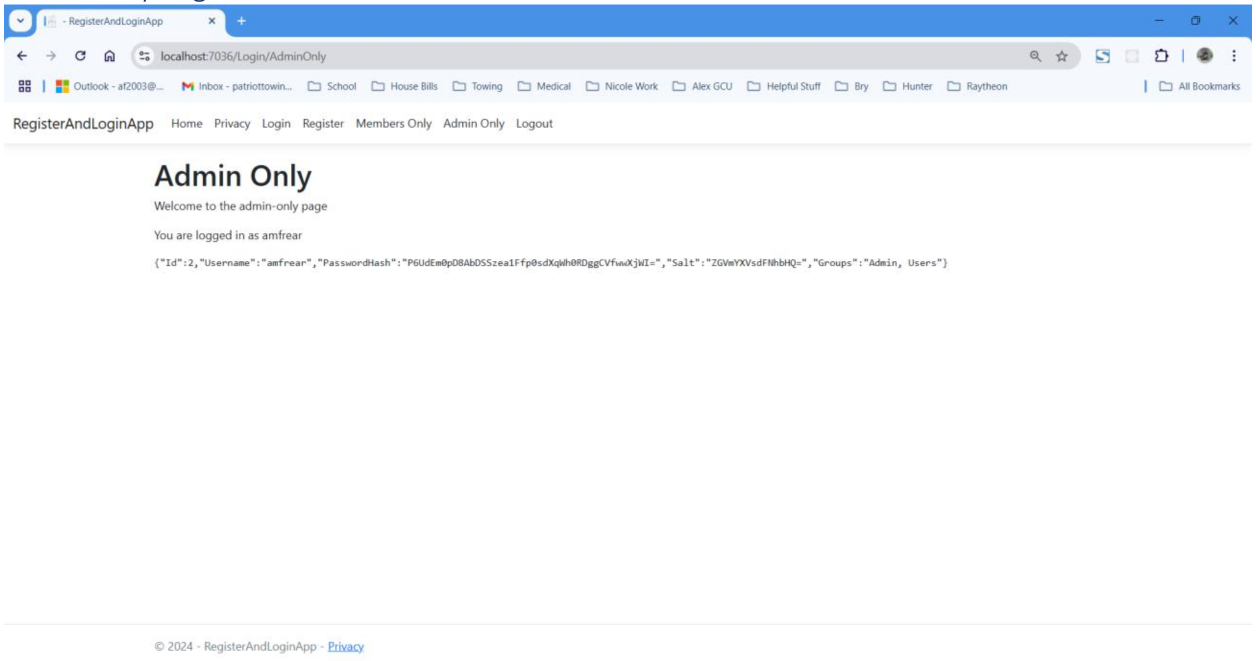
Summary of Key Concepts (Part 1)

In Part 1 of this activity, I integrated the login and registration system with a SQL Server database. This involved setting up a Users table in SQL Server to securely store user data, implementing SQL commands in C# for data insertion and validation, and developing a registration and login system with password hashing and salting to enhance security. Additionally, I created a Members-Only page that uses session-based access control to restrict access to authenticated users only. Through these tasks, I gained practical experience in managing user data within a database and enforcing authentication requirements for protected pages.

Part 2: User Group Access

Screenshots

Admin Only Page



Admin Only

Welcome to the admin-only page

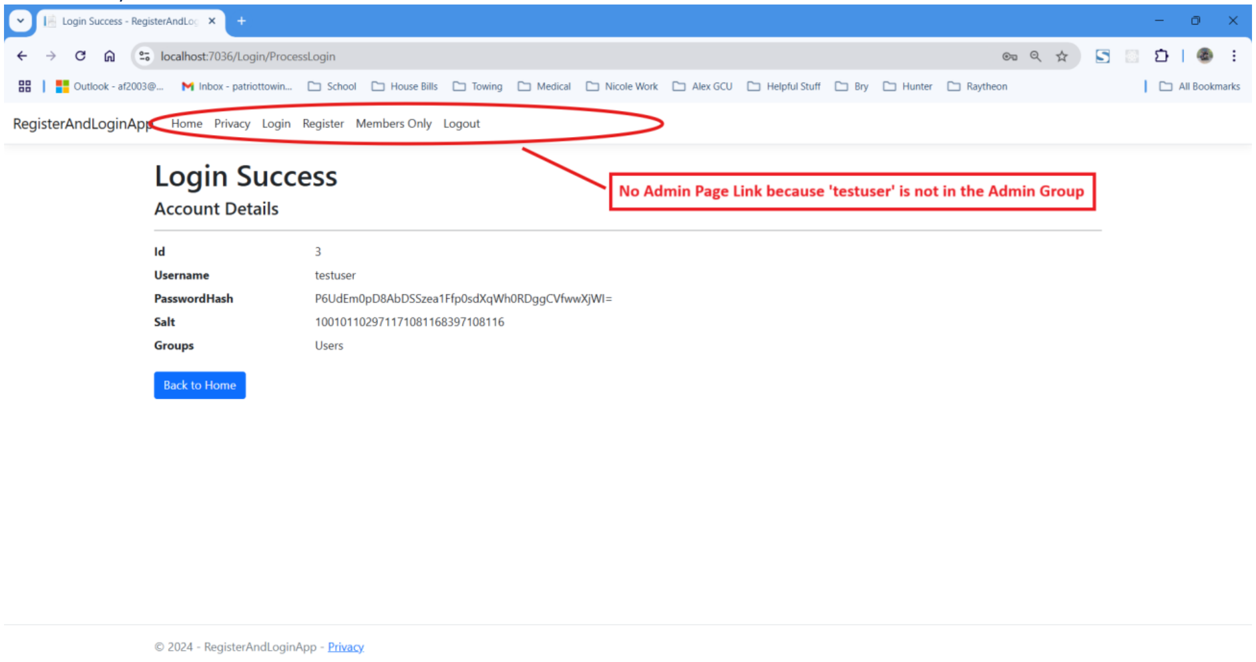
You are logged in as amfrear

```
{"Id":2,"Username":"amfrear","PasswordHash":"P6UdEm0pD8AbDSSzea1Ffp0sdXqWh0RDggCVfwwXjWI=","Salt":"ZGVmYXVsdFNhbHQ=","Groups":"Admin, Users"}
```

© 2024 - RegisterAndLoginApp - Privacy

Figure 7 The screenshot shows the Admin-Only page, which is accessible only to users who belong to the Admin group, demonstrating group-based access control.

User Only Access Restriction


Login Success - RegisterAndLo
localhost:7036/Login/ProcessLogin
RegisterAndLoginApp
Home Privacy Login Register Members Only Logout
No Admin Page Link because 'testuser' is not in the Admin Group
Back to Home

Login Success

Account Details

Id	3
Username	testuser
PasswordHash	P6UdEm0pD8AbDSSzea1Ffp0sdXqWh0RDggCVfwwXjWI=
Salt	100101102971171081168397108116
Groups	Users

© 2024 - RegisterAndLoginApp - Privacy

Figure 8 This screenshot demonstrates that a regular user, who is not part of the Admin group, is restricted from accessing the Admin-Only page, ensuring role-based access.

Summary of Key Concepts (Part 2)

In Part 2 of this activity, I implemented role-based access control by creating an Admin-Only page that is restricted to users in the Admin group. I developed a custom action filter, AdminCheckFilter, which verifies the user's role by checking session data for group membership. This filter redirects non-admin users attempting to access the Admin-Only page back to the login page. Through this process, I learned how to use custom filters to enforce group-specific access restrictions in ASP.NET Core MVC, enhancing both the security and structure of my application.