

ECS-162 :: Final Project

Hassan Ali

06/05/2024

Introduction

ShareStuff is an idea sharing website that allows users to share their thoughts with others. The website allows user authentication through Google and lets users create and read posts from others, comment on posts and delete their own post.

Two Features

The 2 features I implemented are: being able to comment on other posts and being able to update comments. I initially wanted to be able to update posts as well, but there were too many elements to a post and I was unable to update it completely. The main concern was that the code is now very changed and it would take a much longer time to get each element in place, make sure all the data is still there, make sure the order of the post is not impacted, make sure people can view updated content (this is because I don't want people making any changes to spread misinformation. Instead of implementing updates on the post, I was able to implement it on the comments. The comment history is sorted from recent to oldest so people can see how a comment was edited. The comments also don't allow the name of the commenter to be edited, this means a user cannot edit their comment and put someone else's name to imitate another user.

Data Persistence, Environment Variables and Authentication

I was able to implement all 3 of them. There are 4 tables in total: users, posts, comments and likes. The information is persistent and does not go away when the server shuts down. All sensitive items have been moved to the .env file to ensure safety. Google authentication allows a secure way to register and log in. The google ID is hashed and saved into the database.

1 Authentication

Authentication is fully functional. Log in, log out, and register all work as expected. Registering requires a unique case-sensitive username. After entering the username, the users are prompted by Google to login, once login is complete, the users can start posting!

1.1 Screenshots

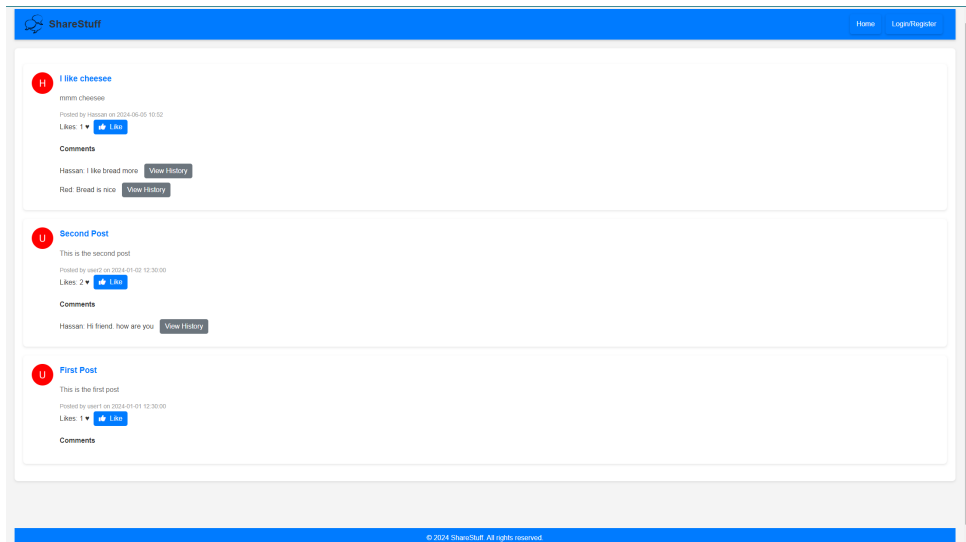


Figure 1: Screenshot of logged out view

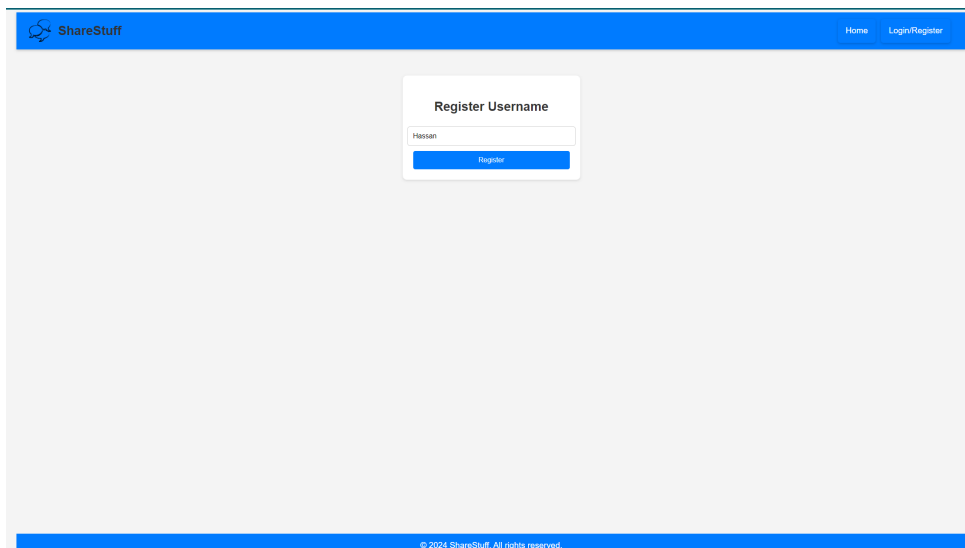


Figure 2: Screenshot of registration/login view

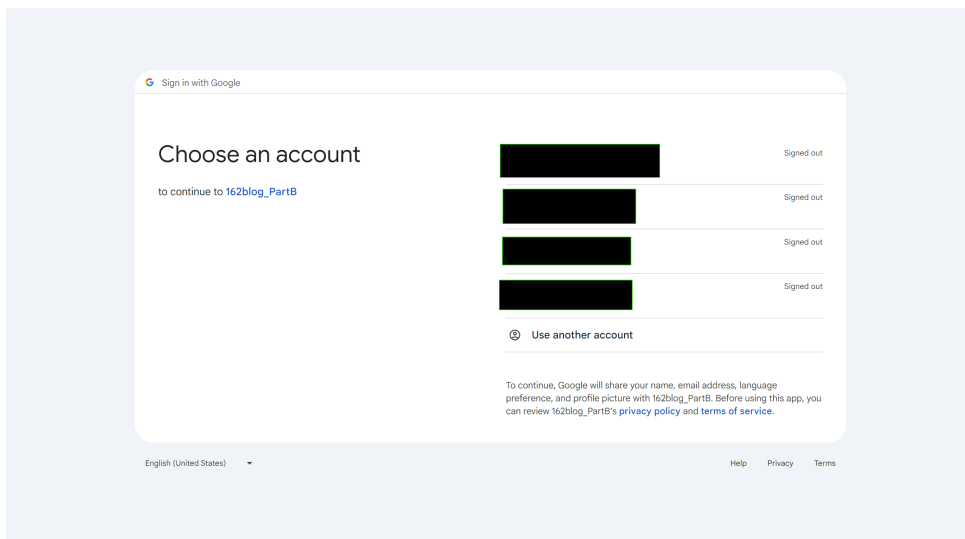


Figure 3: Screenshot of after entering username

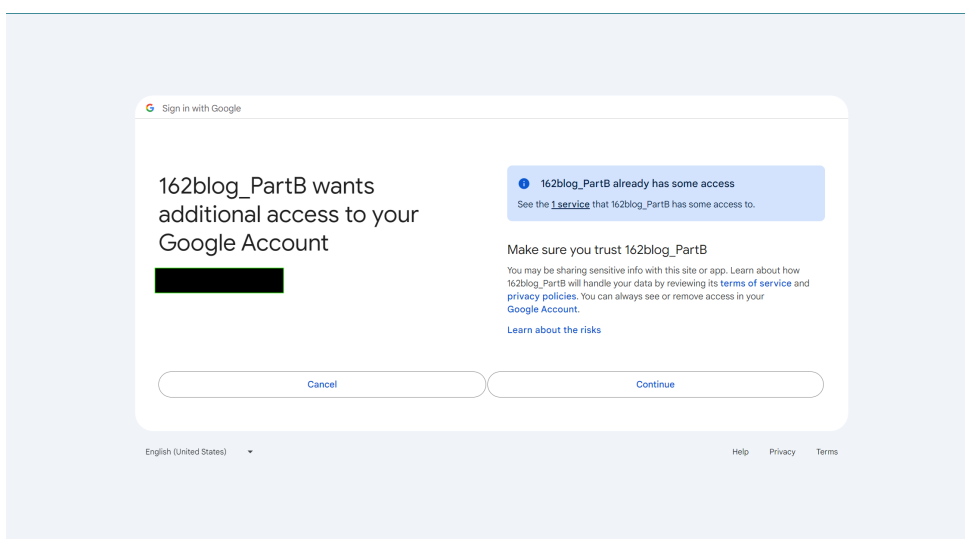


Figure 4: Screenshot of after entering password



Figure 5: Screenshot of Terms and Conditions



Figure 6: Screenshot of Privacy Policy

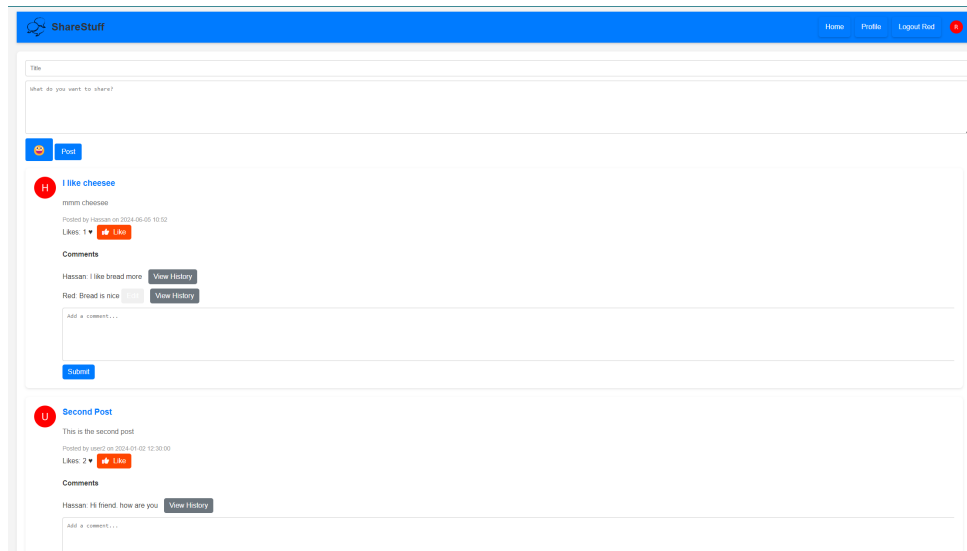


Figure 7: Screenshot of logged in view

2 Post management

Post management is fully functional. All post management operations require the user to be logged in. Each post contains the poster name, posted date, and like info. Users will not see the comment box when they are not logged in. Users can create new posts, like other people's posts, delete their own posts, comment on posts, and view posts from other users. The like button remains red even after we change pages which is something I was unable to do in part A. Users are not allowed to like their own posts and are only allowed to like a post once, clicking like again will remove the like. I created another Table for likes and comments. The like table allows us to implement functionality such as liking a post only once and making sure users cannot like their own posts. The comment table is used to store comments.

2.1 Screenshots

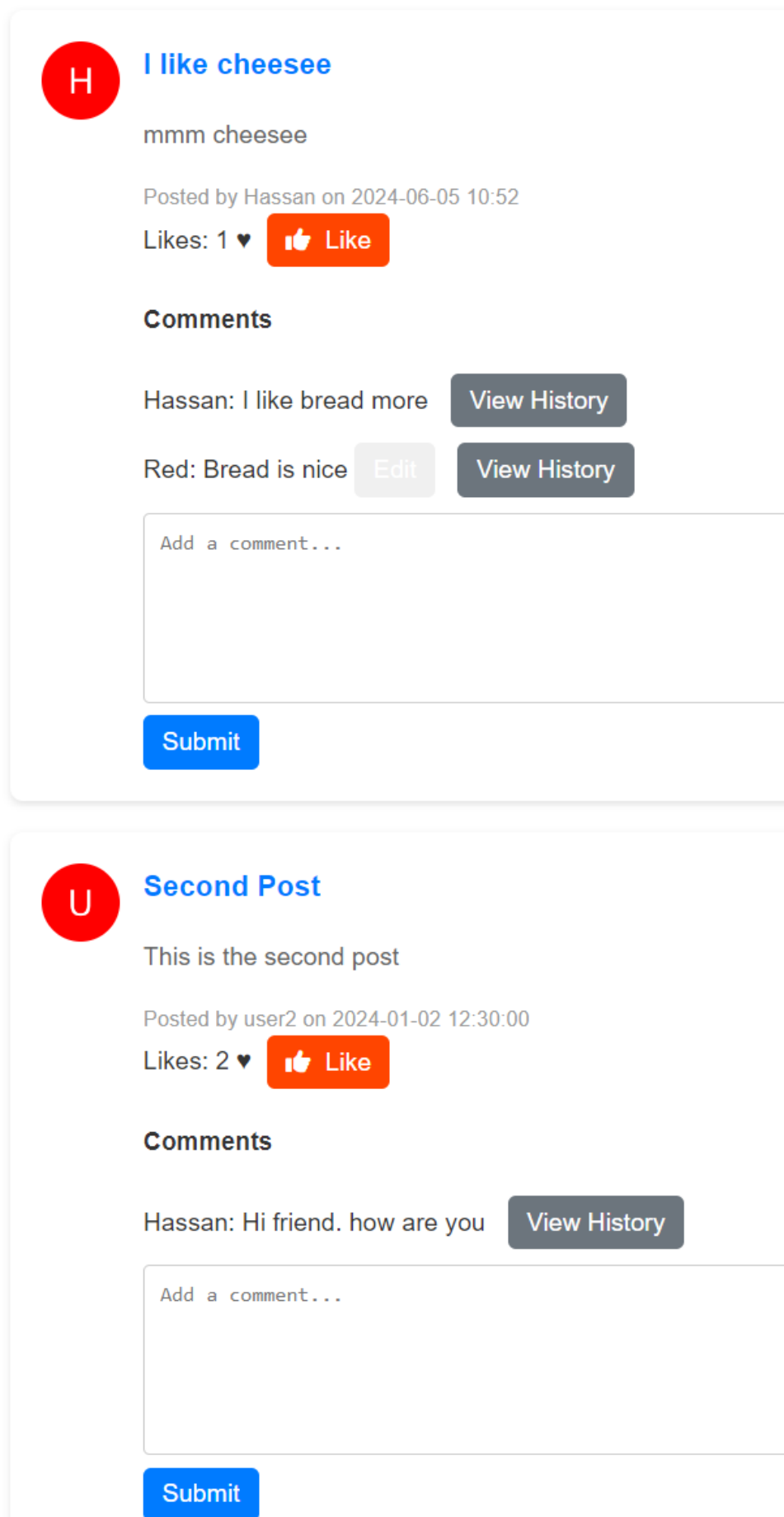


Figure 8: Screenshot of post from logged in perspective

2.2 Screenshots

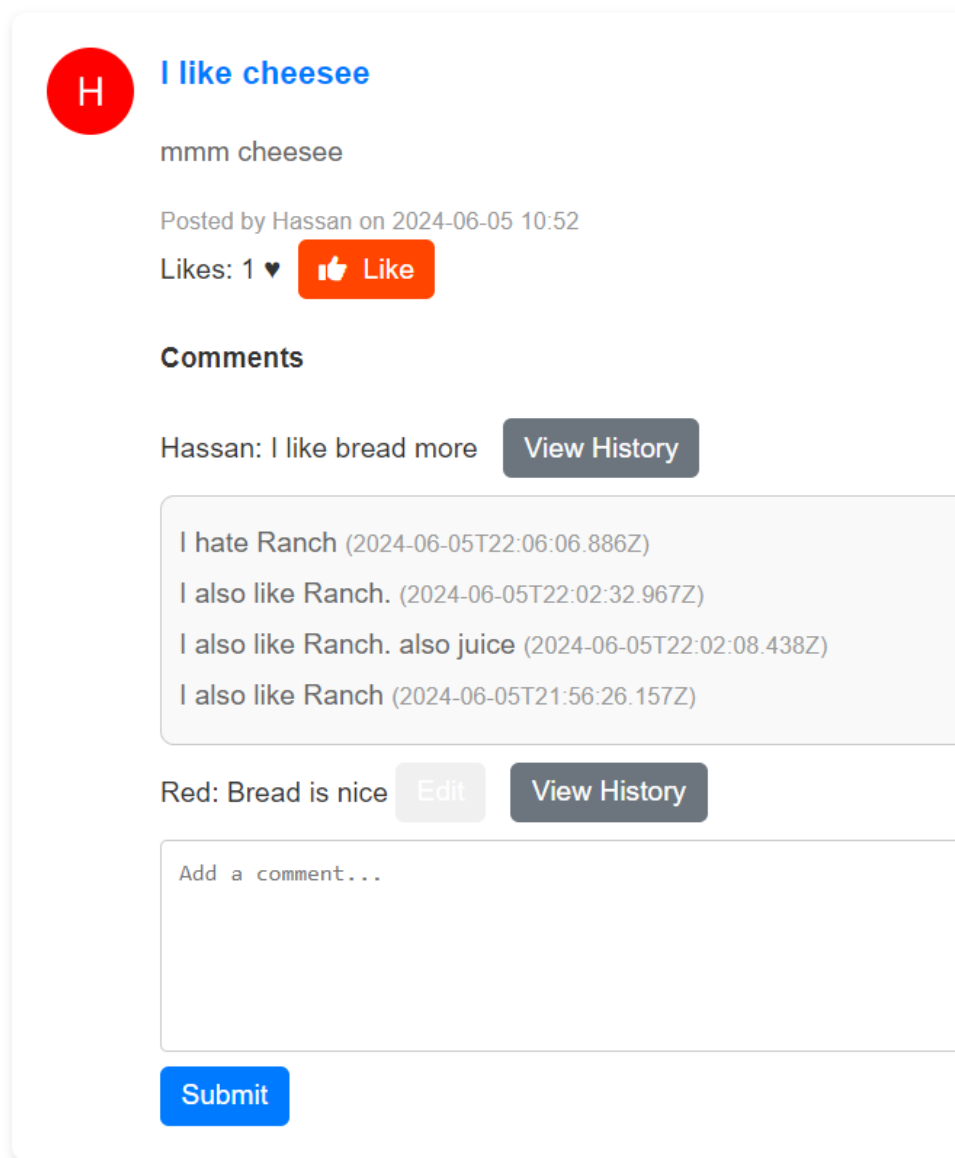


Figure 9: Screenshot of post from logged in perspective with comment history

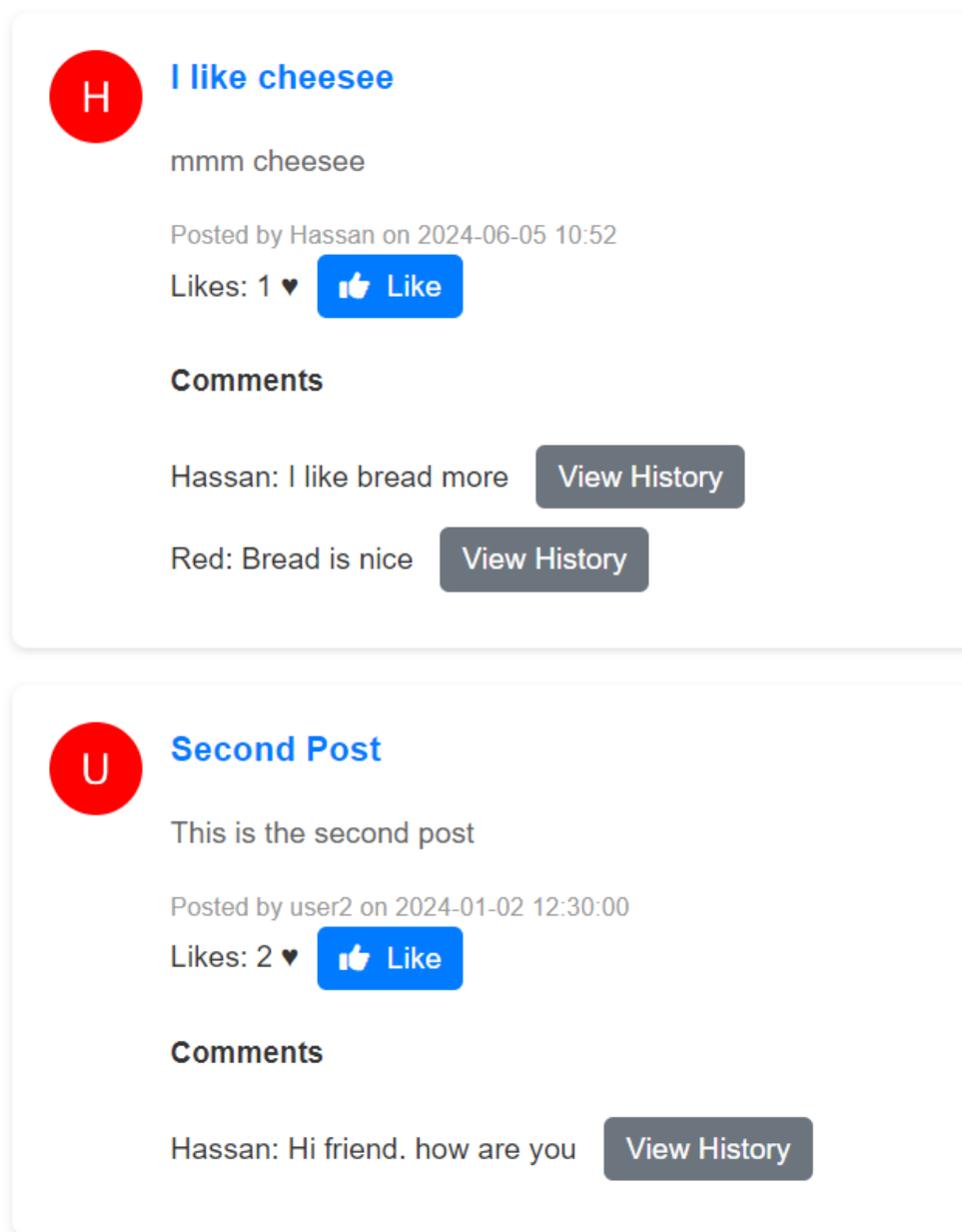


Figure 10: Screenshot of same post from logged out perspective

H

I like cheesee

mmm cheesee

Posted by Hassan on 2024-06-05 10:52

Likes: 1 ♥

Like

Comments

Hassan: I like bread more

View History

I hate Ranch (2024-06-05T22:06:06.886Z)

I also like Ranch. (2024-06-05T22:02:32.967Z)

I also like Ranch. also juice (2024-06-05T22:02:08.438Z)

I also like Ranch (2024-06-05T21:56:26.157Z)

Red: Bread is nice

View History

Figure 11: Screenshot of same post from logged out perspective with comment history

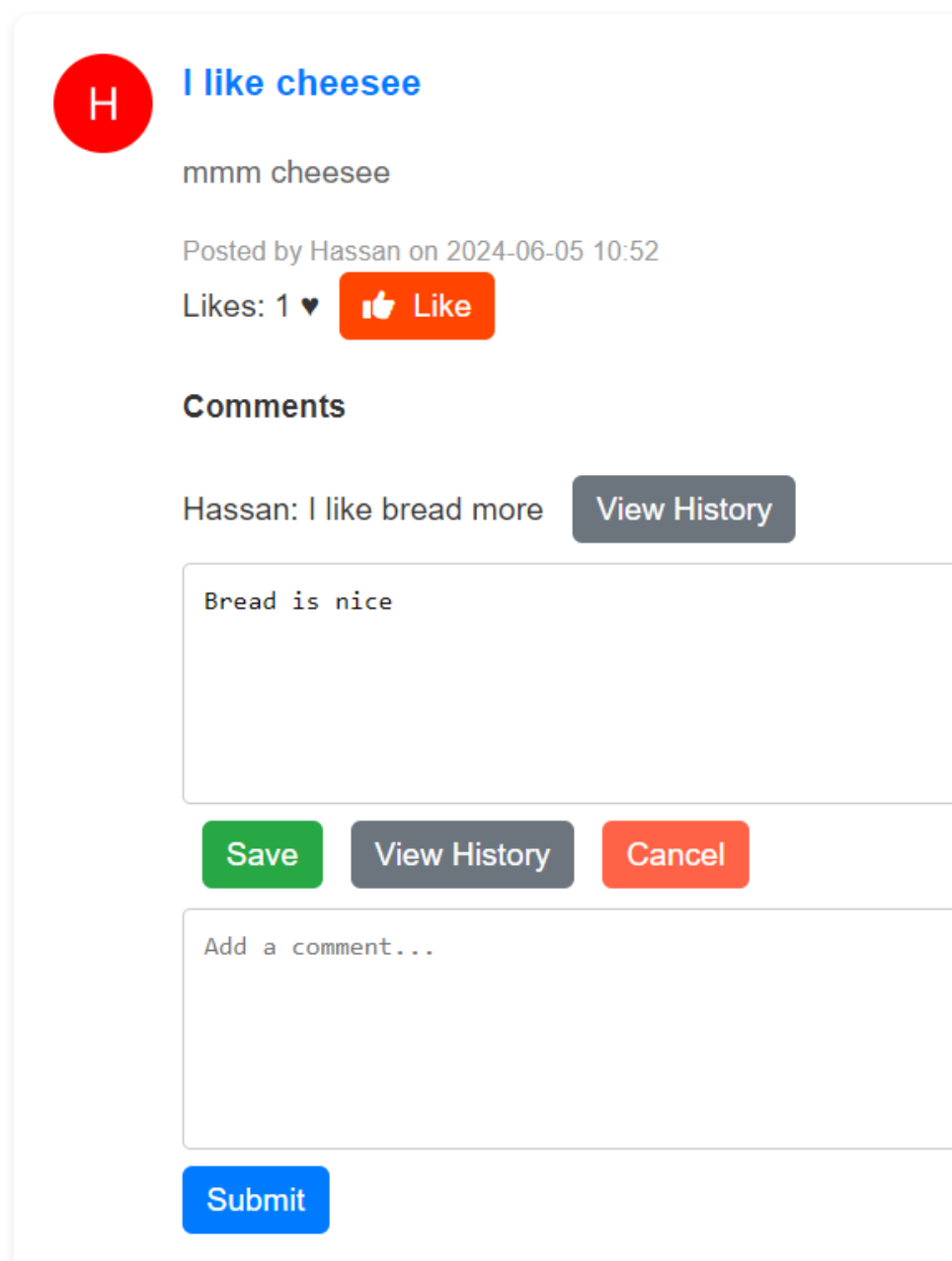


Figure 12: Screenshot edit feature

3 Profile viewing and Avatars

Profile viewing is fully functional. Users can view their own profile, and see information about their account, their avatar picture, and the posts that they have made. The avatar system is completely functional. The app is able to generate a unique avatar image for each user, save it to the public/images folder and display it to them.

3.1 Screenshots

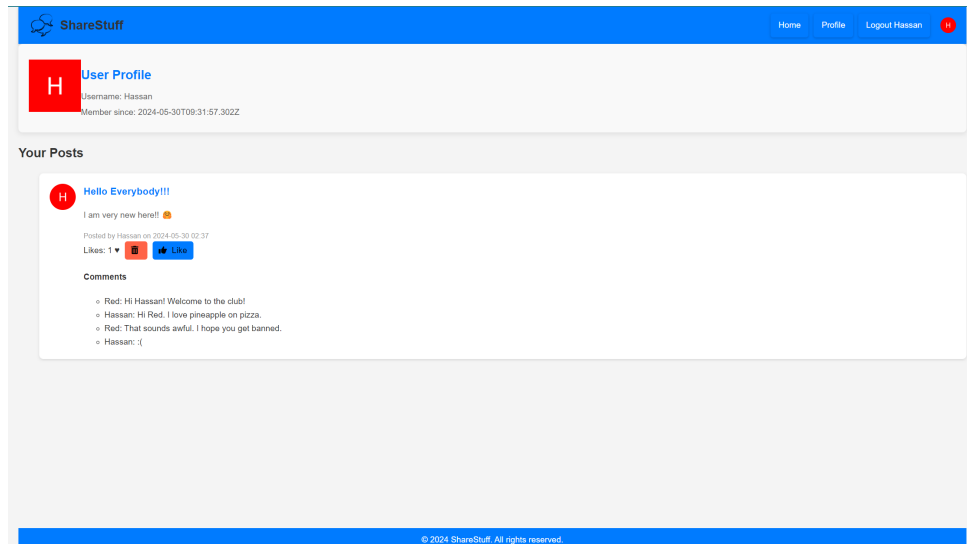


Figure 13: Screenshot of profile with one post

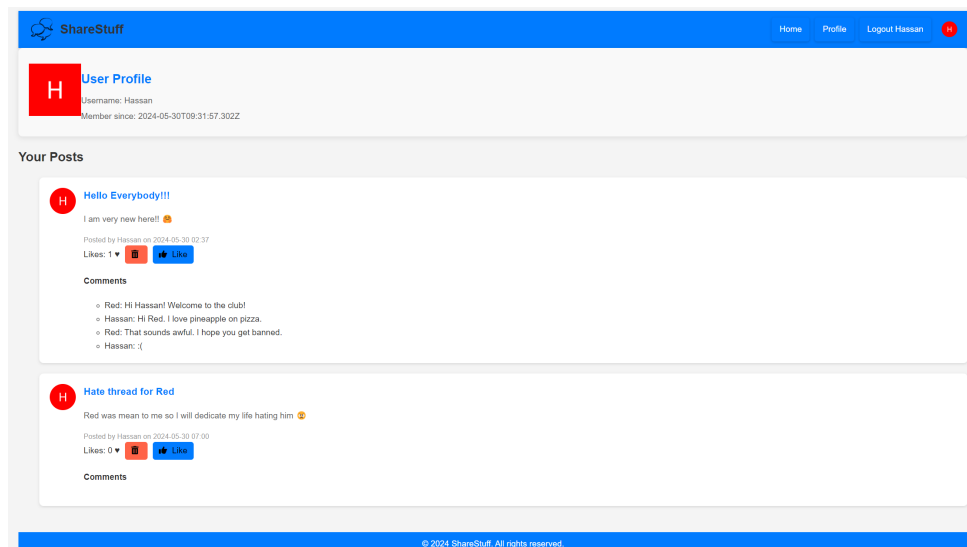


Figure 14: Screenshot of profile with 2 posts

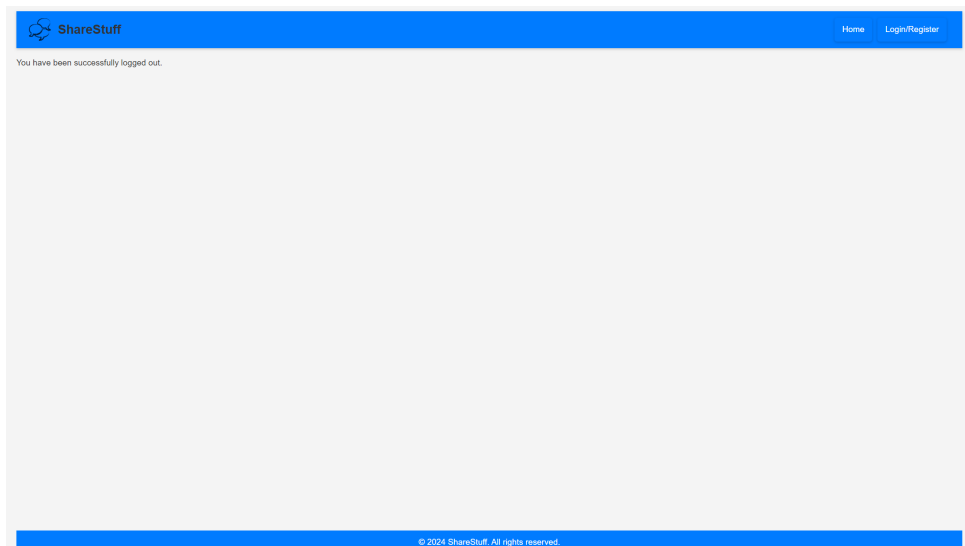


Figure 15: Screenshot of logged out

4 Conclusion

I was able to implement the main features of this project. There are still many features I can implement in the future such as viewing other user's profile, being able to personalize our profile, etc. However, for now I think I need to focus on understanding all the concepts more. I plan on recreating another app during summer using features from this project to understand more.