

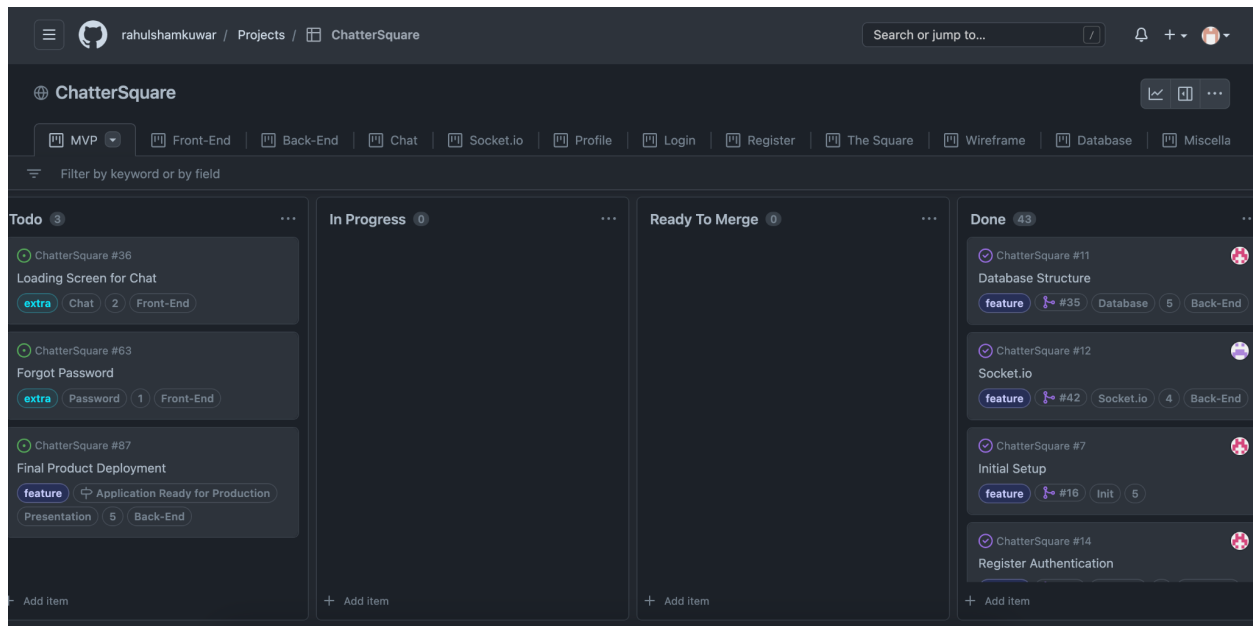
CSCI 3308 Group Project Report

Title of project: ChatterSquare

Contributors: Slater O’Brien, Sam Jansen, Rahul Shamkuwar, Raleigh Darcy, Rachel Lam, Gabriel Khabner

Project Description: ChatterSquare is a messaging application designed to reward communication. Derived from the concept of the term “chatterbox”, ChatterSquare is based on the idea of connecting people with similar interests and providing rewards to excellent communication. The more you talk, the more points you receive! Each message sent over 5 characters is 10 points. Once you gather enough points, head to your profile page to redeem them! Points can be used to redeem perks such as custom name colors, borders, and more! ChatterSquare provides different “squares” for users to chat in. Separated by different topics, users can talk with others who have similar interests. As a start, the provided topics are programming, sports, and travel. A general chat is also provided for users to simply just meet new people. The idea is to connect people while rewarding those who converse. Perhaps chatterboxes aren’t so bad after all.

Project Tracker: <https://github.com/users/rahulshamkuwar/projects/2>



Video Demo of Project:

Since the file was too large to be hosted on GitHub, we uploaded it to YouTube.

<https://youtu.be/ht-1mvag0VQ>

VCS: <https://github.com/rahulshamkuwar/ChatterSquare>

Contributions:

Slater O'Brien : Contributed to the creation of kickstarting the group project. Organizing members' information to allow fluent and constant communication. Showed up to meetings and labs. Helped with the design of the logo.

Sam Jansen : Learned socket.io and implemented the live-chat functionality, added the different topic channels for users to speak in, created the responsive frontend for the main chat page, made the message cosmetics (border type, border color, font, name color) that users purchased actually get displayed to other users, other general backend work like making it so that users must login before they can get to the main chat page.

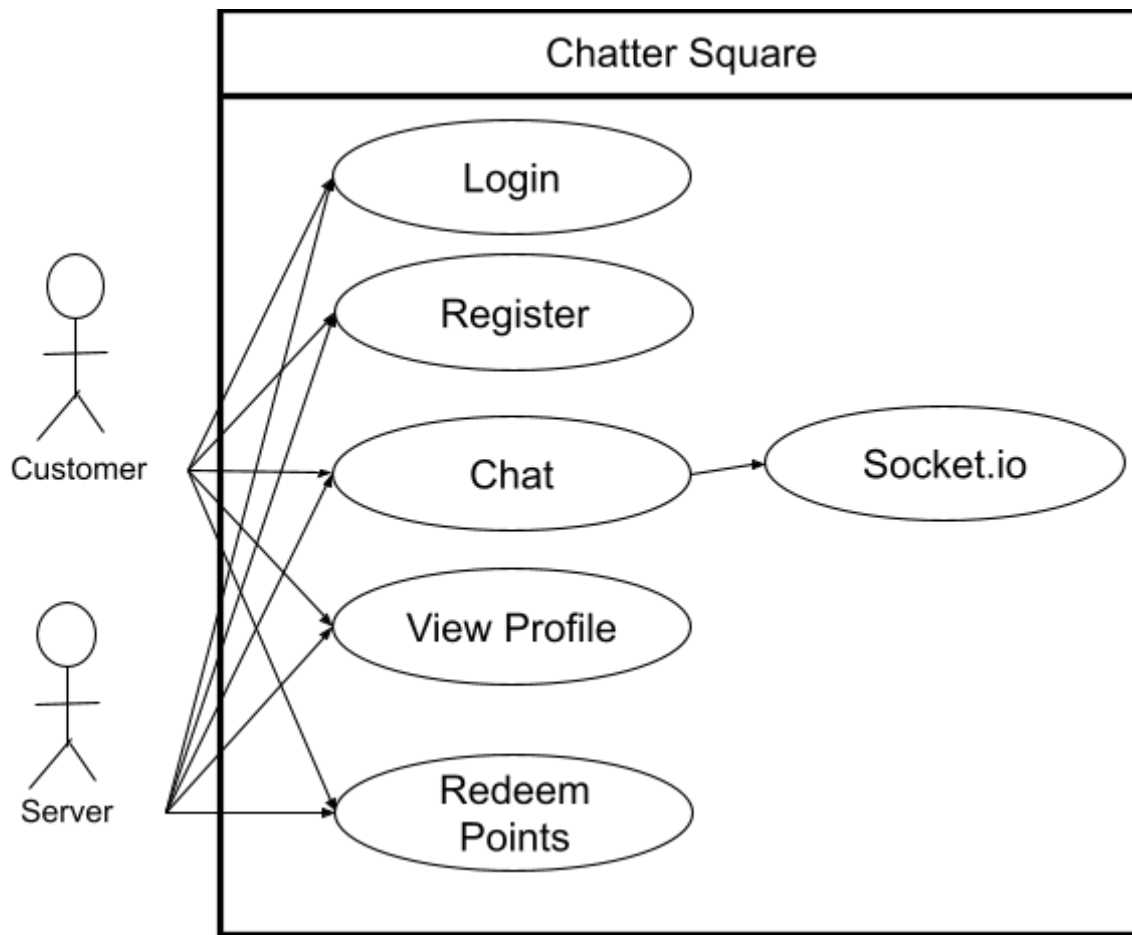
Rahul Shamkuwar : Restructured the folders to include config, controllers, routes and services, implemented the database schema as well as database helper functions, implemented login and register authentication in back-end, created initial login and register page, implemented change username and password authentication in back-end, created the modals for change profile picture and all perks, implemented logout authentication in back-end, implemented back-end functions for points system and all perks, and implemented image/GIF conversion to blobs for database storage.

Raleigh Darcy :

Rachel Lam : worked on creating some of the wireframe (Figma), finalized drawing for logo, helped with the login/register (frontend), created template for navbar (frontend), created base for profile page, provided support/feedback for front/back end when requested, kept track of most meeting minutes/documentation

Gabriel Khabner : created the register page(frontend) , showed up to meetings

Use Case Diagram:



Test Cases:

After opening the site, there are a few things to test with register/login that should lead to the proper error message

- logging in with a username that is not in the db will redirect user to register
- trying to logging in with a valid username but incorrect password
- trying to logging in with a valid username but empty password
- trying to register with an empty username field
- trying to register with an empty password field
- trying to register with an empty confirm password field
- trying to register with passwords that do not match

Login/Register normally

- registering successfully will lead you to the login page

- logging in successfully will lead you to the 'general' Square page

The Square

- message history should be displayed in their respective chats
- you can type your message in the bottom text bar and send the message for others to see
- successfully switch through different chats
- each chat section should display its own chat history
- log out from the navigation bar dropdown
- access square from other pages by pressing the logo/home button from the nav bar

Profile

- view profile from navigation dropdown
- profile should display the client's username, profile picture, and current amount of points
- there are options to change your username, profile picture, and password - each prompting a modal popup when the button is clicked
- changes are seen immediately after a successful change
- perks can be viewed and purchased from the bottom portion of the profile, each costing 100 points
- when a user does not have enough points or they have already purchased the perk, the purchase button is disabled
- logout from logout button

Test Case Observations:

The user started on the login page but switched to the register page because they did not have an account. They filled out all the fields and submitted the form successfully. After being redirected to the login page, the user logged in successfully. The user was able to view chat history and send messages successfully. The user tried to go to the profile page and commented that a dropdown menu icon would be more intuitive, so we implemented that. After clicking on the profile page button in the dropdown menu, the profile page displayed the user's information, points, and available perks to purchase. The user changed their profile picture and went back to the square page. They sent some more messages and then went back to the profile page to redeem some perks. They redeemed the font perk and it successfully showed the font on new messages. Other than the dropdown menu icon, their behavior was consistent with the test cases.

Deployment: <http://csci3308.int.colorado.edu:49167/>

When we deployed the website to CU's hosting service, we followed the process provided in Lab 12 to deploy our website to CU Boulder's private IaaS. Before the hosting service stopped working, the app was accessible through the provided link as long as the user was connected to 'UCB Wireless' or a CU Boulder VPN.

To run locally:

Clone this repository

```
// for ssh
```

```
git clone git@github.com:rahulshamkuwar/ChatterSquare.git
```

```
// for https
```

```
git clone https://github.com/rahulshamkuwar/ChatterSquare.git
```

Run the docker instance

```
docker-compose up
```

Wait for the following message log then open site using localhost:3000

```
chattersquare-web-1 | Server is listening on port 3000
```

```
chattersquare-web-1 | Database connection successful
```