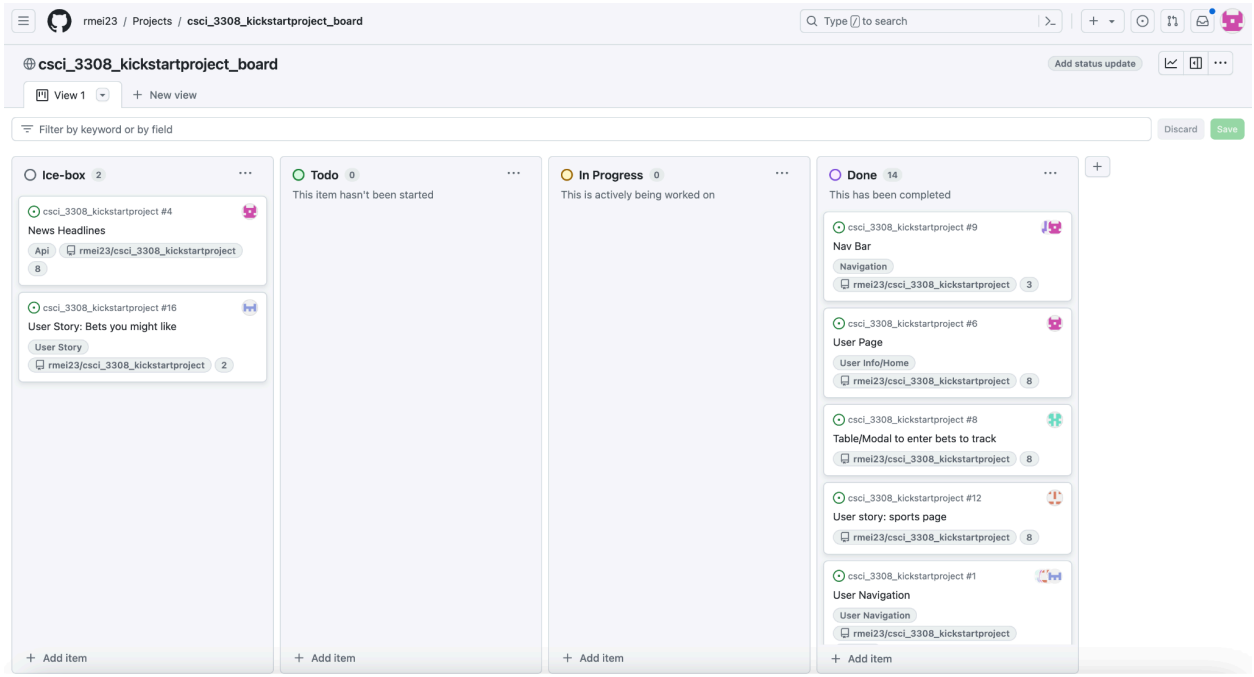
Title: SportsSphere
Team: Yuri Fung , Riley Mei , Tal Rabani , Sophia Crawshaw , Scott Davis

Project Description: SportSphere is a website designed for people that like to bet on sports. Here they can track their previous bets, easily find betting information for upcoming sporting events, and communicate with others in public forums related to specific sports. The “Sports” pages include all upcoming matches (organized by different sports) with the betting odds from the top brokers, which is very relevant information for someone looking to bet on a weekend's events. The “Bets” page includes a field for users to enter information regarding their past bets. This includes the sport, broker, amount, odds, and the outcome. These statistics are displayed in a table so the user can see all of their information in one place. Top statistics are displayed for the user on the home screen once they input some data. The “Community” page includes a chat room for each sport included in our website, where people can publicly discuss sports betting and all things related with others that share a common interest. The primary tools that we used for implementation include but are not limited to, Handlebars, Axios, Postman, Docker, and SQL.

Project Tracker - GitHub project board:
https://github.com/users/rmei23/projects/1



Video: 5 minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.

VCS: Link to your git Repository. Instructor/TAs will check, weekly, to ensure the following are stored in your VCS repository:

https://github.com/rmei23/csci_3308_kickstartproject

Contributions:

Riley Mei:
I made significant contributions across various fronts, including developing functionalities such as the home page, routes, and login/register features with CSS styling. I also collaborated with Yuri on debugging tasks like fixing Handlebar images for the about me section and ensuring consistent header/footer/navigation across all pages. Additionally, I was instrumental in creating test cases and debugging the community chat feature. In the presentation, I collaborated with Yuri on crafting the architecture diagram, tool descriptions, project scope, and challenges, enriching it with comprehensive descriptions and visual aids. I also did lab 9 with Yuri and lab 13 alone.

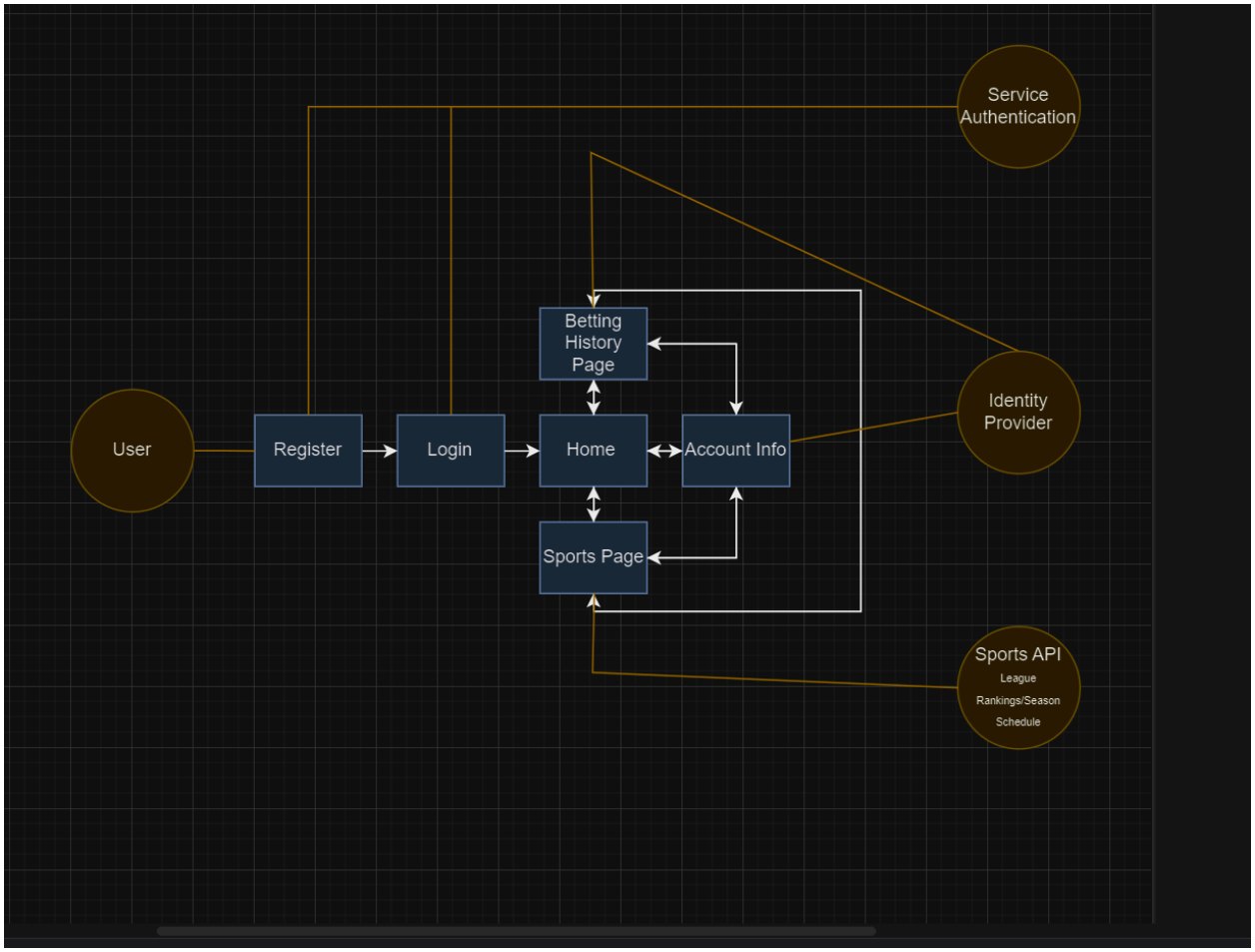
Yuri Fung:
Throughout this project, I've made contributions across multiple aspects. I designed test cases for Lab 11 and brought key functionalities to life, including the Login/Register, Home, and community pages. Debugging was a significant part of my role, from optimizing handle bar images for the About Me page to resolving issues in the Community Chat feature alongside Riley. I handled route management (index.js), crafted CSS for various pages, and maintained team meeting logs. Moreover, I collaborated closely with Riley on our presentation, covering everything. My dedication and versatility were integral to our project's success.

Sophia Crawshaw:
I worked on the front end HTML/CSS involved in the layout and appearance of the web pages, as well as fully implementing a community chat page. I worked on the home page and implemented a color scheme throughout the rest of the website. I designed and coded the community page, which is a chat forum for users to communicate on the website. This included backend, like implementing routes to handle different functionalities like rendering the chat forum page, retrieving chat messages from the database, and saving new messages to the database. I also used PostgreSQL as the database management system to store the users' chat messages. I did the front end for this by using JavaScript for interactive features such as sending messages, toggling likes/dislikes, and displaying chat messages in real-time, and using handlebars.js for templating and rendering dynamic content.

Scott Davis:
In the beginning of the project I made all of the main pages and routes except the about page to get us started, the pages were detailed and included blank fields for everyone to include our planned features. Since then, many changes were made by everyone to the front end, but I still considered that a substantial contribution. More importantly, I did everything regarding the sports page. This included the front end for three subpages (NFL, NBA, and UFC) where a third

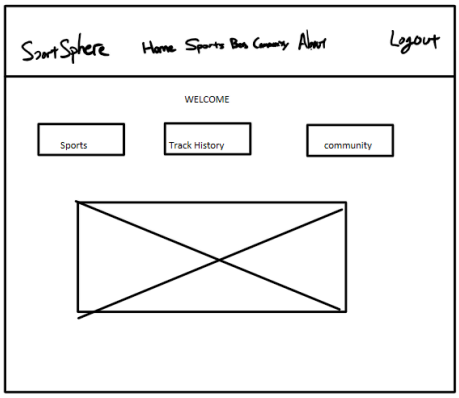
party api was called to display relevant betting information (over/unders, odds) for upcoming events. This involved using the tool axios and postman (for testing) to make the API requests in javascript, as well as handlebars to insert the API request information dynamically into the sports pages. I also did a wide variety of smaller debugging tasks throughout the process.

Use Case Diagram:

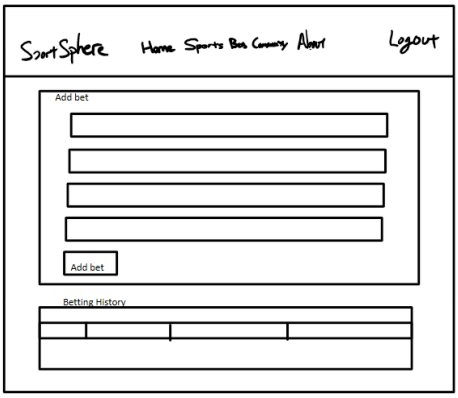


Wireframes

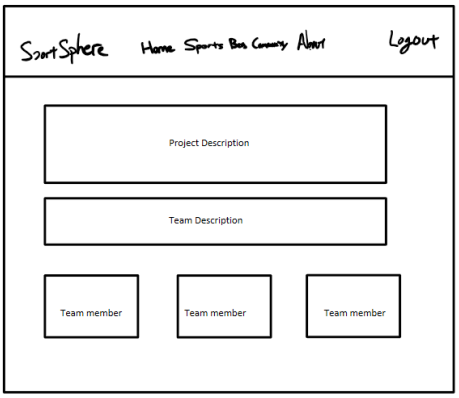
Home Page



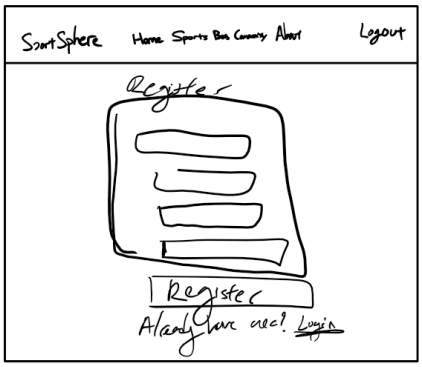
Betting Page



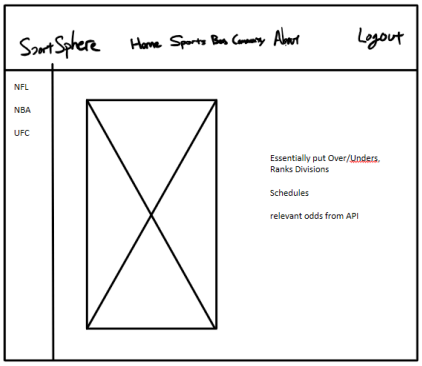
About Me Page



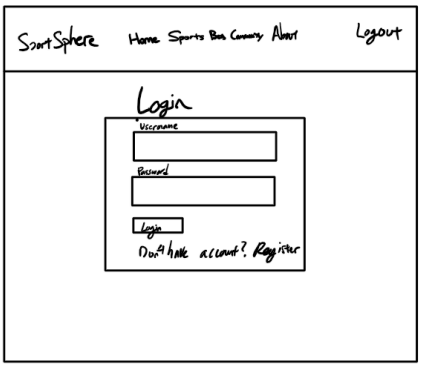
Register Page



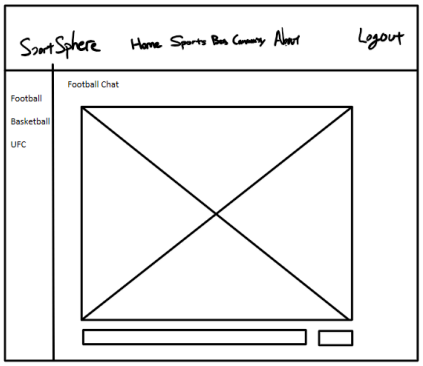
Sports Page with API integration



Login Page



Community Chat



Test results:

```
projectsourcecode-web-1  | Server is listening on port 3000
projectsourcecode-web-1  |
projectsourcecode-web-1  |
projectsourcecode-web-1  |   Server!
projectsourcecode-web-1  | Database connection successful
projectsourcecode-web-1  |     ✓ Returns the default welcome message (68ms)
projectsourcecode-web-1  |
projectsourcecode-web-1  |   Testing Add User API
projectsourcecode-web-1  | Registered User
projectsourcecode-web-1  | openening login page
projectsourcecode-web-1  |     ✓ Positive : /register (931ms)
projectsourcecode-web-1  |     ✓ Negative : /register. Checking Invalid Name/Password and DOB (79ms)
projectsourcecode-web-1  |
projectsourcecode-web-1  |   Testing Redirect
projectsourcecode-web-1  | openening login page
projectsourcecode-web-1  |     ✓        est route should redirect to /login with 302 HTTP status code (97ms)
projectsourcecode-web-1  |
projectsourcecode-web-1  |   Login
projectsourcecode-web-1  | openening login page
projectsourcecode-web-1  |     ✓ Returns the default welcome message (152ms)
projectsourcecode-web-1  |
projectsourcecode-web-1  |   Home
projectsourcecode-web-1  | openening login page
projectsourcecode-web-1  |     ✓ Returns the default logout message (252ms)
projectsourcecode-web-1  |
projectsourcecode-web-1  |
projectsourcecode-web-1  |   6 passing (2s)
```

Test Cases:

Testing Add User API:
Positive: Sending valid user registration data and expecting a successful response.
Negative: Testing with invalid inputs for username, password, email, and date of birth, expecting appropriate error responses.
Testing Redirect:
Verifying that the /test route redirects to the /login page with a 302 HTTP status code.
Login:
Checking if the login page returns the default welcome message.
Home:
Ensuring that the logout route returns the default logout message.

Observations:

Users interacted with various endpoints such as registration, login, and logout. Actions performed by users were aligned with the expected behavior as defined in the use cases.
We successfully passed all integrated tests in addition to those that we added independently to ensure that all features were working as expected and intended. We had some issues when the routes got changed when working on other elements of the project, but Yuri fixed it with some help from Riley. There was also an error with user duplication if the test case tried to create an example user that already existed, since the user id needs to be unique it would throw an error if the user persisted from a past session so we had to fix that as well.

Deployment: The app was deployed to Microsoft Azure through a vm, however since the free student tier has limited uptime, the project cannot be hosted indefinitely. When the server is up, the project is accessible at http://recitation-16-team-02.eastus.cloudapp.azure.com:3000/register.
The server should be up when checking this report, unless we run out of resources.