Ruan Abarbanel
Chasen Goren
Robiel Kennedy
Aaron Van
Joshua Gildred
Rohan Adepu

SafeBets

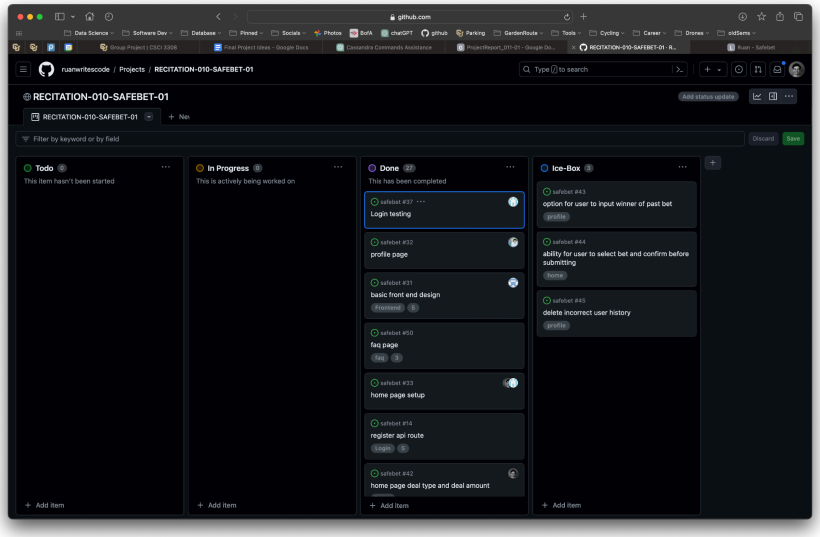
Project Description:

SafeBets is a powerful tool centered around the functionality of transforming real-time odds and sportsbook offers into guaranteed winnings. The application uses an external API to gather betting odds from various sportsbooks and filters them to give a consistent list of the top sportsbooks in the United States. Users have the ability to view a variety of potential outcomes by selecting their desired sportsbook and attaching given offer values. The event granularity consists of all sports leagues currently active from the API and outputs odds for each sportsbook per event. Every user on SafeBets must sign up through the register page requiring basic data like a username, password, full name, email, and age. To comply with state and national regulations SafeBets is limited to individuals over the age of 21.

Upon sign-in, users will have the ability to view the home page which is the core tool of the application containing all events, sportsbooks, odds, and calculated winnings data for making informed betting decisions. Additionally, SafeBet users will be able to select bets and view all their past bets/winnings through the profile page. A useful help page will connect users to an admin in order to ask questions and communicate the functionality of the application. The remaining web pages include an About page and an FAQ section.

Project Tracker - GitHub project board:

https://github.com/users/ruanwritescode/projects/1/views/1



Demo Video:

https://drive.google.com/file/d/1IH_vDaaZQCTLmmBnwgECmWPF9v5V4vFm/view?usp=share_link

VCS Repository:

https://github.com/ruanwritescode/safebet

Individual Contributions

Joshua Gildred

I wrote both the help and the FAQ page. This included implementing the Talkjs api into our project. Setting up the account transfer from our database to the Talkjs database, and creating a page specifically for the admin account to interact and help users. I used handlebars and bootstrap to implement the pages and endpoints. I also helped with the setup for the login route, footer partial and header partial.

Rohan Adepu

I worked on the About page. I created the app description and added team member cards using Vue.js for styling purposes. I also used CSS and bootstrap to make the page look better overall. I also was in charge of setting up and configuring the Microsoft Azure Virtual Machine to house our domain.

Chasen Goren

I worked on the general framework of the profile and FAQ pages. Including but not limited to fixing smaller visual errors from Bootstrap, smaller issues involving the index.js endpoints, and making sure things are formatted correctly in HTML to the w3c guidelines. I also helped Ruan with designing aspects of the home page and endpoints involved in the creation of that.

Ruan Abarbanel

My work encompassed building and populating the index.js file including writing the api endpoints for login, register, the home page, the external API call, saving user history/adding bets to the Safebet database, the user profile page, and logout. I designed and implemented the SafeBet database model, provided Handlebars helpers to increase ease of data processing in HTML and revised/ensured stylesheet consistency. I created the home page design, including implementing the algorithm for querying the “odds-api” API and displaying the outcomes as well as all its front end design.

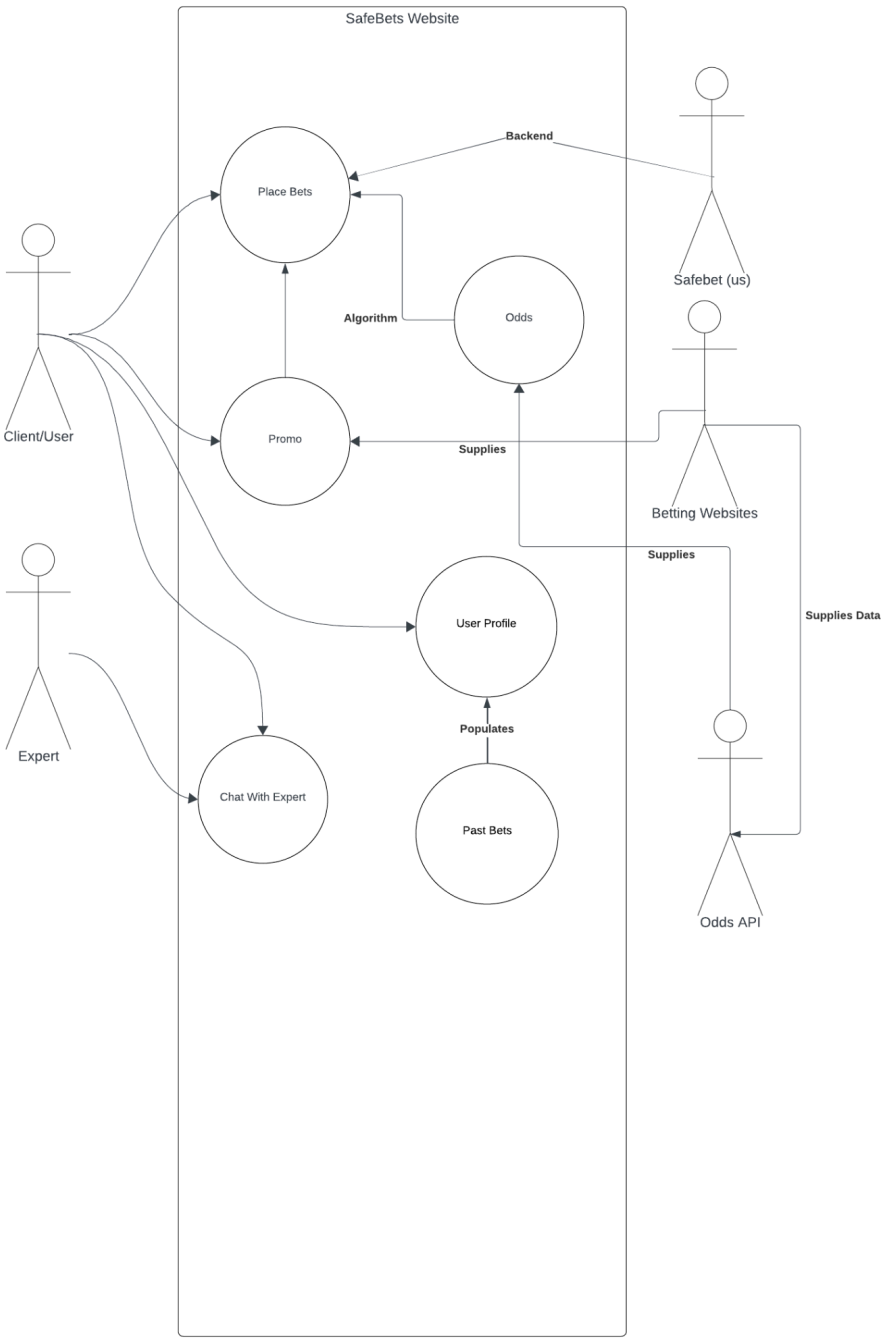
Robiel Kennedy

I worked on the initial design of the help page and incorporated unit testing for the profile page and help page. The main idea for the help page was incorporating sockets where one endpoint can link “admin” and the other endpoint “user” would be linked for communication to occur. However the use of handlebars made it difficult until we used the Talkjs API where we linked our database to the Talkjs database which within the API incorporated sockets and worked well with handlebars. Created test cases that were used for pages more specifically on the profile page and home pages.

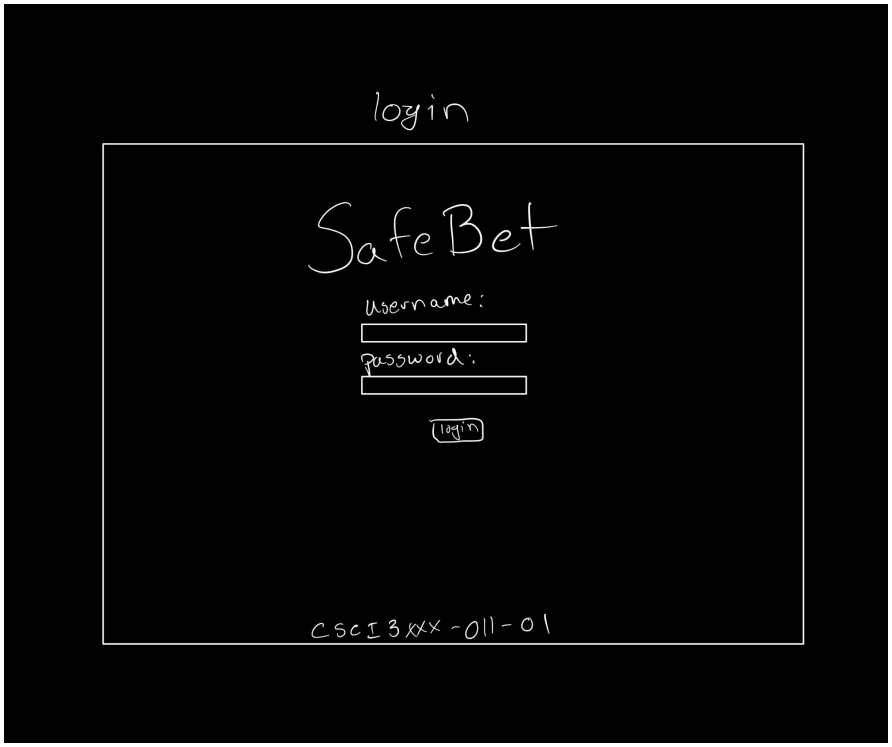
Aaron Van

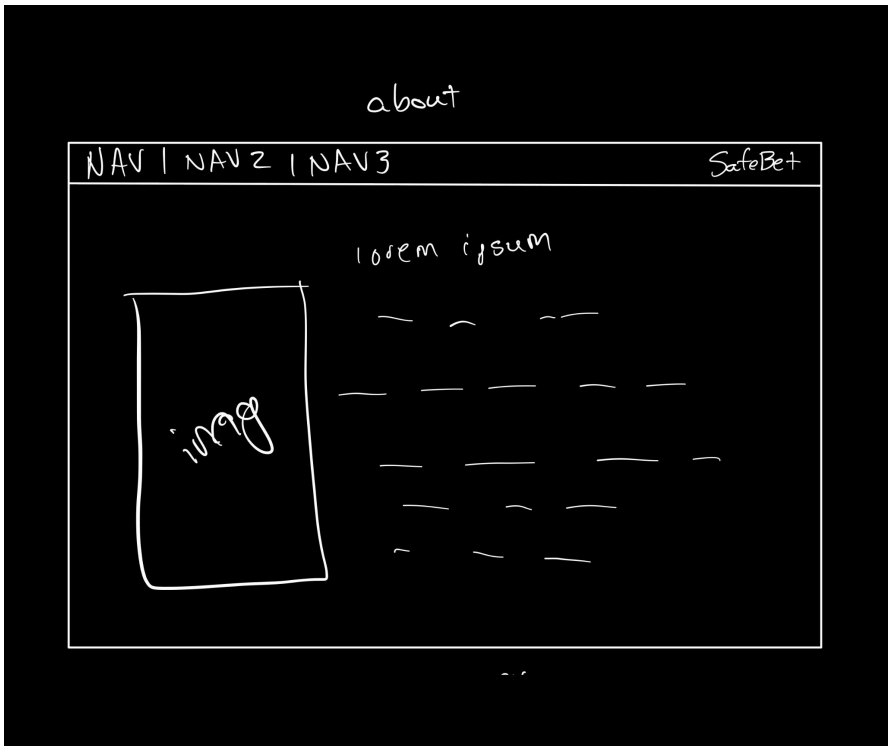
I worked on the initial design and css framework for login and register pages, and created/tested unit tests. To further explain, I provided a rough outline of the login and register page, where throughout the process was improved by the group. For unit test cases, I created positive and negative test cases for the login page (valid/invalid login credentials), profile page (data is displayed correctly), help page (inputs to the chatbot), home page (deal amount when selecting bets), and registration page (testing age verification).

Use Case Diagram



Wireframes


login
SafeBet
username:
password:
login
CSCI3XXX-011-01


about
NAV1 NAV2 NAV3
SafeBet
lorem ipsum
img

Home

HOME | HELP | ABOUT SafeBet

Active Deal : [Free Bet] [$$$ ✓]

Current Eligible Odds [sport✓]

	1st bet	2nd bet	winnings
–	event: bet <	event: bet	$$
–	. ..	. ..	$ $
–	. ...	. ..	$ $

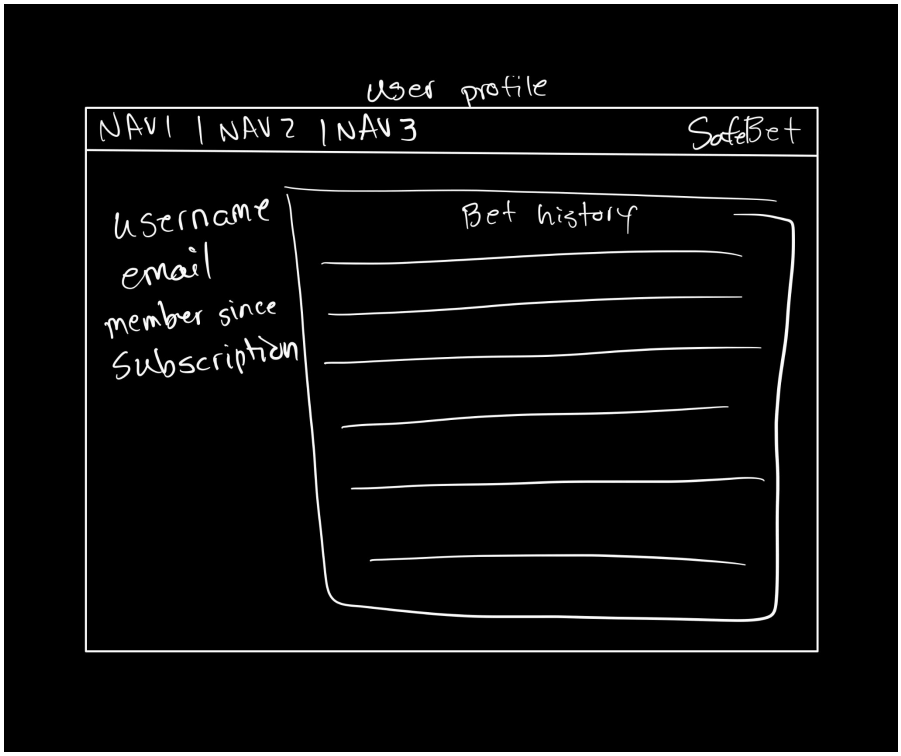
help

HOME | CHAT | ABOUT SafeBet

chat with an expert

user: how does this work
expert: great question!

[ ] [SEND]



Test results

Test 1: Positive Deal amount for bet

- Positive test case: deal amount is positive number.
- Negative test case: deal amount is negative number or zero (any negative number or zero)

Test 1 results: When the user navigates to the home page, the user will have to input a deal amount before they can select a bet. When the user enters the deal amount, depending on how much they inputted, a list of the best possible bets will be displayed, and from here the user can select a bet from the list. If the user enters a negative number, the site will ask the user to input a positive bet amount.

Test 2: Correct Login credentials

- Positive test case: Login with correct username and passcode
- Negative test case: Incorrect passcode with correct username

Test 2 results: After the user registers for an account by creating an username and passcode, the user navigates to the login page to login and be redirected to the profile page. The user inputs their correct username and passcode, and after clicking login, the site redirects them to the user's profile page, displaying their information. When the user enters their correct username but incorrectly types their passcode, the user will be denied access to login in and will be asked to retype their username and passcode.

Test 3: Input for chatbot

- Positive test case: A chat that is entered is non-empty (e.g. “Hello, I need help with placing a bet”)
- Negative test case: A chat that is entered is empty (e.g. “ ”)

Test 3 results: When the user navigates to the help page, a chatbot will load up and allow the user to type to the chatbot (the admin will receive an email and respond to the user that needs assistance). When the user types something like, “Hello, I need help with placing a bet”, the admin will be notified that a user has asked for help through email, and one of our admins will assist. If the user inputs an empty message, the chatbot will not allow the user to even send the message until something is typed.

Test 4: Age verification

- Positive test case: Age is entered above legal limit (21 years)
- Negative test case: Age is entered as below legal limit

Test 4 results: Upon registering for a SafeBet account. Users must enter their birth date to be verified for appropriate age in compliance with state and federal laws. If the user enters an age above the legal limit of 21 years, the registration will submit without delay and create a new user account. If the user enters a birthdate which results in an age less than 21 at the time of creation, no account is created and an error message displays notifying the user they must be 21 or older to participate.

Deployment

The SafeBet web application is cloud-hosted on Microsoft Azure. It is accessible through this link: http://recitation-010-team-01.eastus.cloudapp.azure.com:3000/register