Project Report - MatchPlay

Team Members:

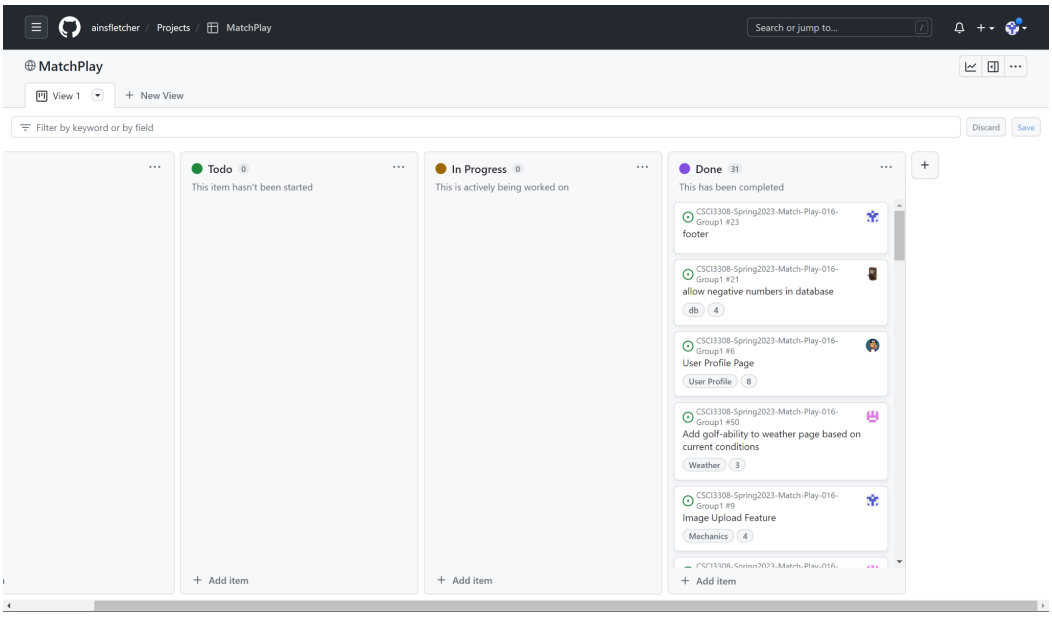
- Ainsley Fletcher
- Matt Lattin
- Nhat Ngyuen
- Shaun Kittrell
- Adam Carlson
- Moses Bazzi

Project Description:

- MatchPlay is a matching platform that links golfers in Colorado based upon personal interests as well as skill level.
- The problem we aimed to solve with this project is to eliminate some of the uncertainty that comes with playing a round of golf with random people
- In order to accomplish we created databases to link things such as matches between users and user's and their respective profile photo

Project Tracker:

- Github Link: https://github.com/users/ainsfletcher/projects/1/views/1
- Screenshot:
-



Video:

- https://drive.google.com/file/d/1tF5E4Tg9QboNOPx0U6UBXl8hySvbr-iv/view?resourcekey

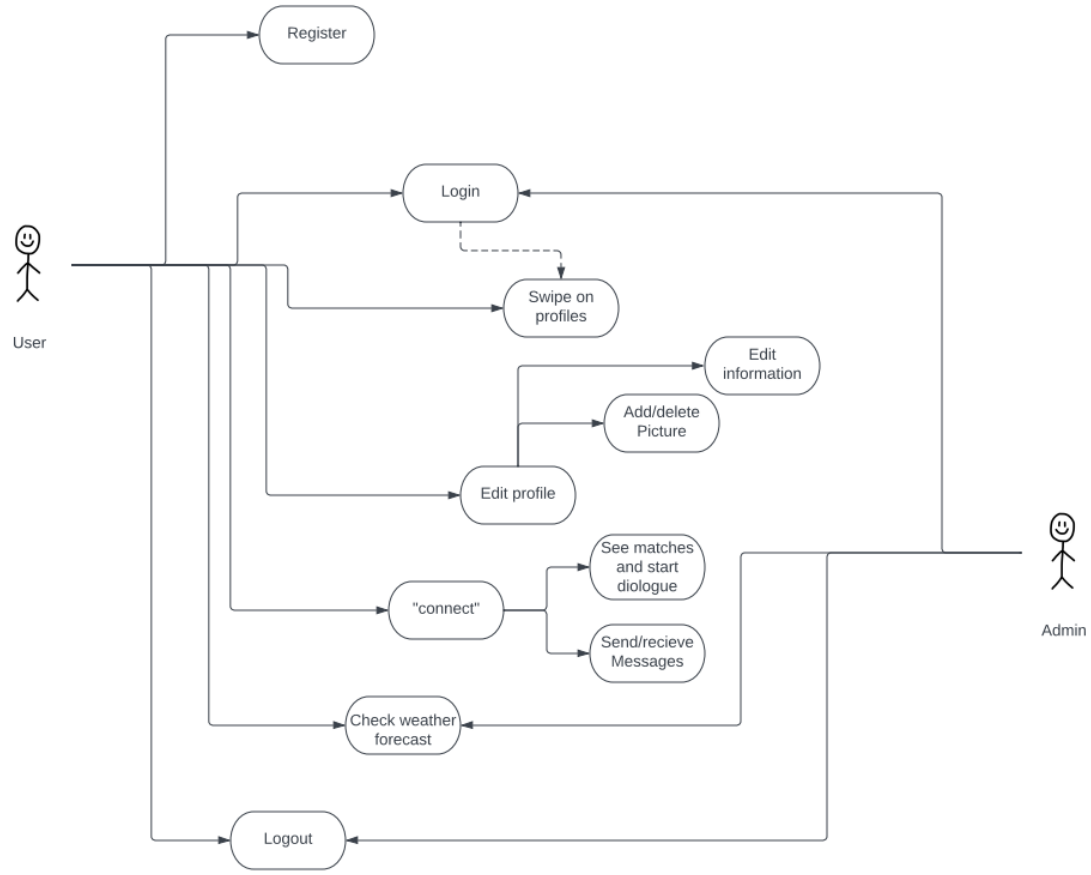
VCS:

- https://github.com/ainsfletcher/CSCI3308-Spring2023-Match-Play-016-Group1

Contributions:

- **Ainsley Fletcher:** I was responsible for the backend for the matching feature and the database. For the matching logic, I created relational tables using primary and foreign keys to relate users that have liked/disliked each other. I also worked on the three match statuses: match, not matched, and pending match. Each time a like or dislike button is pressed, there is an API call to represent this change in the database. The user will be able to see this change through the frontend.
- **Matt Lattin:** My primary focus was to create a sleek, user-friendly interface for the Profile page. The style of this page pays homage to a score card you might receive at a golf course. On top of that I was able to add an edit button that only allows users to edit once they click it which made the profile page look more professional. Additionally, I had to write an API call that would update this information one aspect at a time if necessary. Finally for the background images you saw, I tried to choose some of the most famous holes in golf to really get the user in the golfing mood.
- **Nhat Nguyen:** I mostly worked on the front end side of the project. I worked on the register page, login page, dark mode and footer. For Register, Login page and the footer I worked on styling and format of the page using bootstrap. For the profile page, I worked on typing animation of Find your next MatchPlay!. I also worked on the Darkmode functionality of the website making sure that the user's preferred site setting is saved in their local user storage as well as adding some input control for profile page update.
- **Shaun Kittrell:** I was responsible for the weather page on the front end and the back end. I helped a little bit with the back end of initializing the database tables. Finally I helped debug issues and support my teammates.
- **Adam Carlson:** Ainsley and I were responsible for developing the database to store the user data. The user data consisted of the following: name, handicap, age, home_course, movement, bio, phone_number, image_url. After these tables were initialized in the database it was my responsibility to display the user data on the Discover page. Once the users data was displayed on the Discover page Ainsley and I had the responsibility to implement “Like” and “Dislike” buttons. The “Like” button would move the user to the Matches page
- **Moses Bazzi:** I was primarily responsible for allowing users to upload images using various technologies including Cloudinary. This required implementation of various pages including “Profile”, “Matches” and “Connect”. Moreover, I designed and implemented the custom navigation bar as well as contributed to the overall aesthetics of the website.

Use Case Diagram:



-

Test results:

- The test results for our plan contained the login, the feature for updating user information, and calling other users information. We implemented these cases as we were working on the backend, testing as we went. The mocha and chai tests create a fake user and make sure that the register feature is working. It also logs this user in to test this functionality. For updating information, we ensured that the fields have certain restrictions so that the user won't input the wrong information. This way our database won't run into any errors during the update queries. For calling other users information, it first checks to see if the current user is in the database, then it will not display the current user. It will instead show all other users in the database.

Deployment:

- Access to link and Azure deployment screenshot:
  - https://github.com/ainsfletcher/CSCI3308-Spring2023-Match-Play-016-Group1/blob/main/Milestone%20Submissions/lab13_Azure_screenshot.png

- Primarily our project was deployed using Docker and our local systems. For one to access the project, we would be required to publish it on the Internet which would necessitate a virtual machine or server. Alternatively, one could copy the contents of the “src” folder on our Github repository and run a container on their machine to deploy the project.