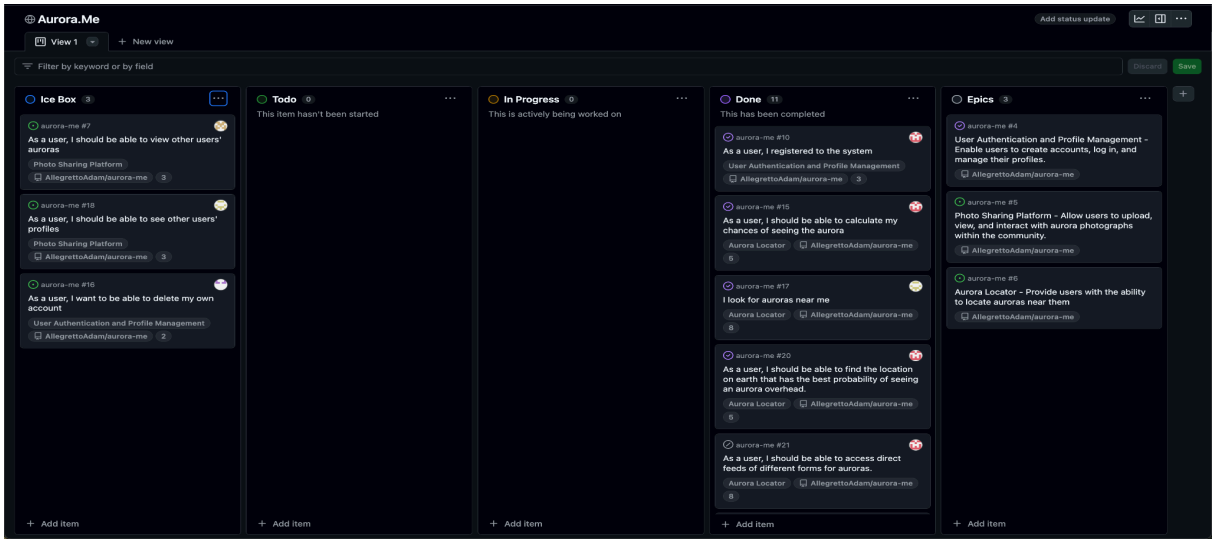
Aurora.me

Team Members:
Abraham Yanez Escobar , Adam Elliot, Aur Shalev Merin, Ramsey Vincent, Ulises Cortez

Project Description:
Our main purpose in creating Aurora.me was to give anyone around the world a place to connect with others through the beautiful celestial bodies that are often only observed in the most northern and southern parts of the globe. From the average person who enjoys a pleasant view, to enthusiasts with genuine wonder for the forces behind the lights, we felt the need to create a refined environment dedicated to the auroras. Apart from connecting people through the internet, we also wanted to add functionality for finding auroras and letting people learn about the scope that auroras reach. Aurora.me is a program designed to show users their chance of seeing an aurora from a coordinate location that they input. It is also designed to connect the user with other aurora enthusiasts and share their experiences with photos. The user is able to create an account through which they can upload and view their own photos, change their profile picture, and see the posts of other users. The social page displays the images in order of the newest posts first so that users can stay up to date with auroras at the current moment.

Project Tracker:

[Project Tracker Link]



Video Demo: Link

VCS: Repo Link

Individual Contributions:

Aur -
I had a couple of different focuses, my main focus was putting out fires and helping other team members get "unstuck" in their assignments. I created the initial API call that outputted the aurora data to the terminal, styled the profile page, created the profile picture and post upload on the profile page, wrote the code to display the posts on the profile page, came up with the logic to upload photos using the base64 encoding and multer, wrote most of the code tests, fixed many of the issues on the index.js file in regards to session management, and wrote the use case diagram, among other tasks.

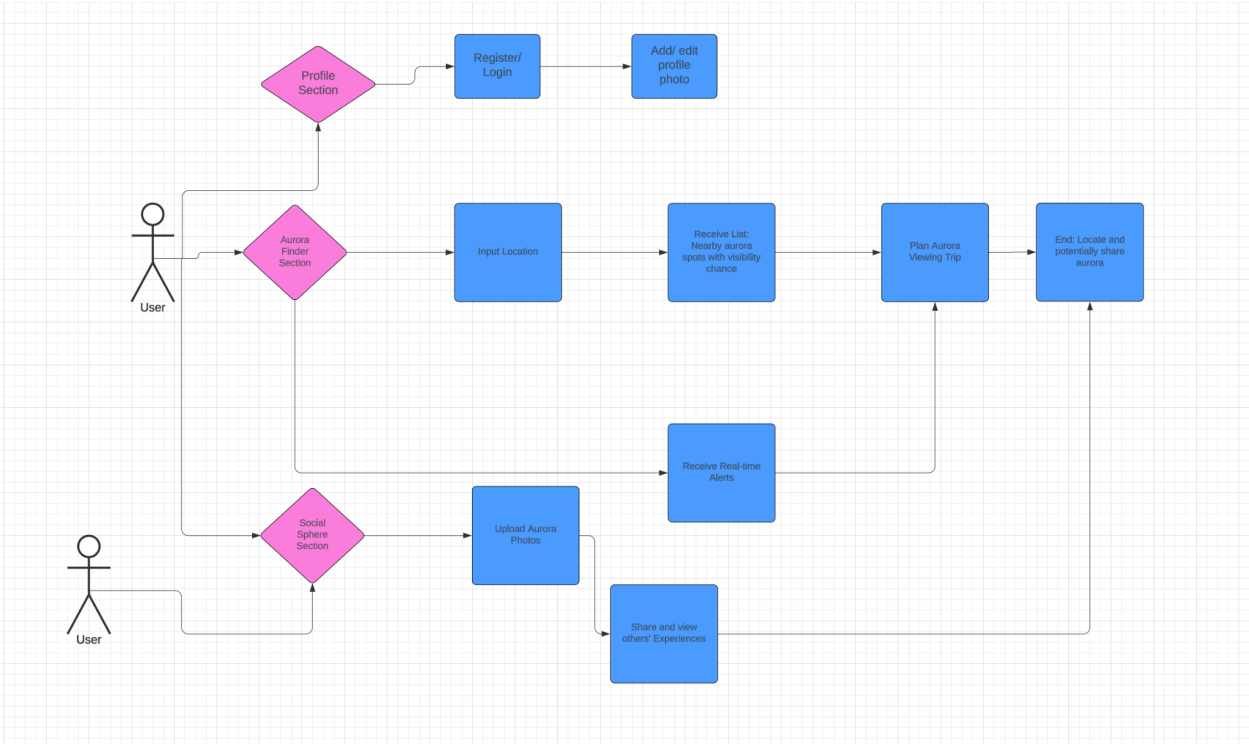
Adam -
I mainly worked on the External API integration with Auroraslive.io while also working on different components of the early versions of the website, from architecture to templates and such. Additionally, I took the lead on setting up Render for our application and helped to review and update my fellow members' code whenever I could. This involved working with nearly the entire tech stack, with the exception of Chai, with most of my focus being on NodeJS and Handlebars as I figured out how to call, retrieve, and route the API data and format it.

Ulises -
My main focus was implementing the API routes in NodeJS for all of the pages in our application, excluding those related to the External API integration and profile. This involved developing the get and post endpoints to ensure seamless functionality across the app. Additionally, I also wrote some of our test cases and made sure they passed, which involved dabbling with the database as well. I also was responsible for the CSS in the home, login, register, and logout pages.

Abraham -
My main focus was on implementing the social page which displayed all of the posts that users had made to the website as well as letting users post on their own. The CSS formats the page with the posts going down the middle and the user information and caption for each post showcased above the images. Using a loop made in the Handlebars file and query that gathered all the posts together in order of newest first, the social page came together properly.

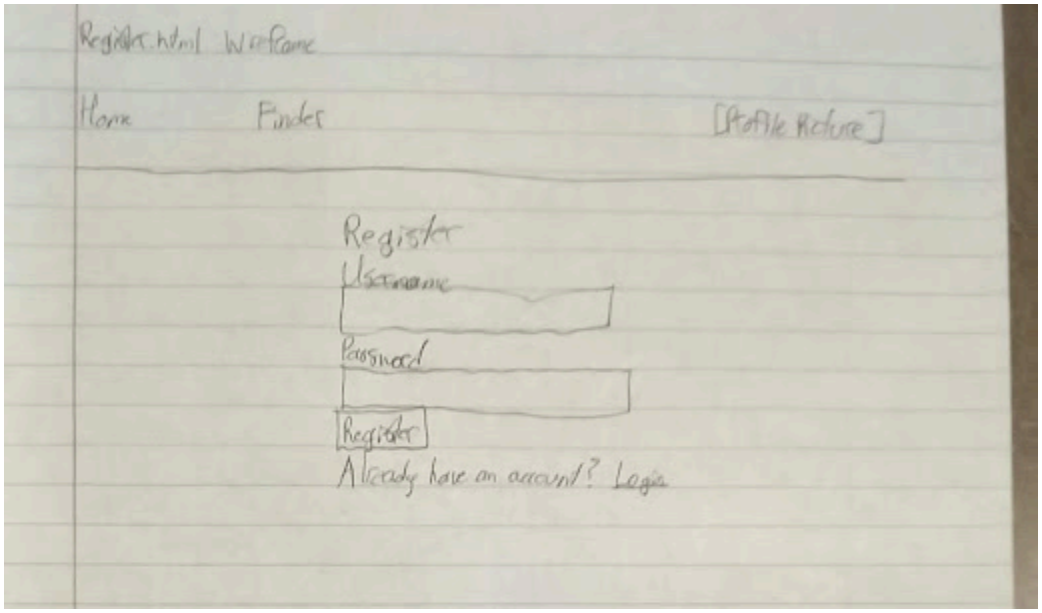
Ramsey -
I primarily setup the PostgreSQL database as well as design the profile page (though some of the css is not my handiwork). Unfortunately, I did not contribute much else because most of my time was spent looking into how to do third-party cloud storage for image uploads, a feature we ultimately scrapped due to time constraints.

Use Case Diagram:

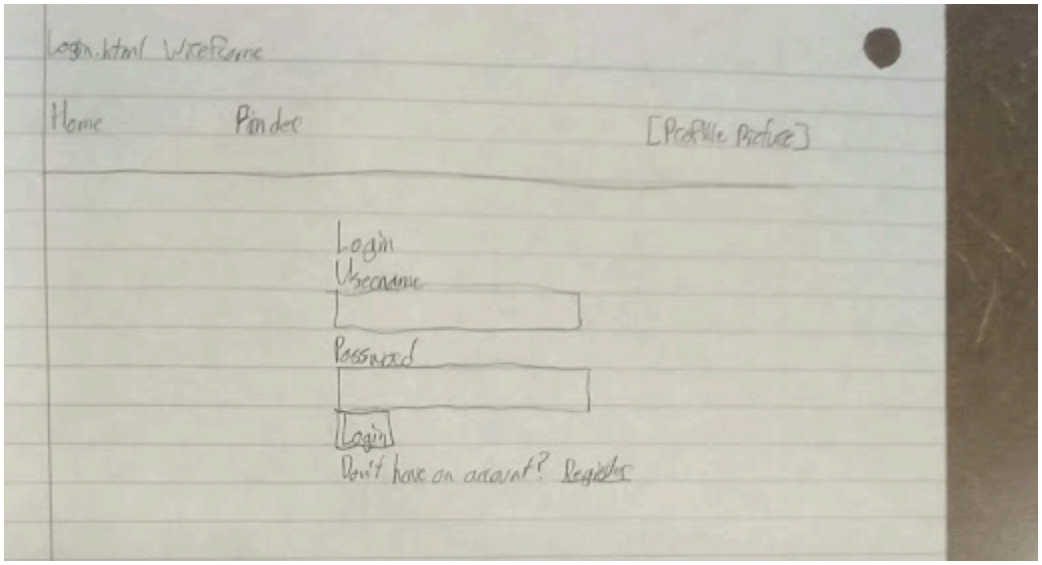


WireFrames:

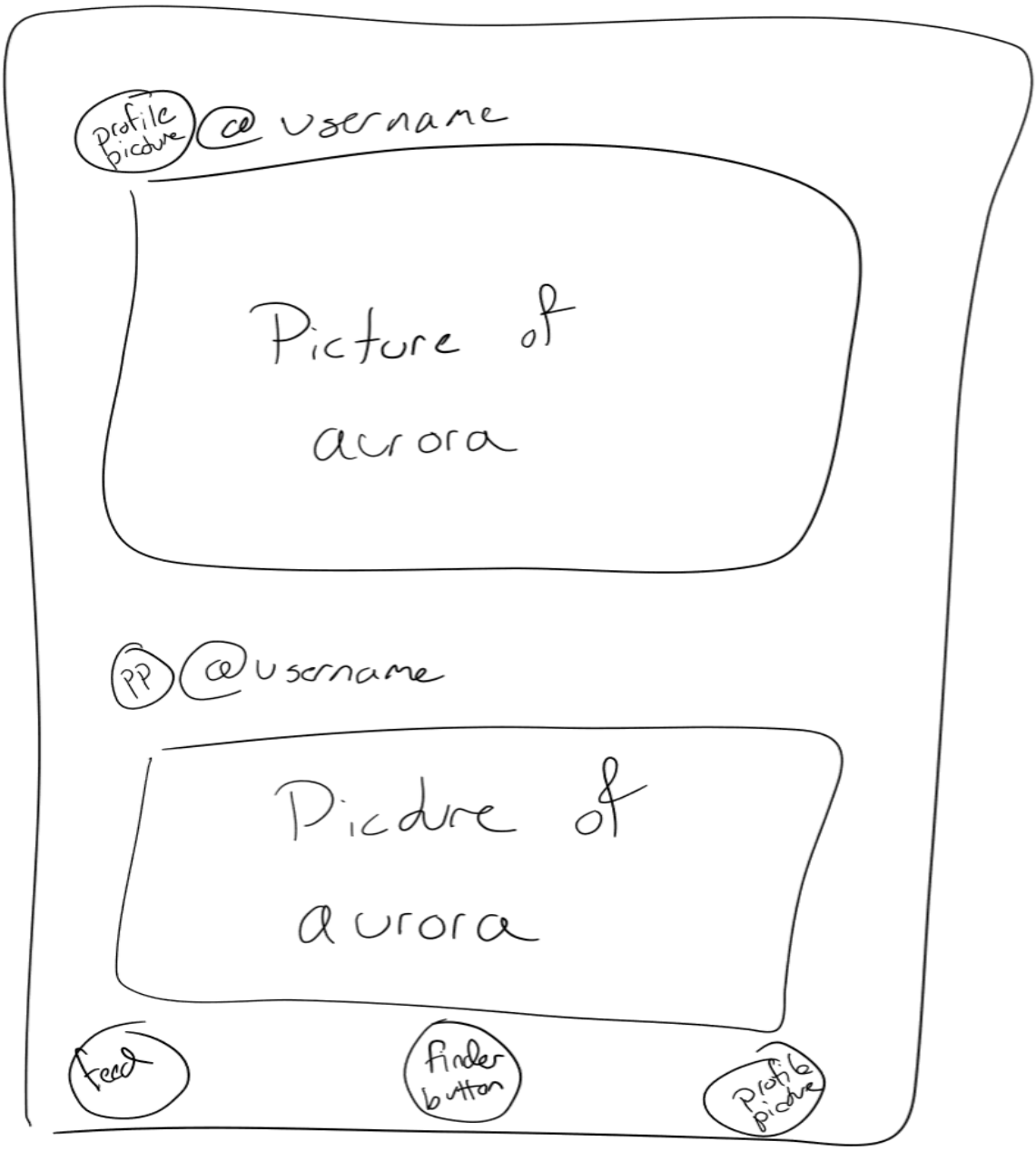
register.hbs -



login.hbs -

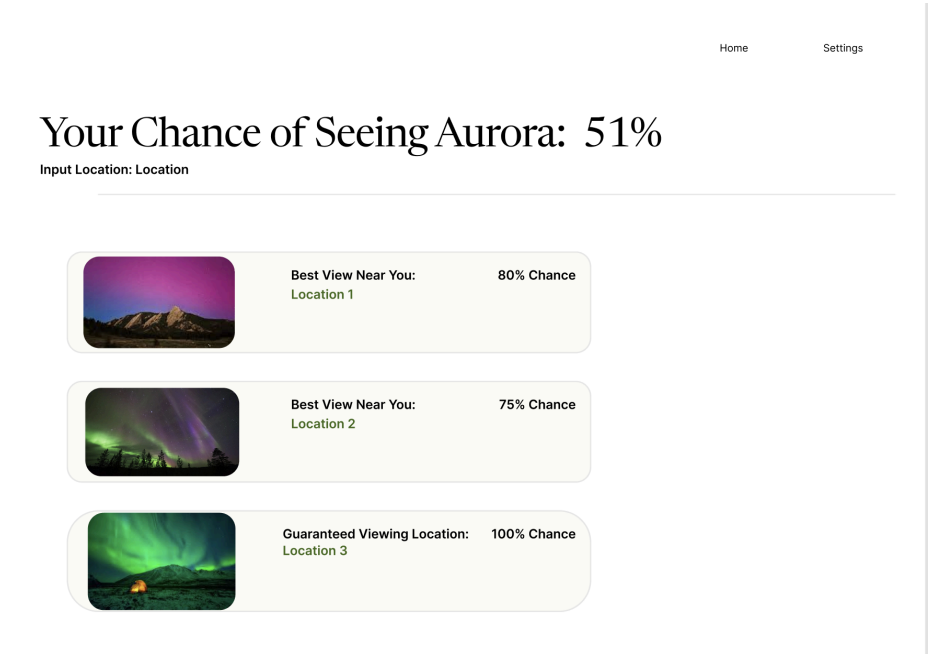


social.hbs -

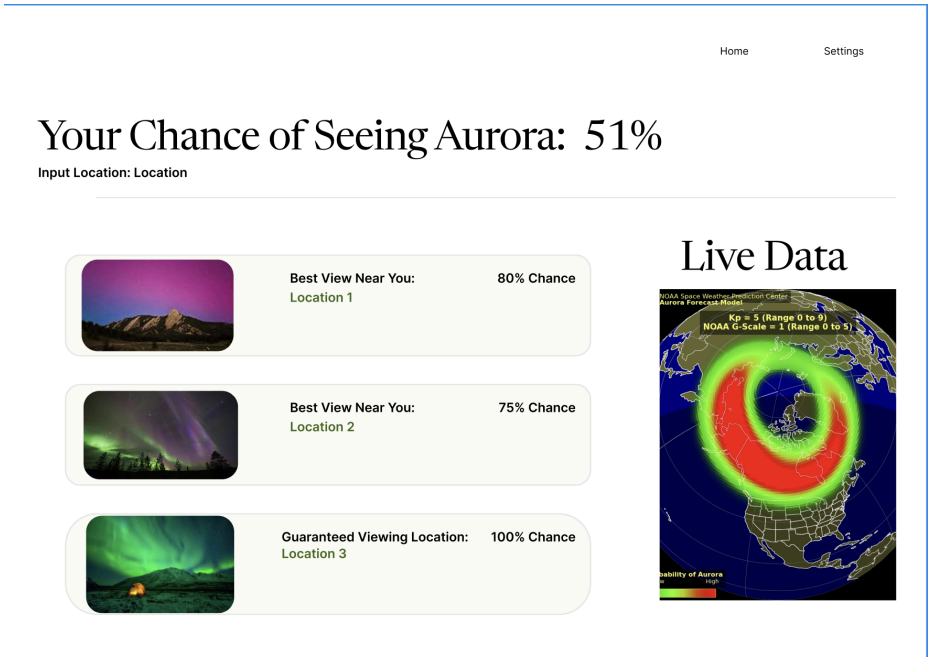

profile picture
@username
Picture of aurora
PP
@username
Picdure of aurora
feed
finder button
profile picture

finder.hbs -

Wireframe 1:

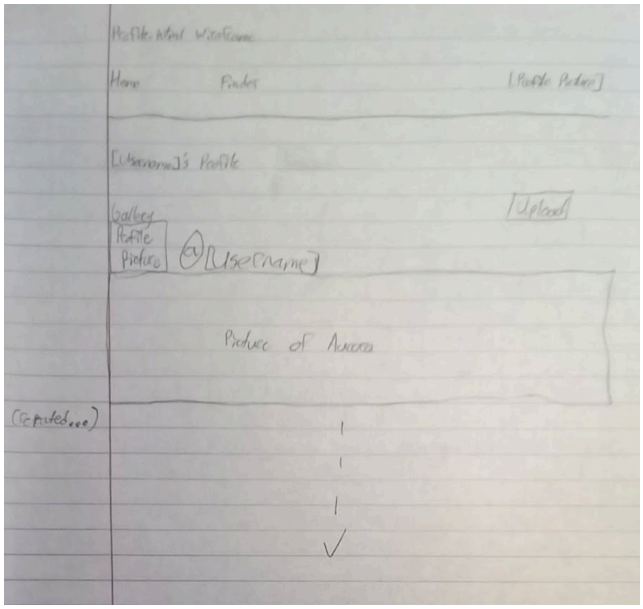


Wireframe 2:

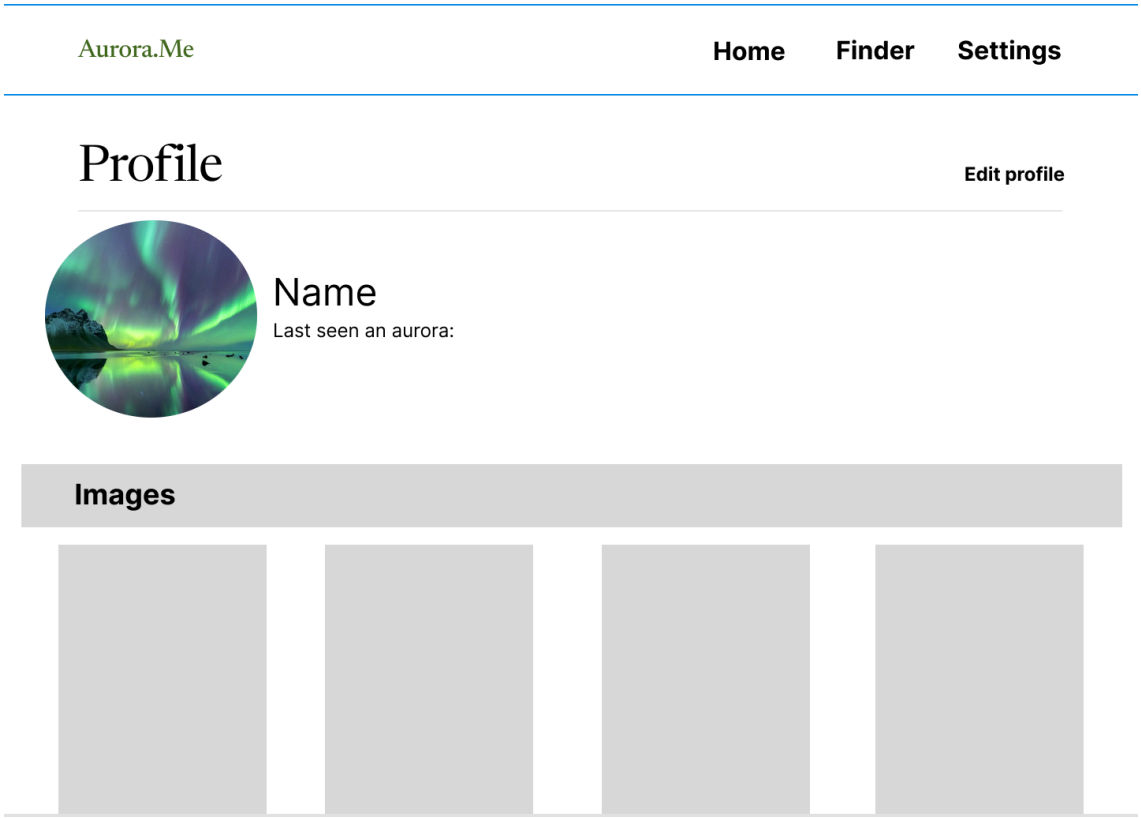


profile.hbs -

Wireframe 1:



Wireframe 2:



Test Results:

For our testing we have made 4 broad tests using Chai to observe what happens with our code when users follow certain, expected processes. In our first broad case, we have a positive and negative unit test case respectively to see how the application reacts when a user is registered using valid and invalid elements for our tables. The unit test that passes through completely valid values for img, username, password, etc. will pass as expected while the unit test case that tries to post mismatching date type values to the server will fail. Our second use case works to see if a user logging in to the website will be correctly redirected to the profile page after logging in. A valid user is sent with the login POST route and the app ends up getting and rendering the profile page as it should. Our third use case checks on our logout functionality and observes the typical user logging in with valid credentials, creating a session that is then destroyed with the logout route. Ensuring that our logout routes are working is important for keeping people's information safe. Our fourth use case implements positive and negative unit test cases to make sure the application is not allowing people who are not logged in to view the social page. The positive unit case passes when a logged-in user accesses the social page, and the negative unit case passes when users who are not logged in are redirected from viewing the social page.

Deployment: https://aurora-me.onrender.com/