Powder Junkie Project Report

Contributors:

Akshay Patnaik (Github: AK8506)

Lance Kluge (Github: lance-kluge)

Josh Huang (Github: joshh9)

Julia DiTomas (Github: JuliaDiTomas)

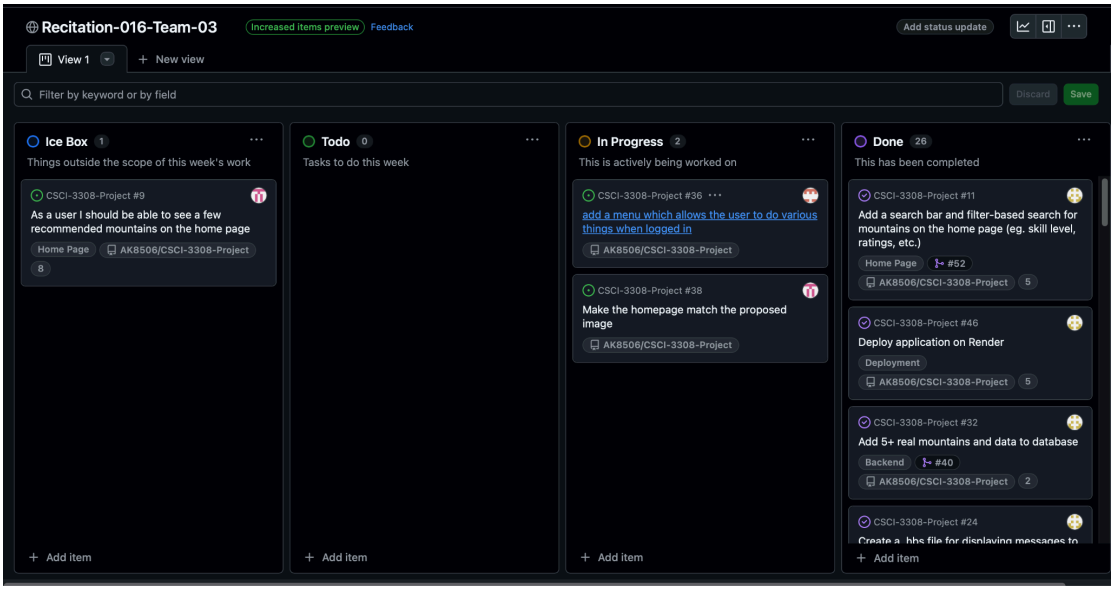
Thomas Parameswaran (Github: Greninja56)

Description:

Powder Junkie is a ski resort review app that provides real-time, crowd-sourced updates on mountain conditions and terrain quality. It is for skiers and snowboarders who want to explore new mountains, to find the best resort to visit each day, and to share their experiences and opinions with the world. Unlike traditional review platforms with static information, Powder Junkie ensures users make informed decisions on the go and get the most out of the ski season. It is also able to serve as your own mountain tracker so that you can look back at your own thoughts and ratings for each mountain that you visit. That can enable you to better keep track of your thoughts on different mountains.

GitHub Project Board

(https://github.com/users/AK8506/projects/2)



DEMO:

csci3308 project demo.mp4

(https://drive.google.com/file/d/15QYEUHMiGYmWTFqgYYtOtLVVc7hkVREx/view?usp=sharing)

VCS / Github Repository:

(https://github.com/AK8506/CSCI-3308-Project)

- Source Code
- Test Cases
- Video demo
- README.md in GitHub
- Project documentation
- Project Board

Contributions:

Lance Kluge:

Lance developed the individual mountain page and the homepage (excluding the search filter) using Handlebars, HTML, Bootstrap and Node.js. Lance also implemented the manage reviews page including updating each mountain's average rating when reviews

are created/deleted. For image storage Lance implemented Multer to handle uploads and integrated them with the database to ensure proper displaying of images across the app.

Akshay Patnaik:

Akshay developed the initial routing for every page (homepage, login registration) and the ability to create reviews in the back end using Node.js. Akshay also created the message partial, which displays confirmation or error messages to the user using Handlebars, implemented in Node.js code. Akshay manually added sample mountains from Colorado with real data to the database using PostgreSQL, pre-loaded on the homepage. On the homepage, Akshay created a search bar and filter-by-rating options for users to look up mountains using Node.js, HTML, and Bootstrap. Akshay also deployed the application with a remote PostgreSQL database on Render.

Josh Huang:

Josh developed the homepage and registry page and built the linkage between the pages and some of the organizational aspects of the page. Using handlebars, HTML, CSS. For the registration page Josh made it so that users could create accounts and that would be saved allowing them to then log in with the credentials they just created.

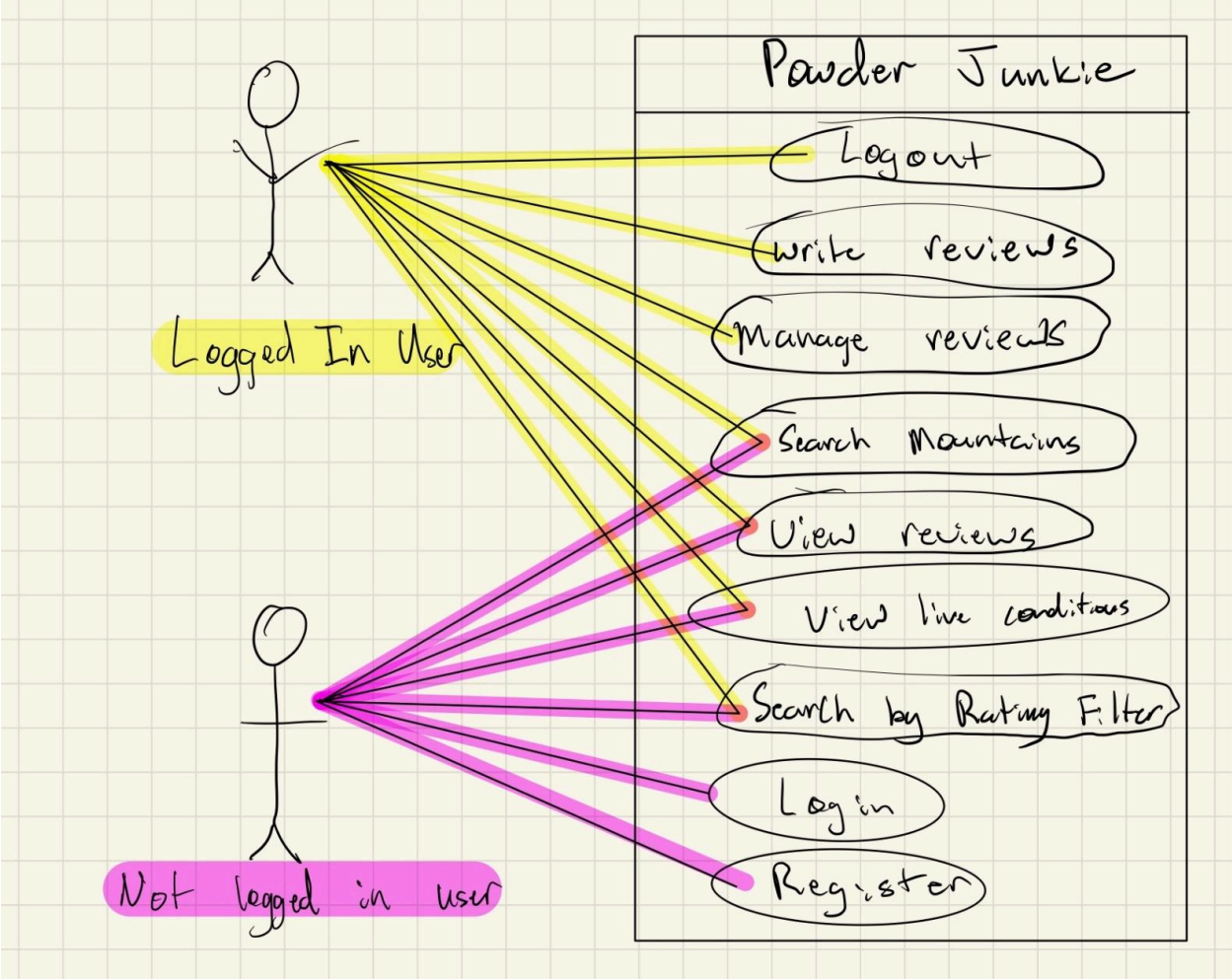
Julia DiTomas:

Julia implemented the retrieval and caching of weather observation and forecast data, using SQL for our database and axios for the external National Weather Service API. To ease the addition of new mountains and stay up to date with NWS mapping changes, she added functionality to initialize and periodically update this data for all mountains.

Thomas Parameswaran:

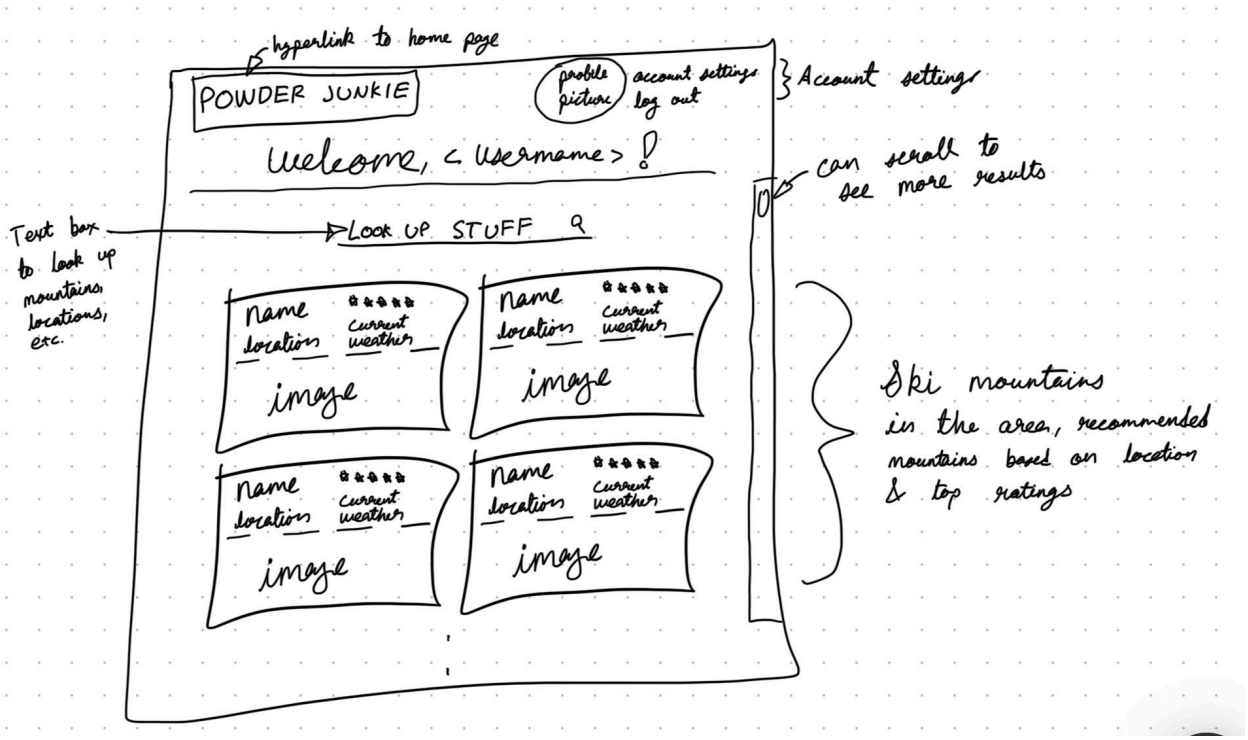
Thomas developed some aspects of the navigation bar using Handlebars and HTML, specifically page routing. Thomas also developed signs of a user being logged in, like displaying the username of who is logged in.

Use Case Diagram:

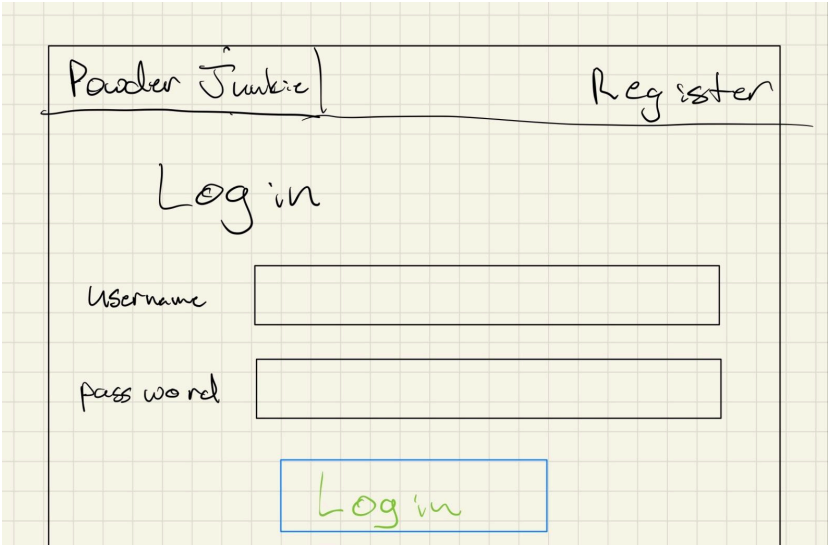


Wireframes:

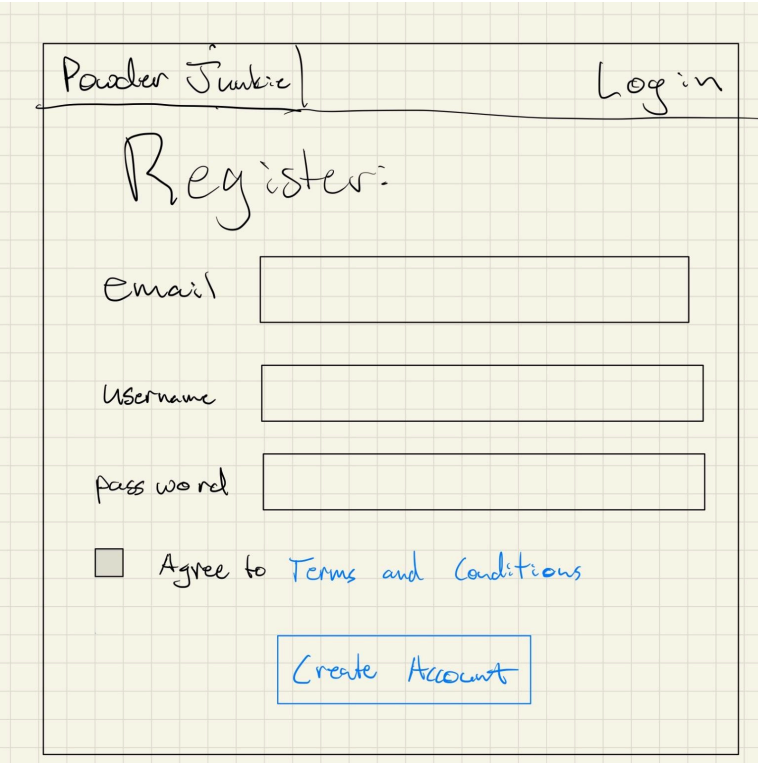
Home:



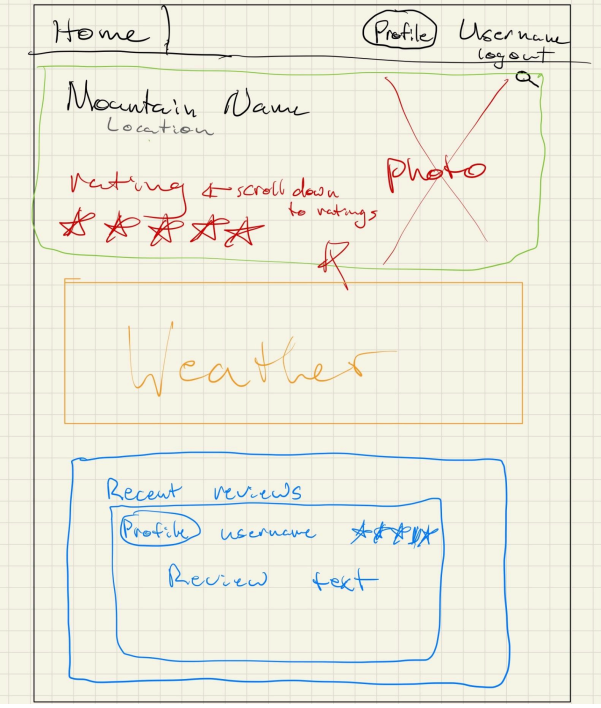
Login:



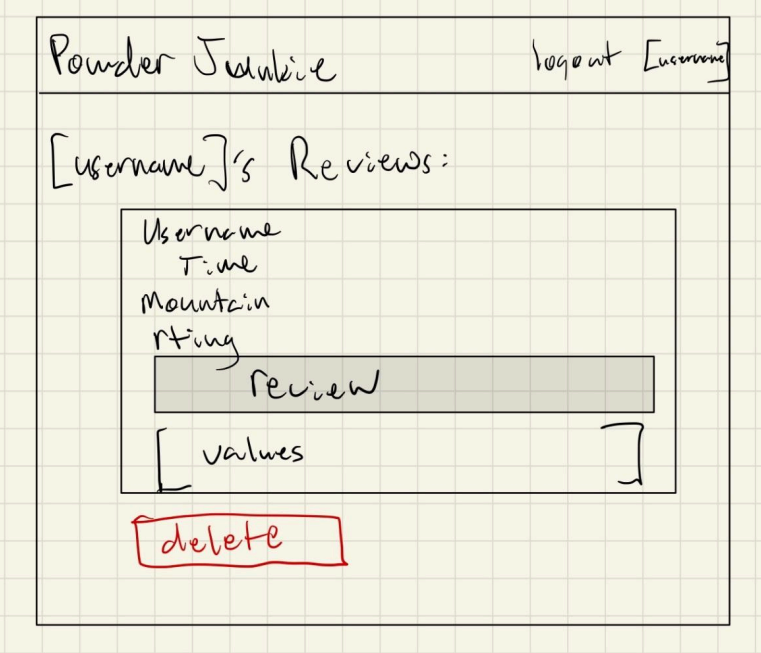
Register:



Individual Mountain Page:



Manage Reviews:



User Acceptance Tests:

- Test 1: Test subject was prompted to create an account. She clicked the 'Register' button, entered her name, email, and password, and hit Submit. She faced no issues (such as invalid email or mismatched passwords). Her behavior was as anticipated.
- Test 2: Test subject was instructed to read some reviews. She chose Copper Mountain and read the existing reviews.
- Test 3: Test subject was asked to look at the weather forecast for a mountain and decide if it would be good this weekend. She looked at Copper, and concluded that although it will be windy, it will be plenty warm.
- Test 4: Test subject was asked to write a review for any mountain. She selected Winter Park, wrote her review, and put in ratings for each category. She asked if she could do half-stars (which she could and did). She did not choose to upload an image. Her review was posted successfully, and then she scrolled down the page to find it.
- Test 5: Test subject was asked to delete her review. At first, she looked for somewhere to delete it where it shows up on the Winter Park page, but then realized that it wasn't there and clicked on 'Manage my reviews'. She then hit the

delete button and the confirmation button. She then observed that her page had no reviews.

Finally, test subject was asked if she thought the website was user-friendly, and she said that she did. Many thanks to Anna for agreeing to test our website.

Deployment:

The application is deployed on Render, using a web service to host the frontend and a PostgreSQL service for remote data storage.

Link to remote application