

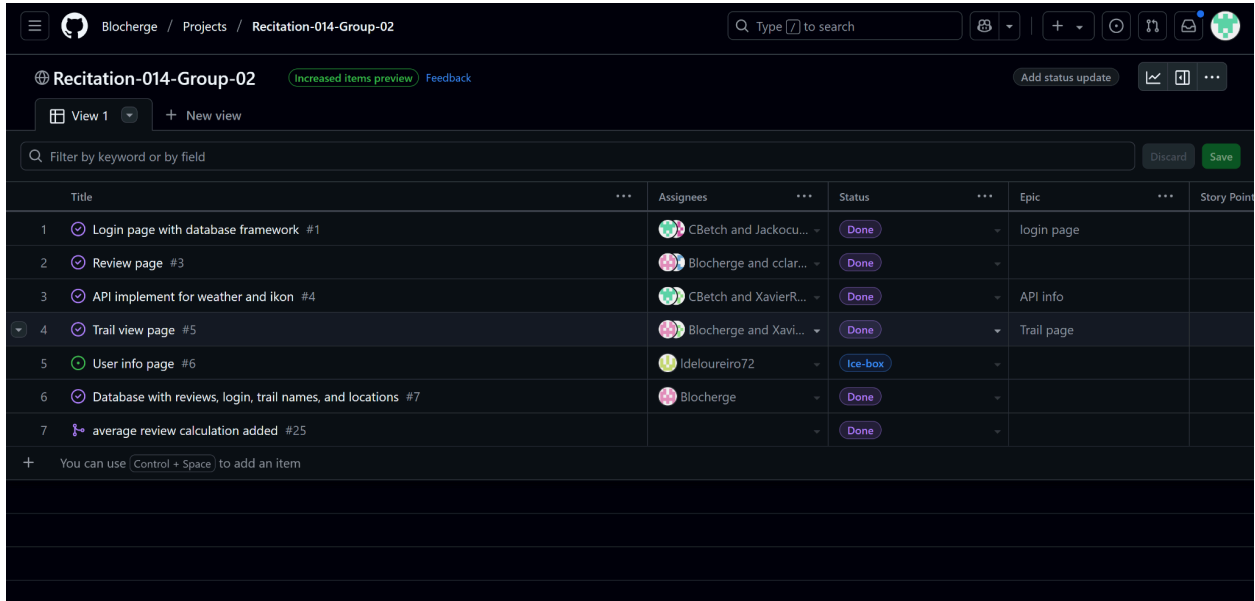
Liftability

Caleb Bettcher
Graham Blocher
Cole Clare
Jack Coan
Logan DeLoureiro
Xavier Rudnick

Project Description

Our project, the website Liftability, was built to provide skiers in Colorado with the Ikon pass up-to-date and important information in a concise and user friendly form. Our website provides users with information regarding weather, lift access, run information and status including difficulty and grooming, crowdedness, and reviews for the most popular Ikon pass destinations across the state. We currently support four Ikon mountains here in Colorado: Steamboat, Copper, Eldora, and Winter Park. For each mountain, the home page will display a basic weather readout and snow report, with links to access more information. Each mountain will also have its own page with more information. This includes a trail map, link with directions to reach the mountain, and information about all of the trails and runs on the mountain. From this page, users will easily be able to access information about the status of their favorite chairlifts and runs: Is it open? Is it groomed? How crowded is it? Additionally, users have the option to leave a review of a mountain in order to provide other users with insights as to the busyness level, overall experience, and mountain rating (which will also be displayed on both the home page and individual mountain pages). We hope users will have a positive experience with Liftability and will be able to receive up to date, and accurate information on the status of their favorite mountain without having to visit several different websites!

Project Tracker - [Link](#)



	Title	Assignees	Status	Epic	Story Point
1	Login page with database framework #1	CBetch and Jackocu...	Done	login page	
2	Review page #3	Blocherge and cclar...	Done		
3	API implement for weather and ikon #4	CBetch and XavierR...	Done	API info	
4	Trail view page #5	Blocherge and Xavi...	Done	Trail page	
5	User info page #6	Ideloureiro72	Ice-box		
6	Database with reviews, login, trail names, and locations #7	Blocherge	Done		
7	average review calculation added #25		Done		

Video - [Link](#)

Git Repository - [Link](#)

Contributions

Caleb Bettcher

I worked heavily with the API implementation and populating the database with information regarding lift and run status for each of the different mountains. Data is pulled from each of the mountain's webpages and converted to a standardized csv format. I also implemented the script to convert these csv files to SQL code which populates the db upon initialization. I ran into some issues; particularly with run difficulty, where different mountains had different difficulty scaling, but with the use of a couple of standardization algorithms, all of the run difficulties are converted to 1-5 which we then use to load a difficulty icon (green, blue, etc.) on the website. In terms of the front end, I built the drop downs which display lift and run information on each of the mountain pages (including visual icons to improve experience), and provided assistance to my teammates where it was needed.

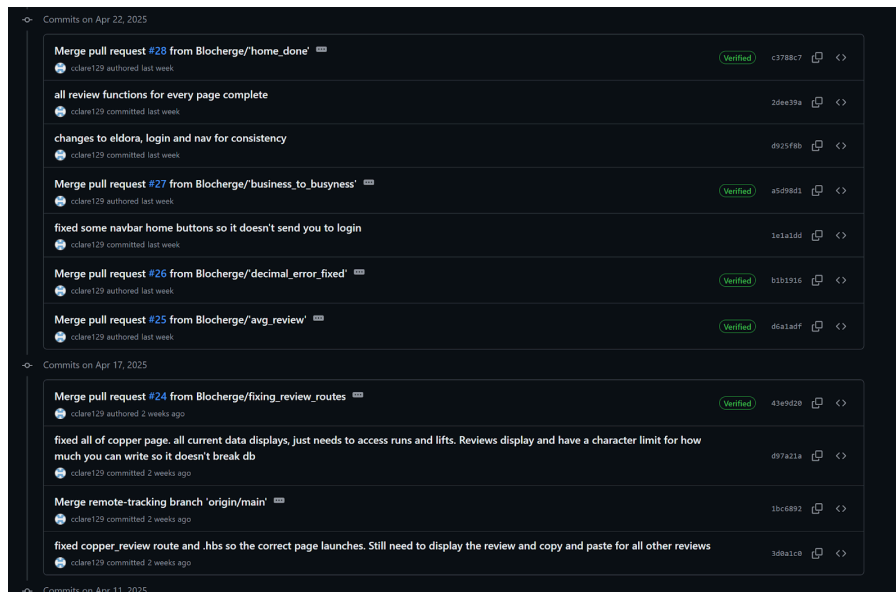
Graham Blocher

I worked with a lot of the base structure of the website, I started with the .hbs and file layout, then setup the base SQL build for the trails, reviews, login, and mountain specific pages. I then used the index.js routes for the base login as well as the database connection from lab 8.

I further built all of the index routes for the home page and each of the mountain pages on boot and built the review routes so that they can take reviews, recalculate the average review, and will update an existing review instead of allowing the user to post multiple reviews.

Cole Clare

Worked on fixing the database, routes, SQL errors, etc. I put most effort into the user reviews and those routes. I did a lot of the handlebars, sending the data between the HTML forms and JavaScript, making sure the routes' queries were correct, calculating averages and sending that data, modal work and some navbar changes.



Jack Coan

I worked on a lot of different things throughout this project. One of the main aspects of our website was to implement the login and register pages so that users were able to create an account on the platform and save data for better usability and continuity. This was not too bad to implement as I could use previous labs to base off of. There was a challenge in that when creating an account, after inputting the necessary info, the site would crash but that was ironed out. I also helped out on the look of the site by sourcing the images for each mountain to give the user a better experience.

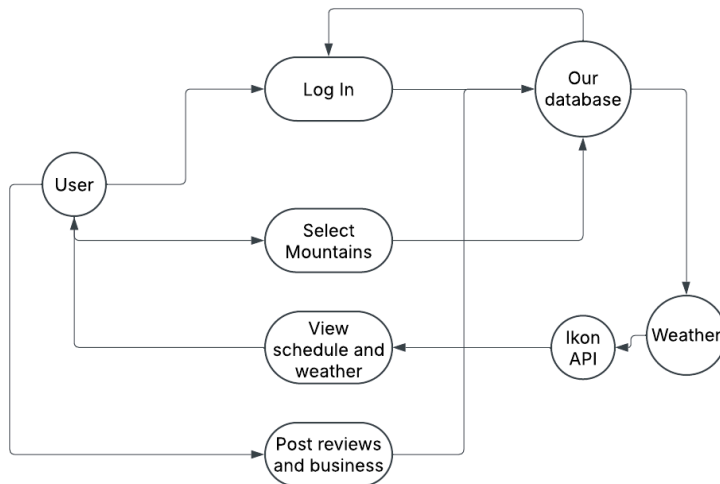
Logan DeLoureiro

I focused more on the front end of the app, helping develop wireframes, html, and bootstrap. I helped populate the various pages with the data from the different API's we used. I worked on navigational routes and focused on improving our pages as our scope and vision for the project changed. Toward the end of the project I tried to help focus more on organization, focusing more on the presentation, layout, and rubric to help make sure we were in a good place as group members finished the last technical parts of the project.

Xavier Rudnick

I worked on the backend of the app by finding and populating the weather data for each mountain. The first objective I had was finding a free weather API, I ended up deciding to use Open Meteo. I decided on using Max and Min temperature of the day, wind speed, snowfall, and UV. temperature and wind speed are important when deciding what to wear, snowfall is important because the more snow, the more enjoyable skiing experience, and UV index impacts if you will need to wear sunscreen or not on the mountain. I had a script grab the data, then store it into a csv to be imported into the database. I also was in charge of setting up the cloud hosting of the web app, which ended up being on render.

Use Case Diagrams



Wireframes

3. Confirm that for each mountain, summary details (ratings, temperature) are visible.
4. Click the arrow for one mountain's dropdown and verify that additional details are revealed.

Expected Results:

- The user should see all four mountains with their summarized details.
- The dropdown should expand when the arrow is clicked, revealing more information clearly (and collapse upon clicking again).

Observations:

- What are the users doing? Users are navigating to the homepage, scanning the mountain information, and experimenting with the dropdown arrows to view more details.
- What is the user's reasoning for their actions? They want a quick overview of mountain conditions and are checking if they can access more in-depth info about each mountain without having to leave the homepage.
- Is their behavior consistent with the use case? Yes. Users naturally use the dropdown arrows to get more details. However, a few users sometimes tap the entire bar instead of the specified arrow.
- If there is a deviation from the expected actions, what is the reason for that? The deviation seems to be that the clickable area isn't prominent enough; users may not immediately recognize that only the arrow triggers the dropdown.
- Did you use that to make changes to your application? Based on this observation, a potential change is to adjust the UI—probably by increasing the clickable area.

Use Case 2: Navigating to an Individual Mountain Page for Detailed Information

Description: A user interested in a specific mountain (e.g., Copper) clicks on the “More Info” link to view detailed information. This page provides comprehensive data, including weather readouts, a trail map (displayed as an image), and dropdowns for the lifts and runs.

Test Steps:

1. Click the “More Info” link for a chosen mountain.
2. Verify that a new page loads with detailed information specific to that mountain.
3. Confirm that data such as weather details, run statuses, and other information (like user reviews) is clearly laid out.

Expected Results:

- The detailed mountain page loads correctly, displaying the trail map, weather information, run and lift statuses, and review sections as configured.

Observations:

- What are the users doing? Users click on the “More Info” link for their mountain of interest and review the detailed page that loads.
- What is the user's reasoning for their actions? They want to see more in-depth data (such as updated conditions and detailed run information) beyond the simple summary provided on the homepage.
- Is their behavior consistent with the use case? Yes. Most users follow the link to get additional details, confirming that the page's navigation is functioning as intended.
- If there is a deviation from the expected actions, what is the reason for that? A few users mentioned that the “More Info” link could be more prominent, as they occasionally missed it on the homepage.

- Did you use that to make changes to your application? The feedback suggests considering design adjustments—such as enhancing the visual emphasis on the “More Info” button—to improve clarity (which we did by changing the standalone color).

Use Case 3: Submitting a Review on a Mountain Page

Description: A user wants to share their experience of a mountain by submitting a review. The process involves entering review text and a rating, then submitting it. Once submitted, the review is stored in the database and displayed on the mountain page, updating overall mountain ratings.

Test Steps:

1. Navigate to the individual mountain page (for example, Steamboat).
2. Locate the review submission section.
3. Enter a valid review (rating and feedback text).
4. Submit the review.
5. Verify that the review appears on the page and that aggregate ratings update if applicable.

Expected Results:

- The review is successfully submitted and immediately visible on the mountain page.
- The submission is stored in the database, and overall ratings reflect the new review.

Observations:

- What are the users doing? Users interact with the review submission form, typing in their rating/feedback, and then clicking the submit button.
- What is the user's reasoning for their actions? They wish to provide input on their personal experience at a mountain, which they believe will help others make informed decisions.
- Is their behavior consistent with the use case? For the most part, yes. Users are following the steps as designed. Some users check after submission to confirm that their review appears.
- If there is a deviation from the expected actions, what is the reason for that? A few users expected a confirmation message (such as “Review submitted successfully”) before seeing the review update. This may indicate a need for better feedback cues.
- Did you use that to make changes to your application? To address the confusion, the application could incorporate an immediate confirmation message using handlebars.

Use Case 4: Accessing the Interactive Drop Downs for Lifts and Runs

Description: A user interacts with the drop-down menus on an individual mountain page to view detailed information about lifts and runs—such as open status, grooming, and crowdedness. On these pages, the entire drop-down bar is clickable to expand or collapse the detailed information.

Test Steps:

1. On an individual mountain page, locate the drop-down area that contains lifts and runs information.
2. Click anywhere on the drop-down bar to expand it.
3. Verify that the drop-down expands to reveal accurate and detailed status information about the lifts and runs.

4. Click again on the bar to collapse the drop-down, confirming that the toggle functionality works as intended.
5. Repeat the process with multiple drop-down elements on the page to ensure consistent behavior.

Expected Results:

- The drop-down toggles its state correctly—expanding when clicked to reveal detailed information, and collapsing on a subsequent click.
- Detailed attributes of lifts and runs (e.g., open status, grooming level, crowdedness) are clearly displayed when expanded and properly hidden when collapsed.

Observations:

- What are the users doing? Users are testing the interactive drop-down feature by clicking on various drop-down bars on the mountain page, looking to see the additional lift and run details.
- What is the user's reasoning for their actions? They want to view more of the page at certain times and ensure that the detailed statuses of lifts and runs are easily accessible by interacting with the entire drop-down area.
- Is their behavior consistent with the use case? Users intuitively click on the full bar to toggle the information display.
- If there is a deviation from the expected actions, what is the reason for that? Most feedback indicates that users understand the entire bar is clickable.
- Did you use that to make changes to your application? Again, most users understand that the entire bar is clickable due to hover animations.

Deployment: <https://csci3308-014-02-group-project.onrender.com>