

Title: CU Snow Share

Who: Aloken Chaudhari, Austin Hardy, Anusha Venkateswaran, Holden Eckert, Ethan Sachs

Project Description: Our project is a platform for CU students who love to ski/ride to find other students who want to shred the mountains. Students are required to register with information such as email, password, name, ski/board preference, skill level, and favorite mountain. Once they are registered, students are added to the database so that other students can see their info when filtering by preferred tags on the discover page. After registering, students can login with their registered info and navigate to the discover, about, or profile pages. The profile page displays the students' preferences that they used to register. The about page gives a short description and photo of all members of the SnowSpices team. On the discover page, students can compare their preferences with others, which will display other students' preferences, including how many tags were matched to the current user. These tags ensure that students are matched up with those who share their preferences on the mountain. The discover page also displays contact information of the students who share tags with them, making it easy for the user to reach out and plan their next trip to the slopes!

Project Board/Tracker: <https://github.com/users/anushaven/projects/1/views/1>

VCS: <https://github.com/anushaven/CSCI3308-Team1-SnowSpices.git>

Video:

Contributions: Austin, Anusha, Holden, Ethan, Aloken

Austin: I worked mostly with forming the database for our website by using SQL, and I also made our separate Testing branch. I was able to connect the SQL tables so that users were able to see who they shared the most tags with on the discover page. Since we weren't able to make a res.status and res.redirect call at the same time, I also made a Testing branch that purely tested our /login and /register POST calls within index.js. After all of these tests passed, I made some finishing touches which involved rendering photos on the about page and I also made sure both newly registered and existing users could login without issues.

Anusha: I mainly focused on making the Discover page and deploying the site, including the site's EJS file and corresponding index routes. In the EJS, I wrote two functions, compareData() and findAndDisplayMatches(), that essentially compared the current user's ski_or_board and mtn_name attributes with those of all other users in the SQL table "tags", and on clicking a button, returned the usernames and email addresses of any matching users. In the index, I wrote the Discover route that rendered the page, GET "/discover", and the route that sent session information to the script functions in the EJS, GET "/api/discover/matches". A lot of docker and javascript troubleshooting were involved, but in the end I was able to reach the acceptance criteria of writing a functional, easy-to-use Discover page.

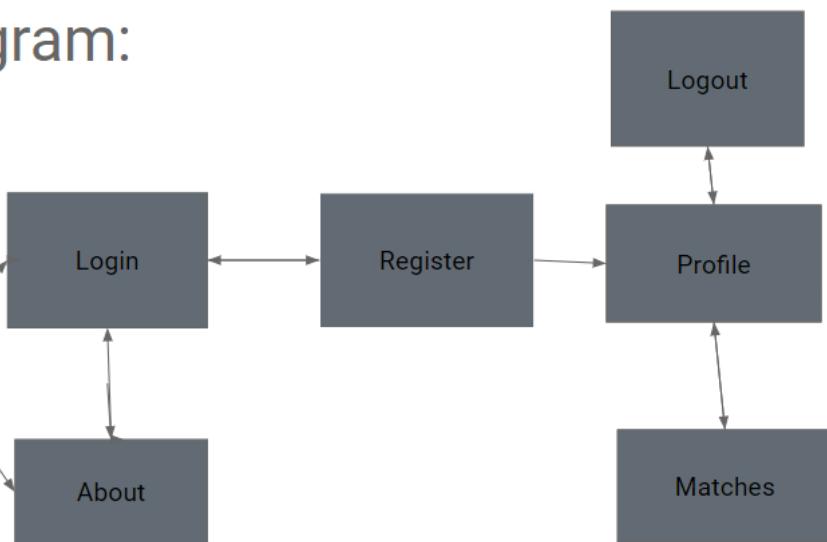
Holden: My main focus for our project was the CSS and the registration features of our website. This involved me implementing CSS into our website that made it easy to navigate along with creating a good user experience. For creating the registration page it involved me manipulating the registration.ejs and index.js page. I implemented code that would add the users inputs into our SQL tables so that the discovery page could use the database to match similar users. I also had to change our SQL files so that the information within the tables and the create tables aligned with the code for actually inserting the users information.

Ethan: I mostly worked as a full stack developer for this project, as well as a semi-scrum master. I was resolving issues with our NodeJS and ejs files, while also troubleshooting css issues and SQL tables. I implemented the original file structure and assisted the group in using github branches to keep our version control operational. I had to overcome issues with bcrypt and npm while also continuing to keep the group on track. Most of my time was spent in office hours troubleshooting issues that were impacting the entire team.

Aloken: The main focus of my work on this website were the about and profile pages. For the about page I mainly focused on the visual aspects. This page did not have to have information dynamically placed/taken from it, so it was fairly easy to implement. The most difficult part was likely figuring out how to work around the CSS and make it look nice, doing this took up most of the time which I spent on the about page. After this I focused mainly on debugging other parts of our code so that the docker and npm modules would work, however this was mostly a group effort. My main struggle came with the profile page, where the implementation should have been very simple and straightforward, as it simply displays data that is in the system. However for a big portion of the project I solely faced issues with the queries, while they worked for the rest of my group mates they did not work on my system. Which meant that I had to code the page and other group mates had to test it, which made this page exceedingly difficult to test once coded.

Use Case Diagram:

Use Case Diagram:



Deployment: <http://recitation-011-team-01.eastus.cloudapp.azure.com:3000/discover>

Note: As per the instructions at the end of lab 13, the Virtual Machine was shut down and all used resources for deploying our site were de-allocated. Below are screenshots of all critical pages of the website, that include the deployment link in the URL field.

The screenshots show the following content:

- Login:** A form with fields for 'Enter Username' and 'Password', and a 'login' button. The background is a snowy mountain landscape.
- About Us:** A page titled "About Us" with the sub-section "Our Team". It contains a paragraph about the project's goal to expand snow sport accessibility through matching.
- Discover:** A page titled "Discover" with a sub-section "Matched Keys". It includes a welcome message and a "Find SkiBoard Partners" button.
- Register:** A form for creating a new account. Fields include 'Username', 'Password', 'Email address', 'Name', 'Mountain Name', 'Skill Ratings (Green, Blue, Black, Double Black)', and 'Ski Level'. The background is a snowy mountain landscape.

Test Results:

Note: For testing, please checkout the Testing branch in our GitHub repository, rather than the main branch.

Use case 1: Positive login test

In our first case, we tested our POST /login API by inputting an existing user in our students table to check whether our database was connected to login properly. Since username: 'icsp' and password 'munch' are in the students table, the test passes, and prints "User login successful."

Use case 2: Negative login test

In this case, we are checking POST /login API by passing in an empty string for username and password, which is invalid and returns status 200. The test should pass and return the message "Incorrect username or password."

Use case 3: Positive register test

In this case, we are checking POST /register API by passing user info into the register page properly, including the correct formatting for email and all other fields. This test case should pass and return a status 200 along with a "Registration successful." message, since all fields were inputted by the user.

Use case 4: Negative register test

In this case, we are checking POST /register API by passing empty user info in an incorrect manner (no fields can be empty on registration). This test case should pass and return a status 200 along with a "Invalid registration." message, since all fields were not inputted by the user.