

Title: PassPair

Who: Sultan Alfoory, Zackary Gacnik, Jake Niedzielski, Owen Flack, Sawyer Reinig, Sunyojita Rattu

Project Description: PassPair is a website designed to connect skiers and snowboarders based on the ski pass they hold and their available dates. Our platform allows users to meet new people who share their passion for winter sports, thereby creating a vibrant community of skiing and snowboarding enthusiasts. Upon logging in, users are directed to the Discover page, also accessible through the navigation bar. This page features a range of tools and resources to help users make the most of their winter sports experience. Users can search for ski passes from select Mountain Companies through links to the company's websites, and the Discover page will automatically update their current pass information accordingly. In addition, the Discover page provides a comprehensive list of ski resorts in Colorado, complete with key information about each one. This enables our users to make informed decisions about where to ski and to find like-minded individuals with whom to hit the slopes. PassPair integrates with various APIs to provide a seamless user experience. Our embedded Calendar API allows users to see which other users are planning to visit specific resorts and on which dates. This feature is particularly useful for individuals who are new to winter sports or who may not have a ski pass yet, as it enables them to connect with others and plan their ski trips with ease. In conclusion, PassPair is a valuable resource for winter sports enthusiasts of all levels. Our platform helps users connect with other skiers and snowboarders alike by helping them find new ski resorts to explore and plan their winter sports activities with ease.

Presentation:

[https://docs.google.com/presentation/d/1ohCiSuCG8_j5jpM6fEm3qz2j8WZ7udGjaSDbrQkwYTs
/edit?usp=sharing](https://docs.google.com/presentation/d/1ohCiSuCG8_j5jpM6fEm3qz2j8WZ7udGjaSDbrQkwYTs/edit?usp=sharing)

Project Tracker- GitHub project board: <https://github.com/users/SawyerReinig/projects/1/views/1>

The screenshot shows a GitHub project board titled "SoftDev Project Board". The board is divided into four columns: "Icebox", "In Progress", "Testing", and "Done".

- Icebox:** Contains two items: "SOFT-DEV-TEAM-011-07 #11" and "SOFT-DEV-TEAM-011-07 #10".
- In Progress:** Contains one item: "SOFT-DEV-TEAM-011-07 #11".
- Testing:** Contains one item: "SOFT-DEV-TEAM-011-07 #11".
- Done:** Contains eight items:
 - "SOFT-DEV-TEAM-011-07 #14 SQL Databases"
 - "SOFT-DEV-TEAM-011-07 #17 Calendar Front End"
 - "SOFT-DEV-TEAM-011-07 #7 Filter Page"
 - "SOFT-DEV-TEAM-011-07 #18 Calendar API"
 - "SOFT-DEV-TEAM-011-07 #19 Discover Front End(Functionality)"
 - "SOFT-DEV-TEAM-011-07 #15 Adjusting Login/Register to fit UI standards"
 - "SOFT-DEV-TEAM-011-07 #8 Username and Password input"
 - "SOFT-DEV-TEAM-011-07 #21 Login Page"

Each card includes a link to the GitHub issue and a "Discover Page" button.

Video: https://youtu.be/lpYEuR2_h0

VCS: <https://github.com/SawyerReinig/SOFT-DEV-TEAM-011-07>

Contributions:

Sultan Alfoory - Created the folder structure, the SQL files and tables. Made a filter that filters by what pass they have, helped with the general theme we are going to be making our theme around, working on creating a logo. Created a script that helped with the filter function, added more json cards to the list of cards.

Zackary Gacnik - I focused on the front end specs of our website, developing the styling and CSS side of the project. I mainly incorporated Bootstrap 5.0 classes while doing so, but also researched other ways to incorporate styling such as background images, tab icons and importing Google Fonts to help our page fit out project's theme and look more user friendly.

Jake Niedzielski - I worked on testing for each of the views and their posts. I added the google calendar for public use, as well as the passpair button that uses a matching function based upon other users and your own passes. This function pulled information from the database and compared it to the current user such that you'll never match yourself and only match those with common passes. There was a bit of design done on my part in the calendar page such that the matching button displays information in an intuitive way as well.

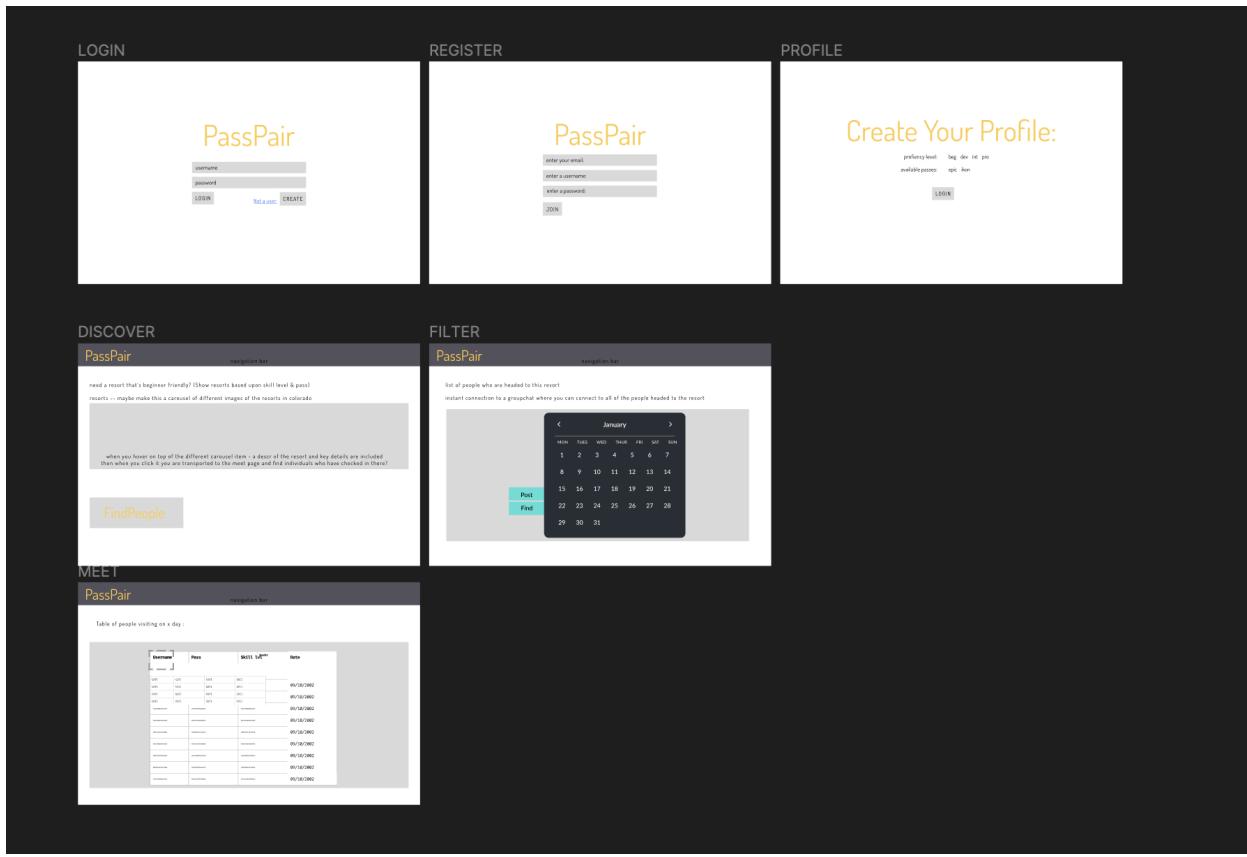
Owen Flack - I built a lot of the back end and basic front end for the register, login and discover page. Also altered the SQL into its final state that we end up using. Furthermore I fixed the rendering of the login page and actually collected some of the data on the resorts we would be using and displaying on that page. Finally I fixed the rendering of the discover page that was discussed in the slide show with the filter and cards not displaying correctly, but I didn't build the filter from the ground up.

Sawyer Reinig - Helped Jake with using the google API to make the calendar. I made the file structure for the project. I went to office hours to get the project to run on not just Owen's computer. I made the video showcasing the website. Worked with Zack to do the styling for all the pages.

Sunyojita Rattu - I built the initial use-case diagram and wireframe. I worked on our automated test plans. I worked on the nav-bar and made additions to the discover and profile page. I constructed test API's for lab 11. I completed lab 13 for my group and committed the documentation required with Microsoft Azure and launched our website. I also helped finish the last steps of the Project including the report and presentation.

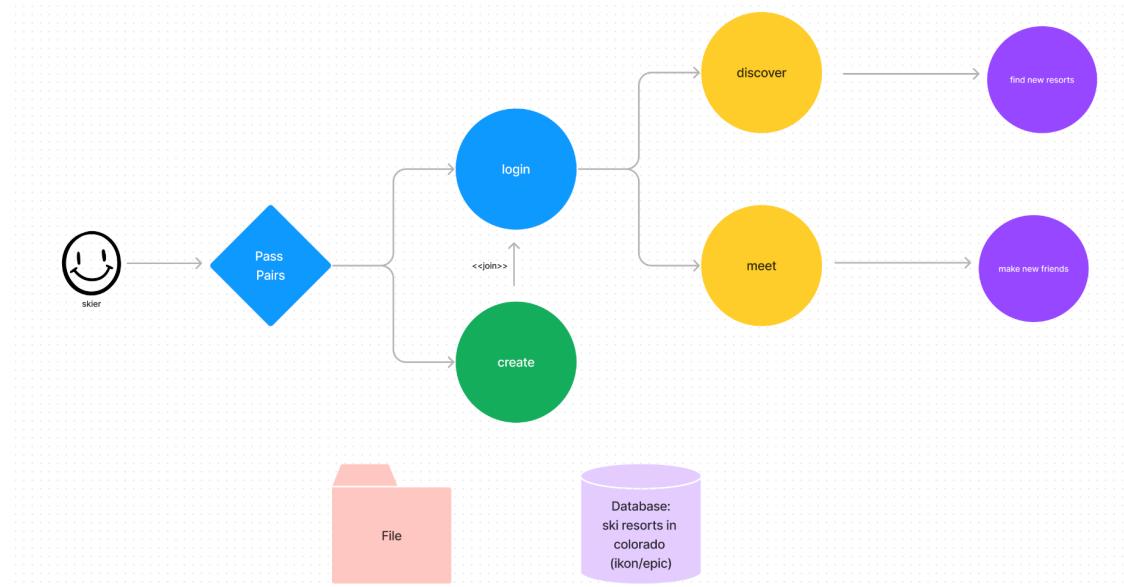
Wireframe:

<https://www.figma.com/file/EOIHc3liT2NJuORLBVfuTI/PassPairs?node-id=0-1&t=M4MRHeUi9hKcpbUd-0>



Use Case Diagram:

<https://www.figma.com/file/EOIHc3liT2NNUORLBVfuTI/PassPairs?node-id=0-1&t=M4MRHeUi9hKcpbUd-0>



Test results:

Through rigorous testing of the login POST api within server.spec.js, we were able to successfully gain access to the page by sending a request to the server for the /login page. Our testing process was not limited to mere access, as we thoroughly considered and tested various test cases to ensure the seamless functioning of our login API.

As part of our testing process, we made sure to test logging into an existing account, as well as an invalid login attempt to evaluate the error-catching mechanism of our API. Our negative login API test case was particularly impressive as we were able to successfully throw a 'null' value from the login function, which prevented a reroute to the discover page. This is particularly significant because it highlights the security measures we have put in place to safeguard our users' personal information, including their location and contact details.

On the other hand, our positive login API test case was equally noteworthy. We made use of a dummy variable stored within our database, which was then thrown as part of the login function. The result was a successful login attempt, which seamlessly rerouted the user to the discover page with all the desired features appearing as expected.

By carrying out such extensive testing, we were able to ensure the robustness of our login API, and guarantee the utmost security of our users' data.

After performing a comprehensive test of the register POST API within server.spec.js, we were able to successfully access the /register page. Additionally, we meticulously considered various test cases, such as registering an account with valid credentials as well as attempting to create an account with invalid credentials. We are pleased to report that all tests passed with flying colors and the system rerouted to the /register page as expected.

Our team also made sure to implement an error catch for usernames and passwords containing invalid characters, including extraneous characters such as "!@#\$%^&*". When these characters were detected, the system displayed an error message and redirected back to the /register page. By providing a resulting body message for each test, we were able to verify the correct functioning of the system.

For instance, in our negative register API test, we intentionally threw a null response, which returned successfully. As a result, the user was prompted with an 'invalid response' error message, alerting them to the issue and allowing them to take appropriate action.

Overall, we are confident in the successful implementation and testing of the register POST API and are pleased to report that it met our expectations in terms of functionality and reliability.

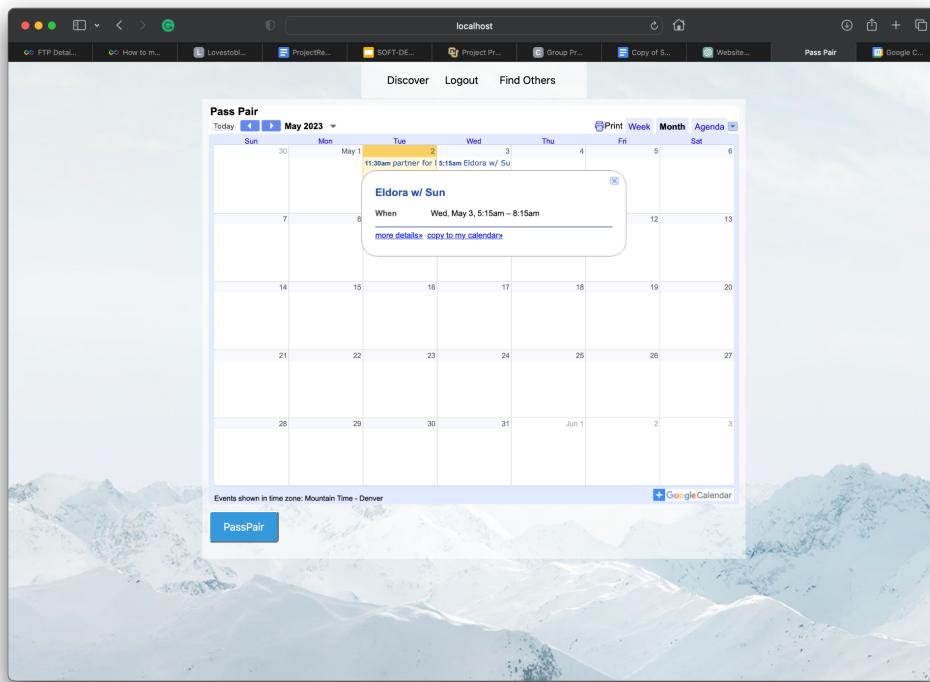
Our team tested the discover POST api within server.spec.js and achieved successful results. We focused on accessing the discover page and receiving a confirmation message after logging in. Within the Discover page, we featured several components such as a table of ski resorts located in Colorado with relevant features listed, a navigation bar that takes users to both the Discover page and the calendar page, and a filter option allowing users to sort through the ski resorts based on their specific needs.

To properly test the functionality of this feature, we had to incorporate a ski resorts database into our project code and ensure that we accessed it correctly. This allowed users to learn about and select the ski resort that aligned with their needs the most. We included several filter options, such as city and elevation, that were included within our resorts database.

Our testing process involved logging in successfully and testing each filter option and/or combination of filters to ensure that the correct ski resorts were displayed. We conducted our testing within our localhost docker container and documented the results with each other. Through this process, we were able to confirm that the discover POST api functioned as intended and provided users with accurate and relevant information regarding ski resorts in Colorado.

Deployment:

<http://recitation-011-team-07.eastus.cloudapp.azure.com:3000>



localhost

FTP Det... How to... Lovesto... Project... SOFT-D... Group... Pass Pair Google... Home -...

Discover Logout Find Others

Welcome, rat!

Proficiency Level:

Beginner
 Intermediate
 Advanced
 Pro

[Save](#)

Pass	Purchase Page	Has Pass?
IKON	You already have this pass!	<input type="button" value="No"/>
EPIC	Get EPIC	<input type="button" value="Yes"/>
Indy	Get Indy	<input type="button" value="Yes"/>
Mountain Collective	Get Mountain Collective	<input type="button" value="Yes"/>

All

Discover Logout

rat

Proficiency Level:

Beginner
 Intermediate
 Advanced
 Pro

Save

Pass	Purchase Page	Has Pass?
IKON	Get	Yes
EPIC	<i>You already have this pass!</i>	No
Indy	Get Indy	Yes
Mountain Collective	Get Mountain Collective	Yes

All



Winter Park
Winter Park, recently voted #1 Ski Resort in North America by USA

[Learn More](#)



EPIC TITLE
Epic's Resort

[Learn More](#)



INDY TITLE
Indy's Desc

[Learn More](#)



MOUNTAIN TITLE
Mountain's Desc

[Learn More](#)