014-2 Project Report

Title: Scuuber

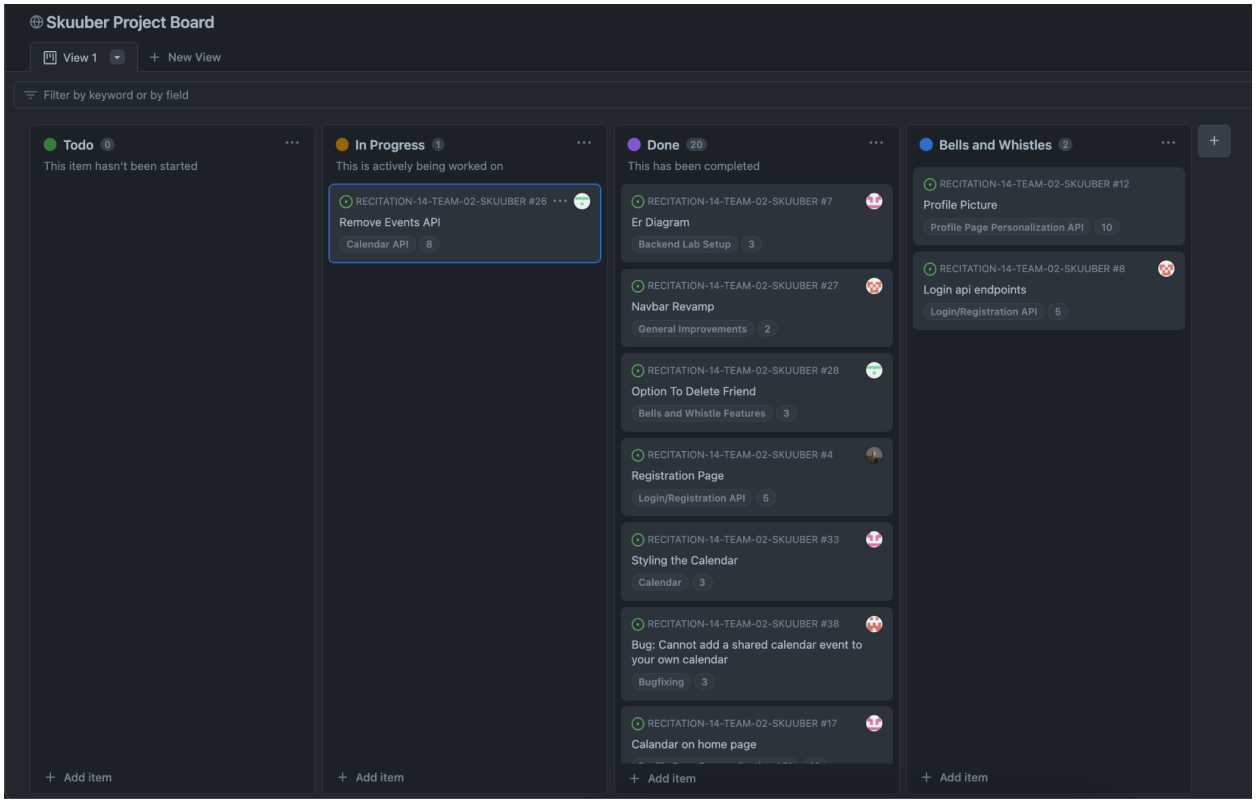
Who: Andrey Lototskiy, Sarah Bian, Ryan Foley, Harley Ewert, Oscar

Project Description:

Scuuber is a way to manage your ski trips with your friends and family. A common problem faced by skiers is the inability to efficiently coordinate and plan rideshares for ski trips. Scuuber, a cutting-edge web app, addresses this issue by providing a collaborative calendar platform for ski enthusiasts. Developed using ejs, node.js, and PostgreSQL, Scuuber streamlines ski trip planning and organization for groups of friends and family. The app offers a user-friendly interface, allowing participants to easily schedule and join rideshares, while keeping track of important trip details such as dates, times, and pickup locations. Scuuber's calendar view enables users to visualize their ski trip plans and make necessary adjustments in real-time. The platform also supports adding friends, fostering a sense of community among ski trip participants.

Link to Project Tracker: https://github.com/users/harley616/projects/1/views/1

Screenshot showing our project in your project tracker:



Recording of our Presentation:
https://drive.google.com/file/d/1sWg0M4AKXRXLkAwBFtAuT0_C5Q75i98U/view?usp=sharing

VCS: Link to our git Repository:
https://github.com/harley616/RECITATION-14-TEAM-02-SKUUBER

Contributions —----------

Harley616's contributions: Harley616 worked on implementing the weather api, this included a weather search to pull up a forecast for a certain area, as well as a popup which contains weather info for the day and location of an event. Integrating the api required making separate post routes with express.js for the pages which used weather, and html and css were needed to display the weather info. Ejs was used to implement logic and pass data. He also worked on bug fixing which arose when merging branches, as well as general styling for the site.

Lototskiyandrey's contributions: Lototskiyandrey focused on updating logs, fixing the friends' calendar, and addressing minor bugs and visual issues. They contributed to improving the UI for accepting friend requests, styling the sending friend request box, and adding a script to the home screen. Lototskiyandrey also worked on updating the register and login pages, creating a new menu, and making UI improvements.

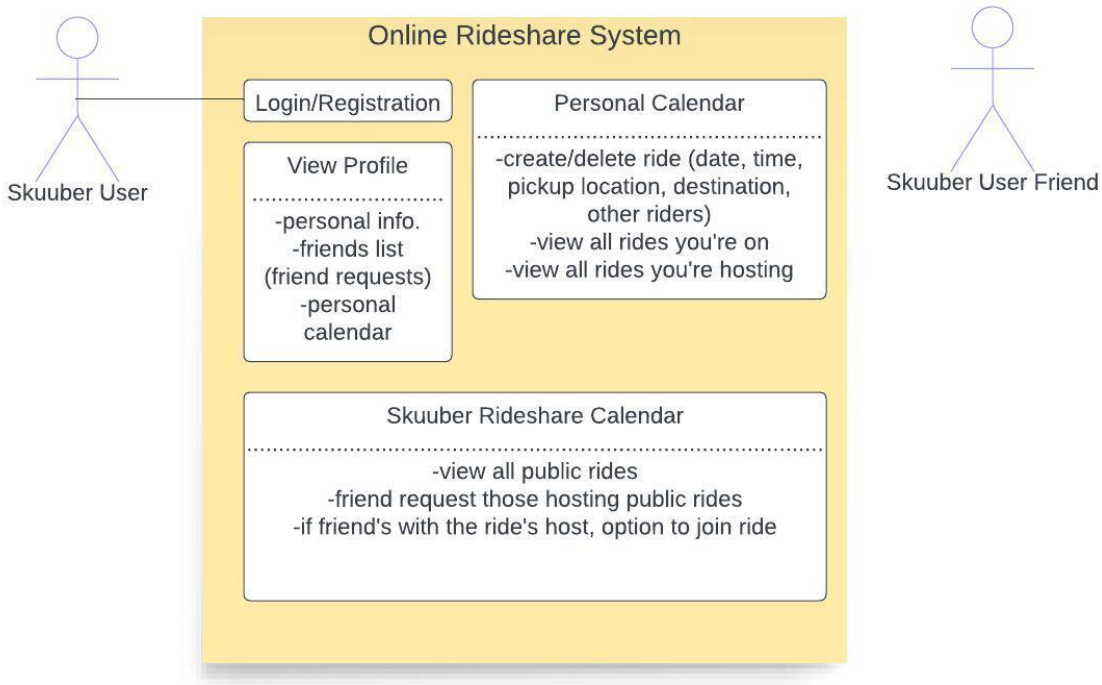
Rfoley0917's contributions: Rfoley0917's work included fixing the delete reload bug, fixing the confirm password bug, and contributing to the final fixes for the deleteEventAPI. They added a backend for updating events, created a target modal for updating events, and worked on a mergable deleteEventAPI. Rfoley0917 also focused on improving event management and calendar functionality.

Artem9k's contributions: Artem9k was responsible for adding the remove friends feature, making friends on the calendar look better, fixing calendar sharing, and improving the UI for both calendars. They worked on scaling for the calendar, adding

shared events to the calendar, and integrating friends' events. Artem9k was also involved in adding events from the modal, working on addEvent logic, and implementing friend request functionality.

Sabi3820's contributions: Sabi3820 addressed try-catch calendar bugs and updated the calendar UI. They were involved in merging the calendar page, working on the start page, and updating the calendar page. Sabi3820 contributed to the formatting, fixing a minor menu discrepancy, and integrating calendar functionalities such as event display and processing calendar data.

Use Case Diagram:



Test Results —----------

Our test plan involved us using the app for ourselves, and trying to see what was wrong, and additionally, our test plan had us let other people outside our group try the app out and give feedback for what could be improved. While using the app, it would be very obvious what issues the app would have, if any. We found this method of testing to

be very useful, because of the simplistic nature of backend, and the relative complexity of our frontend meant that we encountered frontend bugs. This was because as one developer would make the app look pretty on their screen, the app could look weird on another developer's monitor, due to differences in monitor sizes. This required having people who were not familiar with our project test it, and make sure that the website looked presentable. Additionally, we would test certain features of the app manually. While this was certainly not the most efficient way, we found it to be manageable, because of the relatively small size of features of the app. By far the most annoying thing for us to test were the backend api's. This was due to the fact that sometimes it would be hard to tell if it was the backend that was failing, or the frontend that was failing. Additionally, restarting the app takes up to 10 seconds, which while not sounding like much can become extremely annoying when making many changes to the app. Additionally, testing the APIs was easily the hardest tests that we needed to do, because the APIs do multiple tasks, (for example, to add a friend we needed 3 APIs), actually testing the functionality of the APIs could prove challenging at times.

Deployment: This is where our site was deployed

recitation-14-team-02.eastus.cloudapp.azure.com:3000