

- **Title:** Trip Buddy

- **Contributors:**

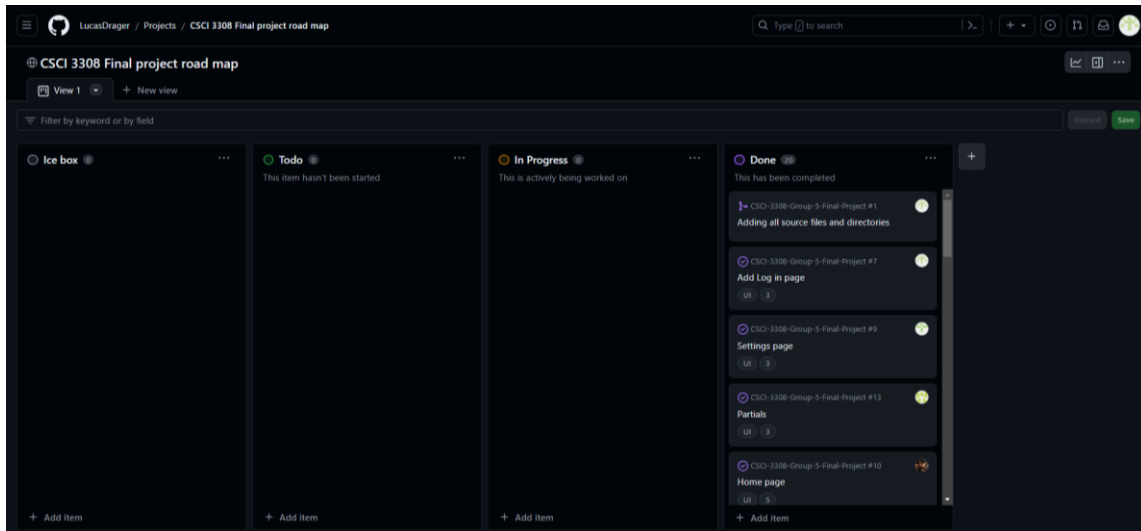
- Adithya Narayanan
- Brian Avner
- Kyle Griffin
- Lucas Drager
- Nasir Naqvi
- Phillip Kononenko

- **Project Description:**

- Trip Buddy is a ride-share platform designed as an alternative to traditional, paid-transit services. The website allows users to list their planned trips and relevant details (destination, starting location, time, etc) to a centralized hub, as well as join trips from the aforementioned hub. This ultimately saves all parties money, as gas money can be split between riders, offers an eco-friendly alternative to individual transit, and circumvents the organizational headache of finding people to carpool with on non-dedicated platforms. Unlike other ride-share apps like Uber, Trip Buddy does not have a built-in pricing structure or service fees. Instead, users are encouraged to operate under their mutual interest. This also allows the site to be used as a free organizational tool for friend groups. The site also provides a user review system and in-app messaging system to ensure that users can make informed decisions on who they chose to travel with, while protecting their own personal information.
- A use-case of Trip Buddy is uniting CU students who want to take a ski day trip to an Epic owned ski resort. While CU offers affordable buses to ICON resorts, such as A-Basin, if a student wants to go to a non-ICON resort they currently must drive themselves or pay for a taxi-adjacent service. This is particularly difficult for students without a car. Trip Buddy offers a platform for these students to find one-another, making trips like these much more reasonable in all regards, and maybe even initiating some new friendships.

- **Project Tracker - GitHub Project Board:**

- <https://github.com/users/LucasDrager/projects/3>



- **Video:** 5-minute or less video demonstrating your project. Your audience is a potential customer or person interested in using your product.

- [https://github.com/LucasDrager/CSCI-3308-Group-5-Final-Project/blob/main/Milestone\\_Submissions/video1026795593.mp4](https://github.com/LucasDrager/CSCI-3308-Group-5-Final-Project/blob/main/Milestone_Submissions/video1026795593.mp4)
- OR
- <https://drive.google.com/file/d/1snghRV6HooV9OZrR9zQOsNaohQKBHODk/view?usp=sharing>

- **VCS:** Link to your git Repository. Instructor/TAs will check, weekly, to ensure the following are stored in your VCS repository:

- <https://github.com/LucasDrager/CSCI-3308-Group-5-Final-Project>

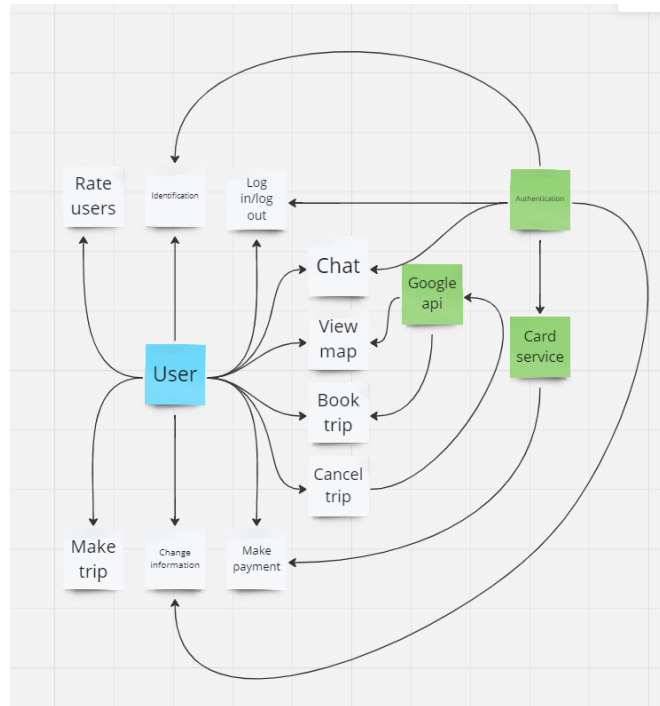
- **Contributions:**

- **Lucas:**
  - My contributions to the project were mainly the initial setup with the repository, making sure the application ran through docker, setting up test cases, trip display on the home page, and setting up the cloud service to host the final version of our project. I also performed general upkeep on the git repository and dealt with merge conflicts.

- **Phillip:**
  - My contributions to the project were set up with API calls and UI. I mainly worked on the settings page myself creating all the necessary components and api calls handling username change and password change and deletion and uploading of profile images that made me alter the DB slightly to accommodate. I worked on the set up of the profile page with Kyle doing the UI. Overall I worked mainly on these two functionalities and some CSS and making sure APIs are being called correctly and frontend functions and styling.
- **Nasir:**
  - My contributions to the project included the design of the database, login/registration pages, and payment system. The database design included minor updates from other team members as we progressed with the project. Additionally, I did the backend and frontend of the login and registration pages; I did the ejs, css, and backend javascript for them. Furthermore, I maintained and added security to the overall web page using json web tokens and browser cookies to keep the user signed in. Finally, I completed the payment page with calls to the Paypal API.
- **Brian Avner:**
  - I was responsible for implementing the “trip” functionality, i.e. allowing users to create/view/edit trips and to join other users’ trips. I also built dev-side tools for utilizing trip data elsewhere on the website. My work involved creating API routes to write, utilize, and modify trip data, updating the database structure and APIs (+ corresponding calls) to accommodate new features (payment requirements, date/time fields, etc), ensuring the data visible to users was session-specific, and building the “Create (and edit) Trip” and (with Lucas) “Join Trip” web pages.
- **Adithya Narayanan:**
  - My contributions for this project were implementing the basic skeleton of the home page followed by working on the chats page. Initially I was working on a user to user chat page which I had working to a certain extent but because of a small mistake where I accessed the page directly without logging in hence was not able to store data properly led me to start afresh and work on a group chat page.
- **Kyle Griffin:**
  - For the *Trip Buddy* project my main contributions were the ‘Buddies’ page, the ‘Profile’ page, and the partials (nav-bar, header, footer). To implement the ‘Buddies’ page I had to set up API routes in order to access all the users in the database and their profiles. For the ‘Profile’ page, I worked with Philip who set up the profile image and settings, but

otherwise I also had to implement API routes there. On all these pages I also made sure to implement CSS styling, and ended up spending a good amount of time implementing the same styling across the entire website.

### Use Case Diagram:



- **Test results:** In Lab 11, you created a Test Plan. You need to include the test results and observations in the project report. Refer to **this** for more information
  - [https://github.com/LucasDrager/CSCI-3308-Group-5-Final-Project/blob/main/Milestone\\_Submissions/Testing%20documentation.txt](https://github.com/LucasDrager/CSCI-3308-Group-5-Final-Project/blob/main/Milestone_Submissions/Testing%20documentation.txt)
- **Deployment:** Link to deployment environment or a written description of how the app was deployed and how one might access/run the app. The app must be live, working, and accessible to your TA.
  - <http://recitation-013-team-05.eastus.cloudapp.azure.com:3000/>
  - If the Azure host stops working you can find instructions for how to run locally in the readme.md file.