EE / CPR E / S E 492 – sdmay21-51

Team 51: Constrained Re-Planning in Spatial Crowdsourcing

Bi-Weekly Report 6: 03/29 – 04/12

Client &/Advisor: Goce Trajcevski

Team Members:

- Logan Anderson: Frontend/Test Engineer
- Steven Sheets: Backend/Test Engineer, Report Manager
- Nicholas Heger: Frontend Engineer, Progress Manager
- Jared Weiland: Backend Engineer
- James Volpe: Frontend/Backend Engineer

Past Week Summary:

- The overarching theme of the past two weeks has been frontend and backend communication. Our team has implemented graphql into our project to accomplish this goal, and we have been communicating with each other often to ensure the frontend and backend stay on the same page and all information is formatted consistently.

Past Week Accomplishments:

- Got proof of concept working for two algorithms - 1 that prioritizes the minimum distance traveled, and another that prioritizes the shortest time to complete the task. For example, if there is a worker A, 1 minute away from a new job, but is 20 minutes from completing their current job, and there is another worker B 10 minutes away from the new job, but has no current job, then worker B is selected for the shortest time, but worker A would be selected for the shortest distance. Tested these using workers and jobs positioned throughout Ames and using graphiql. – Steven
- Fixed several bugs with mutations after further testing. Thank frontend guys for finding some bugs. – Steven, Nicholas
- Implemented Apollo graphql on the frontend. This involved adding a client and making a mutations component that could store the mutations to be used to change data on the backend. – Nicholas, Steven
- Added classes on the frontend for LocationInput and SkillInput that take a latitude and longitude, and a skill and rating, respectively. These are necessary for mutations to update the database that require location or skills. – Nicholas
- Worked on getting the sign-up page finished off. First, fixed a bug where the create account button needed to be clicked twice to create the account. This was caused by running the validator when it was pressed and thus did not know if the account were valid or not until afterwards so

would not navigate users until it was pressed again. Solution was to run validation onChange of any input fields. (validation for account creation is client-side so this is not a performance/network traffic concern). Next, added the createEmployer and createWorker mutations that are triggered when the submit button is clicked. Made the correct one run based on the account type dropdown and return the user's id for either mutation. The user's information is all passed through the app state right now, but the id is stored in local storage because of the delay from receiving it back and because it is needed across any page of the app for future mutations and queries. – Nicholas

- Continued implementation of the assignments page for workers. Progress made on this involved getting the worker's current location using the navigator and pushing that to the backend via an updateWorker mutation that was also added. Currently this happens through a button that pushes information to the backend, but the plan is to switch this to happening onMounting of the assignment page's component. This mutation also requires accessing the userId that is in local storage. Additionally, added a header to the list of assignments to better differentiate between pages. – Nicholas
- Began working on a layout for the team poster. Looked through older posters for good examples and shared them with the team. – Jared
- Began analysis of semester 1 design document for reusable components – All

Pending Issues:

- Need to figure out how to perform queries using graphql on the frontend.
- Centralize States on frontend though route
- User authentication via JWT tokens

Individual Contributions:

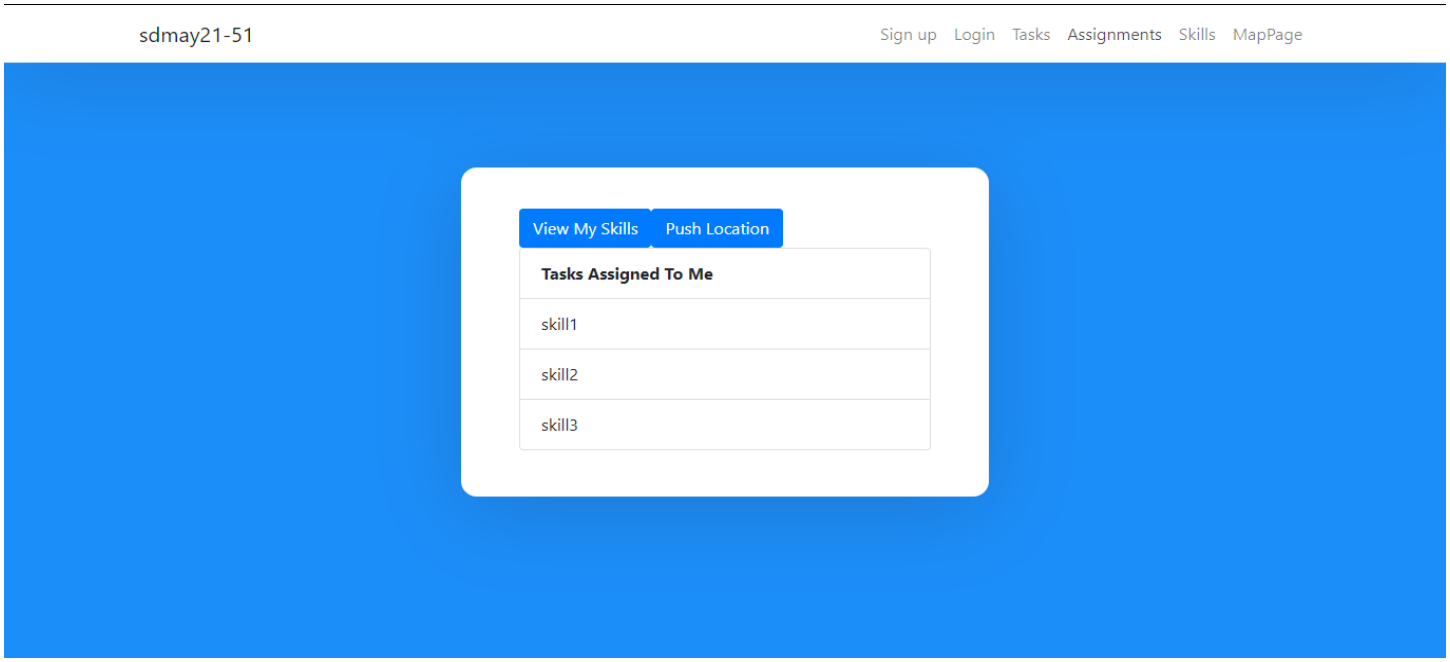
Name	Contributions	Bi-Weekly Hours	Cumulative Hours
Logan Anderson	React, Report, Mapbox	6	40
Steven Sheets	Meeting Organization, Report, graphiql, mutations, algorithms	8	45
Nicholas Heger	React, graphql, Report	14	57
Jared Weiland	Server and Database, Report, Poster	4	27
James Volpe	React/graphql/Mongo Research, Report	7	27

Plans for Coming Week:

- Create poster for presentation
- Update design document for final
- Implement queries on the frontend to fetch data from the database, rather than just modify it with mutations.
- Request data from backend for map display

Some Screenshots:

Assignments Page



graphiql Interface: