Time Tracker Project Discussion

The main constraint, as well as my primary concern for the Time Tracker project, was the 8 hour limit. I had done many hackathons in the past, and I understood the approach for such a limited project. My main goal was to provide a minimum viable product, which meant completing the most important tasks first. I divided the project into 3 sections: frontend, server, and database, and worked on the database and server first.

For the tech stack, I decided to use MERN as it was a popular stack for simple projects. Before starting with the actual implementation, I made sure to design the database collections and draw up a mockup of the UI.

There were many features I wanted to implement, but I only chose the most necessary ones. These were features not explicitly specified in the instructions, but are likely implied:

- Page access: certain pages are not accessible before the user logs in
- Logout: users are able to logout of their account, which then makes certain urls private
- Basic error handling for API calls
- Input field validation for the Signup, Login, and Add Entry pages
- Data persistence

Other decisions I made include:

- A Home page and an Add Entry page: I separated the Home and Add Entry pages because I intended to have a useEffect trigger the fetching of entries in the Home page.
- Storing the Projects list in the frontend instead of the database: I understood that it was more robust to store the projects list in the database, but as it would have made no actual difference to the user and would have taken more time to implement, I decided to store them in a constants file instead.
- Time zone handling: I did not have enough time to implement a way to automatically detect the time zone from within a container, so I settled for having an argument in the Dockerfiles that can be passed from the docker-compose file. Nevertheless, I decided it was worthwhile to account for time zones and added functionality to store and retrieve entries in a given time zone's week. If I had more time, I would look into further solutions that do not require processing all timestamps in UTC.

Overall, my Time Tracker implementation is a minimum viable product as it allows users to login and add work entries for projects, as well as view their work for the week. Data is persisted, and the UI is simple. One design choice I made for the UI was for the project

selection to be from a drop-down menu, as it was the easiest and fastest for the user. However, if the list of projects was larger, I would make sure the drop-down menu also had a search bar.

Take-Home Test Feedback

The Time Tracker project is well specified and is overall a good test for applicants. It is a full-stack project and covers most basic aspects of web development, and the test document also has a good sense of requirements and objectives. However, the test may be a little too difficult for junior developer applicants, especially if they do not have experience with Docker. In an 8-hour project, spending too long trying to make one part function likely means not being able to finish the entire project. For junior applicants, perhaps a simpler test could be administered, such as one without a login page, without complex database querying, or without dockerization.