35L Project Weekly Report

GROUP MEMBER: Cyrus Asasi

Week 4:

After meeting and discussing various application ideas with fellow group members, we came to the consensus of working on an app that would work closely with the Spotify API. During our week 4 meetings, we agreed on a division of labor scheme where I would implement the entire backend on my own, then work closely with each other group member individually to implement each unique feature.

I spent the rest of week 4 researching what the best framework for our backend server would be. I was initially leaning towards Flask, due to its simplicity and popularity, but I eventually decided on using a backend framework called Quart, because I wanted to work with Python's extremely fast asynchronous wrapper of the postgresql database: asyncpg.

Quart was designed to be an async copy of Flask, which means it has nearly every single feature that Flask boasts, but it also uses python's async/await syntax to enable parallel programming and thus better performance.

After deciding the frameworks and modules I would use to develop the backend, I began reading the Spotify documentation in great detail. My goal was to learn and understand the Oauth authentication system spotify implements by the end of week 4 to ensure that I'd be able to start coding the necessary backend endpoints for my groupmates to utilize on the frontend as soon as possible.

Week 5

In week 5, I first focused on setting up my environment and writing a detailed readme outlining how to set up the backend. After designing the database tables, I then got to work at implementing a user login system so I could start working with Spotify data.

By the end of week 5, I had a fully functional web application running that would login users to Spotify, store the necessary credentials, and use http requests to fetch spotify data from the API. I quickly noticed that the harsh rate limits imposed on my app by spotify would cause response times to suffer dramatically, so I planned to implement a system of caching data the following week.

Week 6

Since I had laid down the groundwork for the backend in the previous weeks, I decided to pick up the pace with my development. I implemented a caching system that would cache the result of a function and store it for a given amount of time before invalidating it. After this, I noticed a dramatic increase in speed when accessing any Spotify data. By the end of week 6, I was told that setting up postgres was causing some issues for some members in my group, so I decided to implement a fallback JSON database where data would be stored locally in a directory in the event of any postgres connection failure.

Week 7

Having completed most of the features required for the backend, I started to work more closely with Ryan Oh and Angela Tan to get them the necessary data to implement the MyStats page and the audio upload/search features. From here on out, I focused primarily on bug fixing and writing endpoints in web.py where my group mates could make the necessary get and post requests to upload and fetch data from both spotify and the database.

Week 8

During week 8, I started working on frontend elements. I started to work on the Liked Tracks page, where I decided to use jquery and jinja templating to inject python data from Spotify into an html webpage. I spent the majority of week 8 reading documentation on javascript, jquery, and css to educate myself on how to implement the audio playback feature on hover/click, just to the left of each Spotify track on the liked tracks page.

Week 9

In week 9, I refined and perfected the Liked Tracks page using css and the bootstrap library to create a clean look. I also implemented a feature where the user would be able to filter through their liked tracks using a sorting menu. The rest of week 9 was time spent debugging and cleaning up code to create a readable and portable environment.

Week 10

Having completed all my goals for this project, I spent week 10 trying to assist other members in my group with debugging and styling the frontend. I also spent some time updating the readme and planning out our project presentation.