Week 4 -

Met with members on Sunday to brainstorm ideas and to finish proposals and schedule for submission. I created a proof of concept that is in the poc branch to test interactions between a React and a Flask server which worked properly. I also talked with Cyrus later and we decided to switch from Flask to Quart because the second one had similar syntax and was asynchronous. I also created the repository itself and sent out all the invites. We also created a schedule to meet on Thursday afternoon every week to do some progress.

week 5 -

Met with team members to wireframe the frontend with powerpoint. We also solidified the specific features we wanted and a rough idea on who was going to be in charge of what. I decided to do the audio playback feature which took the MP3 files from the backend to be played.

week 6 -

Met with team members to integrate the frontend React project with the backend Cyrus wrote and created the basic file structure and method of starting the two servers. Started testing out the backend to iron out bugs. Switched the framework from Next to the default react project from npx because it was more familiar to other people.

week 7 -

Met with team members to work on the project together. Dealt with a lot of windows specific errors that did not affect my teammates because they all used unix. Figured out an issue with Spotify API rate limiting and also the cookies situation. The cookie was not set properly because it was tied to “/spotify” instead of the base of the url which was problematic as it was not retrievable by the frontend.

week 8 -

Met with team members to work on the project together. Continued to deal with more problems as I was the only one using the JSON database as windows by itself cannot install PostgreSQL. Figured out a bug in the code that dealt specifically with windows. Also installed WSL and ported so that I would be using the same infrastructure as my teammates. Working in WSL revealed issues with using React as the program tried to use the windows installation of npm by default which was not allowed, but installing it on Wsl was also buggy. In the end, I had to run the servers separately where one was on windows and the other was in WSL so that PostgreSQl worked. Later figured out the mistake in the JSON database.

week 9 -

Met with team members and began to work on the audio playback as my setup finally was working reliably. Researched different ways to playback audio on a web application and decided that using an HTML element and receiving the MP3 file as a url from the backend was the easiest. Experimented with a raw audio html element, but realized that the controls were hard to customize and make it cohesive with the project's themes itself. Created a working version which allowed for a user to playback mp3 files after being uploaded through Angela's upload form. Found an error with the audio file storage itself where trying to insert into the JSON database causes an error because the way that the backend checked for an existing file was to attempt to get it and cause a KeyError that was never catched. Switched it to .get instead which properly returned None if there was an error which prevented error 400.

week 10 -

Met with team members multiple times throughout the week to finish the project. Met with Angela on Thursday to fix a bug in the code where the keys were different from the front and backend which caused a form error. Met with the entire team on Friday to finalize the CSS and few remaining features such as the audio playback volume and start and stop button. Worked with Helen on implementing volume and start and stop feature. We did not have time to finish the volume before the presentation but we managed to switch out the controls from the default controls in the html tag itself which was browser dependent to one that controlled the html element itself by element id.