Kaylee George
CS 194W
24 March 2023

Software Fair Remote Submission

Demo Video Link:
https://drive.google.com/file/d/1gNk2QrOKrPkQLH5P0WscdREPdQnuRysz/view?usp=share_link
Notable Links: Code commits for initial Flask/Next.js setup, frontend, and API communication. Created Figma mockup for frontend.

I think my biggest contribution to this project was implementing the frontend/backend architecture and maintaining this high-level understanding of how the web application worked end-to-end throughout the project. I created the initial project scaffold of the web app using Flask (backend) and Next.js (frontend) and set up the communication between the backend and the frontend. I created an initial sample API where the backend in Flask serves the requests given by the frontend (written in Next.js). I also had to work with CORS in order to properly make the server-client communication work properly.

As part of this project, I think I improved my understanding of how the frontend and backend are separated by functionality, as well as how they fit together to create a cohesive web application. I enjoyed increasing my understanding of how Flask works and what it takes to have a backend written in a different language than the frontend. I think it was a good exercise to work through the tradeoffs of creating this architecture (as I haven't personally done that before) — and ultimately, it paid off in the end. When Alex integrated his code, I worked on hooking the backend up with the frontend and making sure it all ran smoothly.

I also mocked up the frontend design and implemented the initial frontend. The frontend was implemented using React, where I created the index page and router in case we wanted to add more pages. For the UI, I used the Chakra UI library for components (such as for the stylized input bar) — which is a well-known design library for React components. I really enjoyed doing the frontend! I also gave guidance for other team members when they wanted to learn more about frontend.

When it came time to deploy, I worked on implementing the Heroku flow and the S3 bucket handling, so that we could move our interface off localhost and into production. I set up the S3 bucket itself and the Python communication with the S3 bucket — which was written in Flask. There was a challenge because the client code and server code are both contained in one repo but you can only deploy one in Heroku (one Heroku project attached to a repo). It was cool to learn about what goes into production and this helped me learn more about the build process.