Sprint 2 Outline

Our second sprint involves finishing the minimum viable product, and ensuring that the product works for every person involved in production and development. As of now, there are a few **loose ends** to handle especially in regards to local development versus Docker-based development, and the proposal of linting and schema design.

Intended features to implement

These descriptions should involve:

- The rationale behind how those features have been prioritized.
- A relatively specific breakdown of what needs to be done to deliver each of these features.
- What specific tasks need to be accomplished - by which people, due on what days
- What acceptance criteria exist for these features - in other words, how will you know that this work is finished?

In assignment 2, we were able to reach most of our goals. In assignment 1, we stated that our goals for A2 were to connect our backend to our front end, support different types of requests, and have basic functionality on the frontend to display our data. For A3, we planned on having different visualizations and have users be able to import similar datasets.

For assignment 2, we prioritized features from both parts 2 and 3 so we could keep our project on track. Those features include a dataset uploader, a visualizer view, a dataset selector, and the backend for those respective features.

Dataset Importer

The Dataset Importer is an important feature in order for the user to be able to use the application since all other features are dependent on this. We want the user to not only be able to upload and work with the crime dataset that we chose, but any dataset of their choice. Currently, the user is able to upload a CSV file which gets sent to the backend to be stored in an SQLite database. In the future, we plan on improving this feature by allowing the user to upload multiple files all at once instead of one by one, drag and drop to upload, and support for other file formats.

Importing datasets is the first step in the connection between the front end and the back end. It ensures that the application works when arbitrary valid data has been imported.

Visualizer view

The visualizer view was deemed as an important addition early on as our main goal for this project is to visualize crime data throughout the United States. So far, we have been able to plot crime data for counties that we have data for using a pin on a map. Users are able to click this pin to see more data about the crimes.

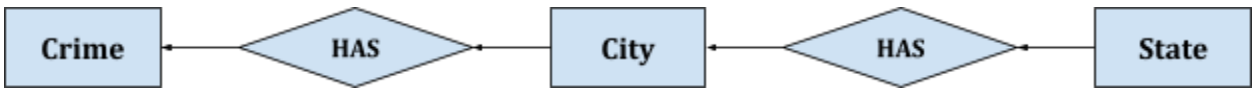
In the future, we plan on improving this feature by adding color coded data representing crime severity so that users are able to easily determine the crime statistics for a certain area. Additionally, we will have this color coded data fill the entire geographical area of that county similarly to how if you click the name of a city in google maps, the geographical boundaries show up. Lastly, we will determine a way to compare crime across various different areas with vastly different demographics and populations such that those differences do not matter as much.

Dataset selector

Our Dataset selector allows us to filter out the datasets. Filtering out the dataset allows us to focus on the data that are relevant for analysis, while avoiding clutter when the remaining data are not to be used. The dataset selector also helps improve the load time.

Database Schema

Our current database schema should be simple.



*
**

Blameless postmortem

- What goals have been achieved?
    - Simple dummy application to build on
    - A rough plan for our milestones
    - Tech stack and reasoning behind each technology planned out
- What goals have been missed, and by how much?
    - The containerization of the application was not met. Instead, we automated using a shell script as a temporary solution
    - Splitting up the work for the milestones among ourselves
    - Project planning board setup. For example, a Kanban board in our GitHub repository.
- How plans will need to be adjusted, based on remaining time available and project scope.
    - The plan is feasible. Additional features were given low priority to ensure we complete the essential features.

What constitutes success in milestone 2

In milestone 2, the completion of the core features of the application which other features will depend on constitute success. Having a foundation on the frontend in the form of a **React Dashboard** with navigation setup, and having a **Flask** backend API with endpoints setup allow the application to not only be functional but easy to build upon for future milestones.

Milestone 2

Milestone 2 is mainly focused on creating a frontend with basic functionality, the acceptance of local data, and connecting that to our backend.

Tasks that should be completed are

1. Support different types of requests
2. Have basic functionality on the frontend to display our data
3. Connect our backend to our front end

Additionally, we want to validate data by providing GitHub Actions and validation testing.

How did we validate

Unit tests, discussions about whether or not our output is acceptable.

1. **Demonstrated progress towards one or more of your next milestones:**
    - Clearly state what constitutes success in this milestone, in a paragraph.
    - Each team member's responsibilities for reaching the milestone, with a status (complete, in progress, not started)
    - Your validation process: again, how do you know that you have achieved this milestone?
    - Demonstrating progress includes:

Philippe	- Created a frontend user interface for data visualization. Users are able to visualize points on the map where a dataset has information. - Added a popup that shows up when clicking a point on the map to see crime data. - Created a **backend** endpoint that will serve data set information as json to the front end - Created a docker container that will install all required python libraries and node packages
Piyush	- Created a frontend dashboard user interface in React to house the components as features. Added navigation between these components and made the frontend extensible. - Implemented the import dataset feature by adding the uploading functionality on the frontend, creating a POST request to be sent to the Flask backend, and creating a function to process this request and create an SQLite database from the file received. - Both of these items are **Complete** and functional but will likely be improved upon in the later milestones for a better user experience.
Mack	- Created **Github actions** to lint code and validate the update, but need to revise the YML. - Considered proposing database schemas. - Faced **blocker** after unable to run in node. Reported this incident to Michael Hoye. - Resolved as of Sunday - Attempted to establish connection points between frontend and backend, but have trouble doing so. - Proposed a better paradigm on how to conduct future goals and any blockers one has encountered.

Mack's Extension

Unfortunately, due to professional and personal circumstances, Mack has to request up to a week of extension due to a previous Intel second-round interview that happened on 2022 November 7th at 13:00. He has spent up to seven days fervently preparing algorithms, competitive coding, and working on his full-stack internship to ensure the success of his interview. He will also encounter further interviews and career guidance, in addition to academic guidance to resolve a past crisis that has affected him in the past months which subsequently impacted his performance.

Unfortunately, Mack encountered considerable troubles in progressing his part of the project due to him encountering Node.js and NPM issues, given that he had reinstalled his computer on November 6th. As a result, he has to reinstall Node.js, but due to APT complications, he does not want Node.js v12, but the latest LTS version, namely v18. He had trouble installing the environment given that he is a full-time Linux user, and that installation instructions for Windows and Linux users may differ.

As of November 9th, he no longer has any major issues with Node.js and Docker after rereading through the documentations to tie up any loose ends. He has solved this issue with the assistance of his manager of his internship project, in which he has to develop another full-stack application.

Henceforth, Mack requested a one-week extension to deal with current career development, plus having to fix his Node.js on his end, with the help of his internship manager. Unfortunately, with the unexpected Node blocker, he needs to be more transparent in his approach in resolving this issue, even though it is his issue.