Team: Caution

Features (user stories) to Implement in Next Sprint:

- **US Accidents:**
    - **Feature 1: Data Store**
        - The data will be stored in a new data structure on the server side which will be updated if there are new entries in the database or if things are updated. The graphs will load from this cached data if possible to decrease load time.
    - **Feature 2: Updating graphs quickly based on updating input**
        - New input will be added to the cache which will then be used to be displayed by the graph. This should speed up the time compared to sending over the big data array again
    - **Feature 3: Updating graphs quickly based on adding new records**
        - New records will be added to the cache which will then be used to be displayed by the graph. This should speed up the time compared to sending over the big data array again
    - **Feature 4: Incremental Analytics**
        - Existing analytics will display based on average values on other analytics or top 10 most viewed, depending on the analytic.

Test Cases

- **Feature 2 Test Cases**: as a user, I want analytics graphs to be quickly updated after adding new records to the CSV
    - **Test Case 1**: as a user, in the New Record section, I add new record(s) to the CSV
      Correct Output: A new record request is sent to the backend. The backend updates the CSV with the new record(s). Analytics is performed only on the new record(s) and analytics data is sent to the frontend. The frontend displays an updated graph of the analytics data.
- **Feature 3 Test Cases**: as a user, I want analytics graphs to be quickly updated after updating existing records in the CSV
    - **Test Case 1**: as a user, in the Update Record section, I update existing record(s) in the CSV

Correct Output: An update record request is sent to the backend. The backend updates the record data in the CSV. Analytics is performed only on the updated records and analytics data is sent to the frontend. The frontend displays an updated graph of the analytics data.

- **Feature 4 Test Cases**: as a user, I want analytics to show average values and by frequency
  - **Test Case 1**: as a user, I want analytics to show average values and by frequency in the table and graphs

    Correct Output: The backend processes the data but only returns a specific portion as requested by the user

TO-DO LIST:

Done list of last sprint:

- Separated and modularized front end and back end code
  - Finished by Albert, Matthew, Rahul and verified by everyone
- User-friendly front end
  - Finished by Albert, Matthew, Rahul and verified by everyone
- Graph UI improvements
  - Finished by Rahul and verified by everyone
- Graph bug fixes, making one graph display at a time
  - Finished by Rahul and verified by everyone
- Updated tables for analytics
  - Finished by Matthew, Rahul and verified by everyone
- Backend development to get the average between two filters and putting that into a 2D array for analytics
  - Finished by Kenny and Danial and verified by everyone
- Modularized HTML and added tab functionality
  - Finished by Matthew, Rahul, Kenny and verified by everyone
- Implementing new features, zipcode and side of the road
  - Finished by Danial and verified by everyone
- Bug fixing and making all code work together with each other
  - Finished by Matthew and Danial and verified by everyone

To-Do for next sprint:

- Implement a new data structure which will hold the new data and improve runtime performance

- Acceptance Criteria: Update the csv with new data/entries and based on the new inputted data we add or update the preexisting data table if there is one.