Team: Caution

Features (user stories) to Implement in Next Sprint:

- **US Accidents:**
  - **Feature 1**: as a user, I want to be able to view the number of US accidents in a given year/month/day.
  - **Feature 2**: as a user, I want to be able to view the number of US accidents in a given start time and end time.
  - **Feature 3**: as a user, I want to be able to view the number of US accidents in a given country, state, county, zip code, street, and number.
  - **Feature 4**: as a user, I want to be able to view the number of US accidents within a starting latitude, ending latitude, starting longitude, and ending longitude.
  - **Feature 5**: as a user, I want to be able to view the number of US accidents of a given severity.
  - **Feature 6**: as a user, I want to be able to view the number of US accidents by existing speed bumps, stop signs, and traffic signals.
  - **Feature 7**: as a user, I want to be able to view the number of US accidents by weather condition.
  - **Feature 8**: as a user, I want to be able to view the number of US accidents by humidity.
  - **Feature 9:** as a user, I want to be able to clear filters individually.
  - **Feature 10:** as a user, I want to be able to clear all filters at once.

Test Cases

- **Feature 1 Test Cases**: as a user, I want to be able to view the number of US accidents in a given year/month/day.
  - **Test Case 1**: as a user, in the Search page, I input the year/month/day and select "Search".

    Correct Output: A list of US accidents and their descriptions matching the inputted year/month/day and the existing queries is displayed.
- **Feature 2 Test Cases**: as a user, I want to be able to view the number of US accidents in a given start time and end time.

  - **Test Case 1**: as a user, in the Search page, I input the start time and end time and select "Search".

    Correct Output: A list of US accidents and their descriptions matching the inputted start time and end time and the existing queries is displayed.
- **Feature 3 Test Cases**: as a user, I want to be able to view the number of US accidents in a given country, state, county, zip code, street, and number.
  - **Test Case 1**: as a user, in the Search page, I input the country, state, county, zip code, street, and number and select "Search".

    Correct Output: A list of US accidents and their descriptions matching the inputted country, state, county, zip code, street, and number and the existing queries is displayed.
- **Feature 4 Test Cases**: as a user, I want to be able to view the number of US accidents within a starting latitude, ending latitude, starting longitude, and ending longitude.
  - **Test Case 1**: as a user, in the Search page, I input the starting latitude, ending latitude, starting longitude, and ending longitude and select "Search".

    Correct Output: A list of US accidents and their descriptions matching the inputted starting latitude, ending latitude, starting longitude, and ending longitude and the existing queries is displayed.
- **Feature 5 Test Cases**: as a user, I want to be able to view the number of US accidents of a given severity.
  - **Test Case 1**: as a user, in the Search page, I input the severity and select "Search".

    Correct Output: A list of US accidents and their descriptions matching the inputted severity and the existing queries is displayed.
- **Feature 6 Test Cases**: as a user, I want to be able to view the number of US accidents by existing speed bumps, stop signs, and traffic signals.
  - **Test Case 1**: as a user, in the Search page, I input the existing speed bumps, stop signs, and traffic signals and select "Search".

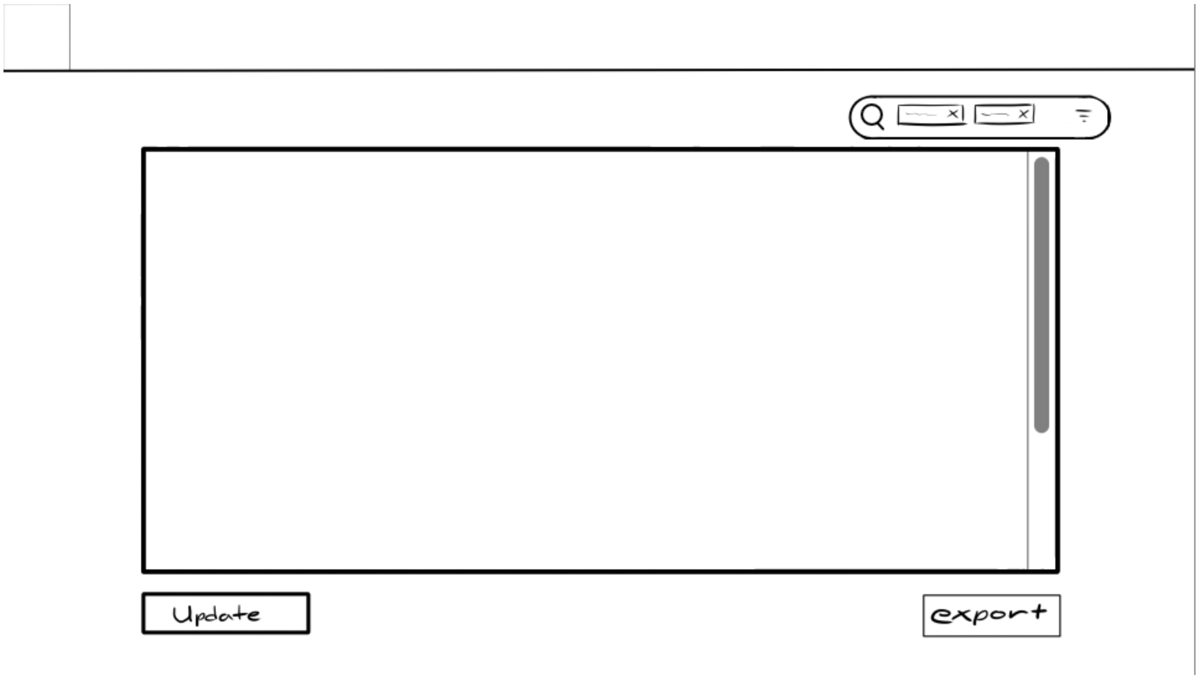
    Correct Output: A list of US accidents and their descriptions matching the inputted existing speed bumps, stop signs, and traffic signals and the existing queries is displayed.
- **Feature 7 Test Cases**: as a user, I want to be able to view the number of US accidents by weather condition.
  - **Test Case 1**: as a user, in the Search page, I input the weather condition and select "Search".

    Correct Output: A list of US accidents and their descriptions matching the inputted weather condition and the existing queries is displayed.

- **Feature 8 Test Cases**: as a user, I want to be able to view the number of US accidents by humidity.
    - **Test Case 1**: as a user, in the Search page, I input the humidity and select “Search”.
      Correct Output: A list of US accidents and their descriptions matching the inputted humidity and the existing queries is displayed.
- **Feature 9 Test Cases**: as a user, I want to be able to clear filters individually.
    - **Test Case 1**: as a user, in the Search page, I click an “X” button next to a given filter inside the search field
      Correct Output: The given filter is cleared from the search field.
- **Feature 10 Test Cases**: as a user, I want to be able to clear all filters at once.
    - **Test Case 1**: as a user, in the Search page, I click an “X” button next to the search field.
      Correct Output: All filters are cleared from the search field.

Design:

- **Search Page:**



TO-DO LIST:

Done list of last sprint:

- Setting up client-side
    - Finished by Danial and verified by everyone
- Setting up server and implementing socket.io for connection

    - Finished by Danial and Matthew and verified by everyone
- Implementing client-server communication
    - Finished by Danial and verified by everyone
- Hosting for testing
    - Finished by Matthew and verified by everyone

To-Do for next sprint:

- Backend program to search for preferred criteria in a CSV
    - Acceptance Criteria: When clicking a button to search for a certain criteria we will be able to pull results from the CSV about the search criteria
- CSV parser to print out data from CSV
    - Acceptance Criteria: CSV data can be stored as objects in JS
- Front-end work in order to search through the backend CSV
    - Acceptance Criteria: A button that has a feature listed on it that then searches using filters for that feature in the CSV and returns the occurrence