

<u>Team: Caution</u>

Features (user stories) to Implement in Next Sprint:

- **US Accidents:**
 - **Feature 1:** as a user, I want to be able to compare the number of accidents based on weather.
 - **Feature 2:** as a user, I want to be able to compare the average severity of accidents based on weather.
 - **Feature 3:** as a user, I want to be able to compare the number of accidents based on location. (near a highway vs. not near a highway)
 - **Feature 4:** as a user, I want to be able to compare the number of accidents based on the city.
 - **Feature 5:** as a user, I want to be able to compare the number of accidents based on the month of occurrence.
 - **Feature 6:** as a user, I want to be able to compare the number of accidents based on having stop signs and/or speed bumps at the accident.
 - **Feature 7:** as a user, I want to be able to compare the number of accidents based on humidity level.
 - **Feature 8:** as a user, I want to be able to compare the average severity of accidents based on humidity level.

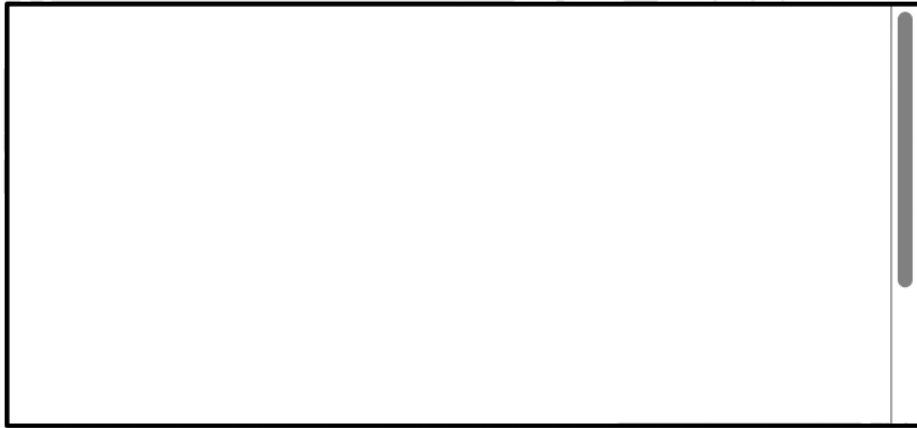
Test Cases

- **Feature 1 Test Cases:** as a user, I want to be able to compare the number of accidents based on weather.
 - **Test Case 1:** as a user, in the Statistics section, I select “Number of Accidents on Weather” and click the “Compare” button.
Correct Output: A statistic request is sent to the backend. The backend finds the number of accidents based on weather. The backend returns the values to the frontend. The frontend displays a graph relating weather conditions and number of accidents.
- **Feature 2 Test Cases:** as a user, I want to be able to compare the average severity of accidents based on weather.

- **Test Case 1:** as a user, in the Statistics section, I select “Severity of Accidents on Weather” and click the “Compare” button.
Correct Output: A statistic request is sent to the backend. The backend finds the different types of weather conditions and based on the conditions, looks at the severity of the accidents. The backend returns the values to the frontend. The frontend displays a graph relating weather conditions and severity of accidents.
- **Feature 3 Test Cases:** as a user, I want to be able to compare the number of accidents based on location. (near a highway vs. not near a highway)
 - **Test Case 1:** as a user, in the Statistics section, I select “Number of Accidents on Location (highway)” and click the “Compare” button.
Correct Output: A statistic request is sent to the backend. The backend finds the number of accidents based on location (highway). The backend returns the values to the frontend. The frontend displays a graph relating highway existence and number of accidents.
- **Feature 4 Test Cases:** as a user, I want to be able to compare the number of accidents based on the city.
 - **Test Case 1:** as a user, in the Statistics section, I select “Number of Accidents on Cites” and click the “Compare” button.
Correct Output: A statistic request is sent to the backend. The backend finds the number of accidents based on cities. The backend returns the values to the frontend. The frontend displays a graph relating cities and number of accidents.

Design:

- **Search Page:**



Update

export

New Entry:

Update:

Update

New Entry

column | col | ...

| col

Insert/Update:

Update

New Entry

Analytics1:

	<div><div></div></div> <p>If <u>condition/factor</u>, then <u>statistic</u></p> <div>Sorted list/ plot</div>

TO-DO LIST:

Done list of last sprint:

- Updated table styling to space out data in order to make it look more visually appealing
 - Finished by Rahul and Kenny verified by everyone
- Search bar on data table to quickly search for data points
 - Finished by Rahul and verified by everyone
- Multiselect feature functionality
 - Finished by Kenny and verified by everyone
- Input form for inserting data into the table
 - Finished by Albert and Danial and verified by everyone
- Input fields for updating and deleting data
 - Finished by Albert and verified by everyone
- Front-end styling to make the form more visually appealing
 - Finished by Matthew and Kenny and verified by everyone
- Back-end development to update data array when user inserts data, updates the data, and deletes the data
 - Finished by Danial and verified by everyone

- Back-end development in order to display data that the user has inputted
 - Finished by Danial and Rahul and verified by everyone
- Parsing the data array and downloading it as a backup CSV
 - Finished by Matthew and verified by everyone
- Saving the downloaded CSV into a local file in the working directory
 - Finished by Matthew and verified by everyone
- User inputted file name for custom data input
 - Finished by Danial and verified by everyone

To-Do for next sprint:

- Front end displays statistic plots/tables
 - Acceptance Criteria: When clicking the compare button we are able to see from the plot/table the statistics and text area updated
- The compare button works to display the analytics
 - Acceptance Criteria: The compare button when pressed will be able to query the specific data we need for that analytic and connect that to the table/plots.
- Backend needs to be able to query the values
 - Acceptance Criteria: The correct values will be queried when accessing the database and displayed on the frontend