Artifact 2

Feature 1
As a user I want to search accidents by date and time.

Feature 2
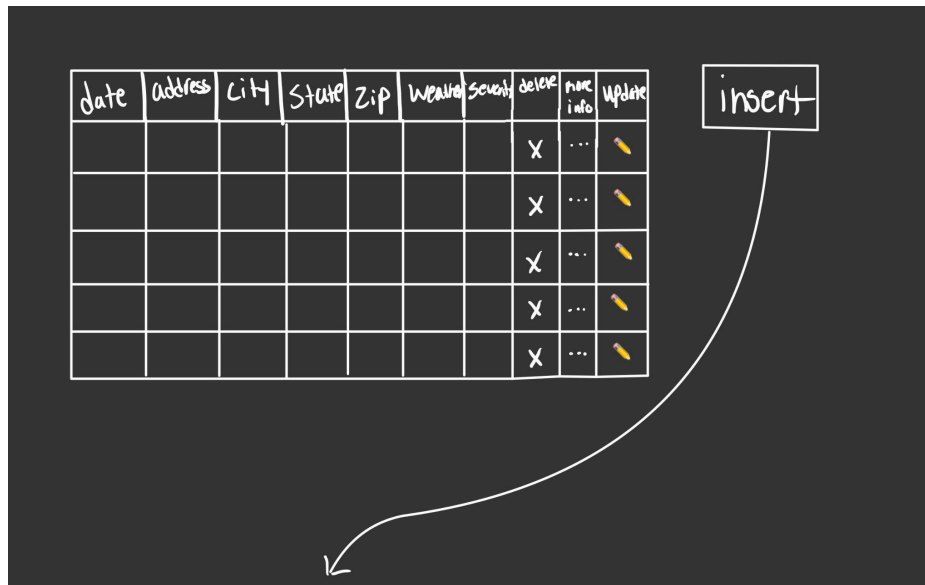As a user I want to search accidents by severity (1 being the lowest and 4 being the highest) or average severity.

Feature 3
As a user, I want to search up accidents by state, city, county, zip code or address.

Feature 4
As a user, I want to specify my search by toggling options such as weather / rain.

GUI

Test cases

Feature 1 test cases: As a user, I want to search up the number of accidents by date and time

       Test case 1: as a user, I want to be able to write into the search bar .

       Correct Output: The search bar allows me to input into the search bar

       Test case 2: As a user, I want to check for incorrect output where a letter is entered instead of a number.

       Correct Output: The users gets a message that tells them to retype their input (error message)

       Test case 3: As a user, I want the search button to process my input

       Correct Output: The website displays a table with all the records

Feature 2 test cases: As a user I want to search accidents by severity (1 being the lowest and 4 being the highest) or average severity.

- Test case 1: As a user I enter something that is not a number.
  Correct Output: The user gets a message prompting them to only input numbers.
- Test case 2: As a user I enter a number out of range.
  Correct Output:The user gets a message prompting them to only input numbers from 1-4.

Feature 3 Test Cases: As a user, I want to search up accidents by state, city, county, zip code or address.
- Test case 1: as a user, I type "california" in the search bar and hit enter
  Correct output: The website displays a table with existing records for accidents recorded in California
- Test case 2: as a user, I type "" in the search bar and click the magnifying glass icon.
  Correct output: The website displays an empty table with no records
- Test case 3: as a user, I type "riversid" (misspell city name of Riverside) and hit enter
  Correct output: The website displays an empty table with no records

Feature 4 Test Cases: As a user, I want to specify my search by toggling options such as weather and visibility
- Test case 1: as a user, I toggle on the weather option and select if there was light rain during the accident and press search.
  Correct output: The website displays a table with accidents that happen when there was light rain.
- Test case 2: as a user, I toggle on the visibility option and input the miles of visibility that I am interested in and press search.
  Correct output: The website displays a table with accidents that happen with the inputted visibility.

Taskboard:
- Back end: load data from CSV files.
- Back end: parse data (date, city, state)
    - Turn csv to json format.
- Back end: Respond to a search request by calling the functions that return the correct data
- Back end: Implement a function that returns all the accidents for a specific date in the format ('mm/dd/yyyy')
- Back end: Implement a function that returns only accidents where it is raining.
- Back end: Implement a function that returns accidents of a specific severity.
- Back end: Implement a function(s) that returns accidents according to city, county, state, zip code or address.

- Front end: Read a string / search and send it to the back end after hitting submit.
- Front end: Turn strings into JSON format (with booleans for all the toggle options).

Done List of Last Sprint
- Yiu Ming Wong & Ivan Carrillo & Estela Ramirez & Jacob Alamparambil & Thuan Vu = Worked on connecting the front-end to the back-end (worked together on call)\
- Ivan Carillo -Set up initial reactJS
- Yiu Min Mong recorded the demo