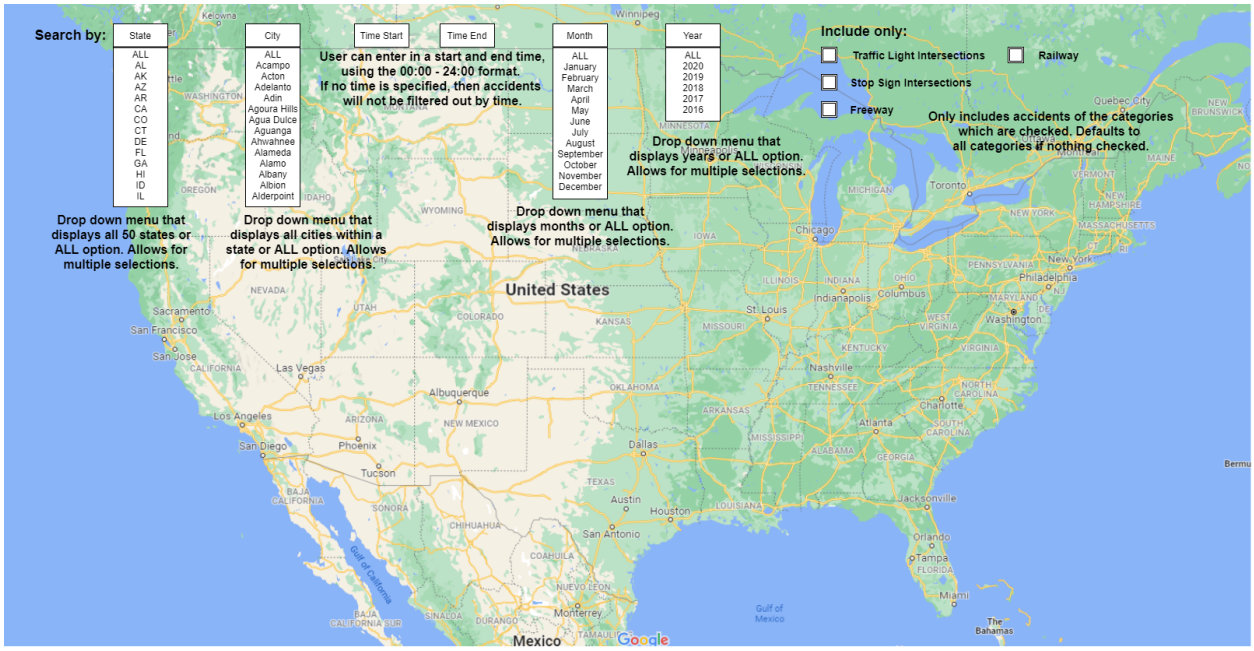
Features to implement in next Sprint:

- **Map**
  - **Feature 1**: Display a heat map of accidents using coordinates
- **Database**
  - **Feature 2**: CR(UD) operations of in-memory database.
- **GUI**
  - **Feature 3**: A search bar that can search for accidents by “State”, “City”, “Time Start”, “Time End”, “Month”, ”Year”

GUI Mock-up



Test Cases

- Feature 1 Test Cases: (Internal testing)
  - **Test Case 1**: Input a pair of coordinates.
    Correct Output: Map displays a red dot at the coordinate of the accident.
  - **Test Case 2**: Input an array of coordinates.
    Correct Output: Map displays multiple red dots corresponding to the coordinates.
  - **Test Case 3**: Input all coordinates from dataset.
    Correct Output: Map displays all 3 million red dots
- Feature 2 Test Cases: (Internal testing)
  - **Test Case 1**: Insert a new accident from the dataset into the database.
    Correct Output: The database returns the newly inserted accident.
  - **Test Case 2**: Read an accident by its ID.
    Correct Output: The database returns the accident with the ID.
  - **Test Case 3**: Read an accident by unknown ID.

Correct Output: The database should return 404.
    - **Test Case 4**: Query accidents by various parameters, for example: “state”, “city”, “time”, etc
      Correct Output: The database returns the array of accidents according to the query parameters.
- Feature 3 Test Cases: UI Input
    - **Test Case 1**: User selects individual search criterias from each drop down menu and/or inputs a valid date and time.
      Correct Output: Displays array of accidents at the given location, and within the given time if specified.
    - **Test Case 2**: User does not select anything from the drop down menus, but inputs a valid date and time.
      Correct Output: Displays all accidents that happened in the given time interval across the country.
    - **Test Case 3**: User does not do anything and press search.
      Correct Output: Displays all accidents from the database.
    - **Test Case 4**: User enters a wrong format of time.
      Correct Output: Displays a pop up warning telling the user the correct time format
    - **Test Case 5**: User selects multiple states, cities, months, and years.
      Correct Output: Displays array of accidents from all the selected locations during the specified dates.
- Feature 4 Test Cases: URL Address Bar
    - **Test Case 1**: User types in variables/values into the address bar, in which any amount of the input doesn’t exist in the database.
      Correct Output: Displays a pop up warning telling the user that the specified input is invalid.

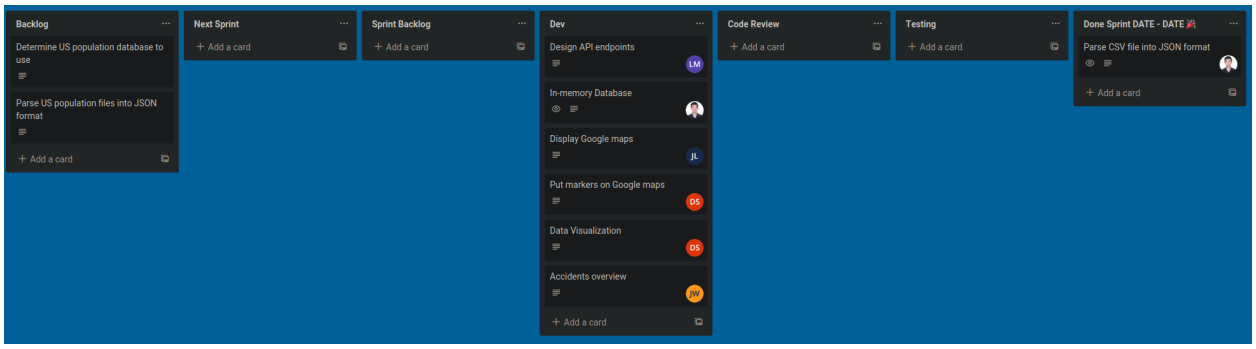
TODO LIST

Done list of last sprint

- Sending and receiving messages between server and client
    - [finished by everyone together]
- Parsing CSV file to JSON format
    - [Finished by Jiahe Gellert Li and verified by Luke McFadden]

Upcoming Sprint

- Backend storing dataset into in-memory data structure for better RAM efficiency.
- Backend returning accidents data according to query parameters.
- Frontend displaying heat map of accidents.
- Frontend displaying returned accident data.


Backlog
Determine US population database to use
Parse US population files into JSON format
Add a card
Next Sprint
Add a card
Sprint Backlog
Add a card
Dev
Design API endpoints
LM
In-memory Database
Display Google maps
JL
Put markers on Google maps
DS
Data Visualization
DS
Accidents overview
JW
Add a card
Code Review
Add a card
Testing
Add a card
Done Sprint DATE - DATE
Parse CSV file into JSON format
Add a card