Team: Caution

Features (user stories) to Implement in Next Sprint:

- **US Accidents:**
    - **Feature 1**: as a user, I want to be able to update the severity of a given accident record.
    - **Feature 2**: as a user, I want to be able to update the weather of a given accident record.
    - **Feature 3**: as a user, I want to be able to update the humidity of a given accident record.
    - **Feature 4**: as a user, I want to be able to update the bump of a given accident record.
    - **Feature 5**: as a user, I want to be able to update the timezone of a given accident record.
    - **Feature 6**: as a user, I want to be able to update the airport code of a given accident record.
    - **Feature 7**: as a user, I want to be able to update the pressure of a given accident record.
    - **Feature 8**: as a user, I want to be able to insert a new accident record.
    - **Feature 9**: as a user, I want to be able to delete an existing accident record.

Test Cases

- **Feature 1 Test Cases**: as a user, I want to be able to update the severity of a given accident record.
    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “severity” from the drop down, enter in the new severity value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the severity value of the specified record in the .csv file. The updated record is displayed.
- **Feature 2 Test Cases**: as a user, I want to be able to update the weather of a given accident record.
    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “weather” from the drop down, enter in the new weather value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the weather value of the specified record in the .csv file. The updated record is displayed.
- **Feature 3 Test Cases**: as a user, I want to be able to update the humidity of a given accident record.

    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “humidity” from the drop down, enter in the new humidity value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the humidity value of the specified record in the .csv file. The updated record is displayed.
- **Feature 4 Test Cases**: as a user, I want to be able to update the bump of a given accident record.
    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “bump” from the drop down, enter in the new bump value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the bump value of the specified record in the .csv file. The updated record is displayed.
- **Feature 5 Test Cases**: as a user, I want to be able to update the timezone of a given accident record.
    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “timezone” from the drop down, enter in the new timezone value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the timezone value of the specified record in the .csv file. The updated record is displayed.
- **Feature 6 Test Cases**: as a user, I want to be able to update the airport code of a given accident record.
    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “airport code” from the drop down, enter in the new airport code value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the airport code value of the specified record in the .csv file. The updated record is displayed.
- **Feature 7 Test Cases**: as a user, I want to be able to update the pressure of a given accident record.
    - **Test Case 1**: as a user, in the Update section, I enter in the ID of the record I want to update, select “pressure” from the drop down, enter in the new pressure value, and click on the “Update” button

      Correct Output: An update request is sent to the backend. The backend updates the pressure value of the specified record in the .csv file. The updated record is displayed.
- **Feature 8 Test Cases**: as a user, I want to be able to insert a new accident record.

- **Test Case 1**: as a user, in the Insert form, I fill out the attributes of the new accident record and click the “Insert” button

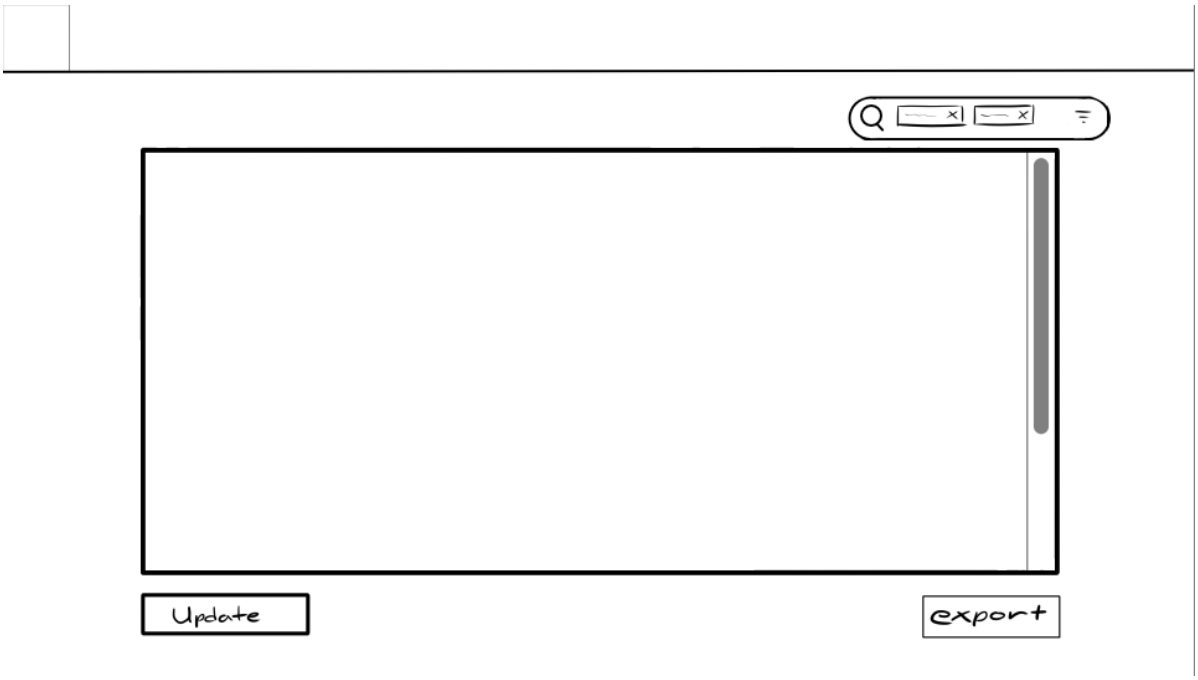
  Correct Output: An insert request is sent to the backend. The backend inserts a new record in the .csv file. The inserted record is displayed.

- **Feature 9 Test Cases**: as a user, I want to be able to delete an existing accident record.
  - **Test Case 1**: as a user, in the Delete section, I enter in the ID of the record I want to delete and click on the “Delete” button

    Correct Output: A delete request is sent to the backend. The backend the specified record in the .csv file. The deleted record is removed from display.

Design:

- **Search Page:**



TO-DO LIST:

Done list of last sprint:

- UI to search for specific filters on the data
  - Finished by Kenny and verified by everyone
- Frontend functionality to add multiple filters or remove multiple filters at a time - multiselect of features
  - Finished by Kenny and verified by everyone

- Client-server communication between search buttons and the server to let the server know what queries to search for in the CSV
    - Finished by Albert and Danial and verified by everyone
- Frontend function to take an array of JSONs and display it in a table
    - Finished by Rahul and verified by everyone
- Front End styling for table and multiselect
    - Finished by Rahul and Kenny and verified by everyone
- Parsing of CSV file using RegEx
    - Finished by Matthew and verified by everyone
- Storing parsed data into a 2D array (array of arrays where an individual array held a row of records and the big array overall stored all the rows)
    - Finished by Matthew and verified by everyone
- Replacing blank values in the CSV with void characters in order to correctly parse through all data
    - Finished by Matthew and verified by everyone
- Functions to receive request from client and do filtering in the large 2D array based on input from the client
    - Finished by Danial and verified by everyone
- Converting array of arrays in the final filtered data to an array of JSONs to make displaying easier
    - Finished by Danial and verified by everyone
- Adding in column labels in the table view to make it easier to identify what data is what
    - Finished by Danial and verified by everyone

To-Do for next sprint:

- Front-end works in order to update the table
    - Acceptance Criteria: When clicking the add or delete row button we are able to see from the table the updates to the row which were made on the CSV
- A new CSV has been generated and updated with the row updates
    - Acceptance Criteria: Can add or delete rows to the 2D array, which is then used to update a new CSV
- Add a way to break up multiple fields of data being shown on one page - have multiple “pages” of data that show at maximum X number of rows
    - Acceptance Criteria: Large amounts of data are split into multiple pages to ease the load of display and make visualizing a large dataset easier

- The new CSV can be backuped/imported
    - Acceptance Criteria: The server displays that the new CSV has been posted