BBCDS Artifact

Platform: Web App
Languages: Javascript, Typescript, HTML, CSS
Technologies:

- Next.js (Frontend)
- Node.js (Backend)
- Chart.js (Data Visualization)

Dataset:

Novel CoronaVirus 2019 Dataset (Link)

1. Number of cases confirmed by country, date.
2. Number of cases in the U.S. by state, date.
3. Number of deaths by country, date.
4. Number of deaths in the U.S. by state, date.
5. Number of recovered by country, date
6. All cases by location, age, gender, date, symptom.

Fields of interests for the features

1) Are there any particular areas where COVID-19 is most vulnerable?
2) What age range is more prone to contracting the virus?
3) Have the number of deaths decreased since the beginning of this year?
4) How many new cases are increasing in the U.S, by state and globally every day based on the user's choices ?
5) Charts that show confirmed cases, number of deaths, recovered cases, number of new cases, respectively.
6) Global mortality rate

Fields of interests to search:

- Search by **age** of previous confirmed corona cases
- Search by **sex** of previous confirmed corona cases
- Search by **country** of previous confirmed corona cases
- Search by **city** of previous confirmed corona cases
- Search corona cases in specific **date range**

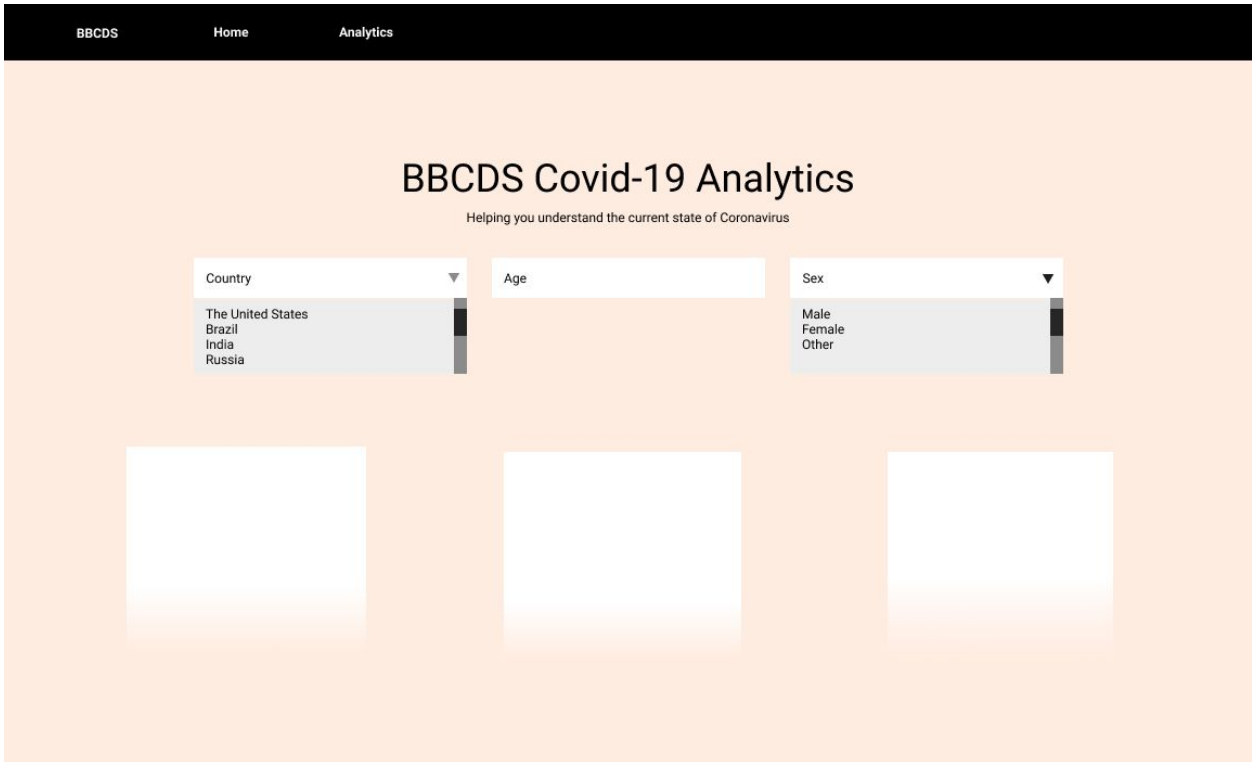
Features:

1. **10/08 Sprint-1 Feature**
    a. Basic Layout of the website
    b. Two page (Home page and analytic page)
    c. NavBar in both page on the top
    d. Front-end and Back-end communication
2. **10/15 Sprint-2 Feature**
    a. The user can select the cases they want to see by country, age, sex.
    b. Once users confirm their choice of categories, they can submit the choices.
    c. The backend server returns data based on the parameter the user chose.
    d. Based on the return data, display a data table and graphs visualization for the result.
3. **10/22 Sprint-3 Feature**
    a. The user can select more categories other than country, age, sex in home page.
    b. An admin page that shows the table that contains all the data from the dataset.
    c. The user can do modification such as insert new cases, edit existing cases and delete cases.

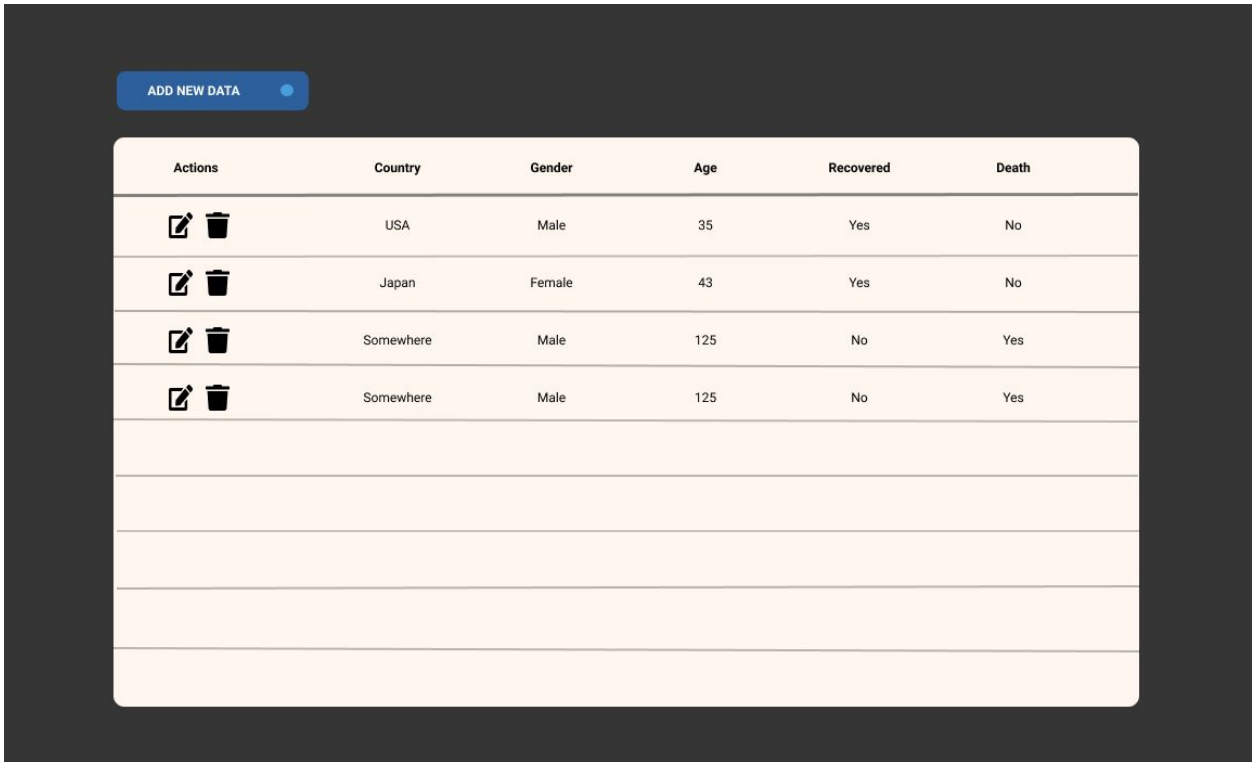
GUI design:

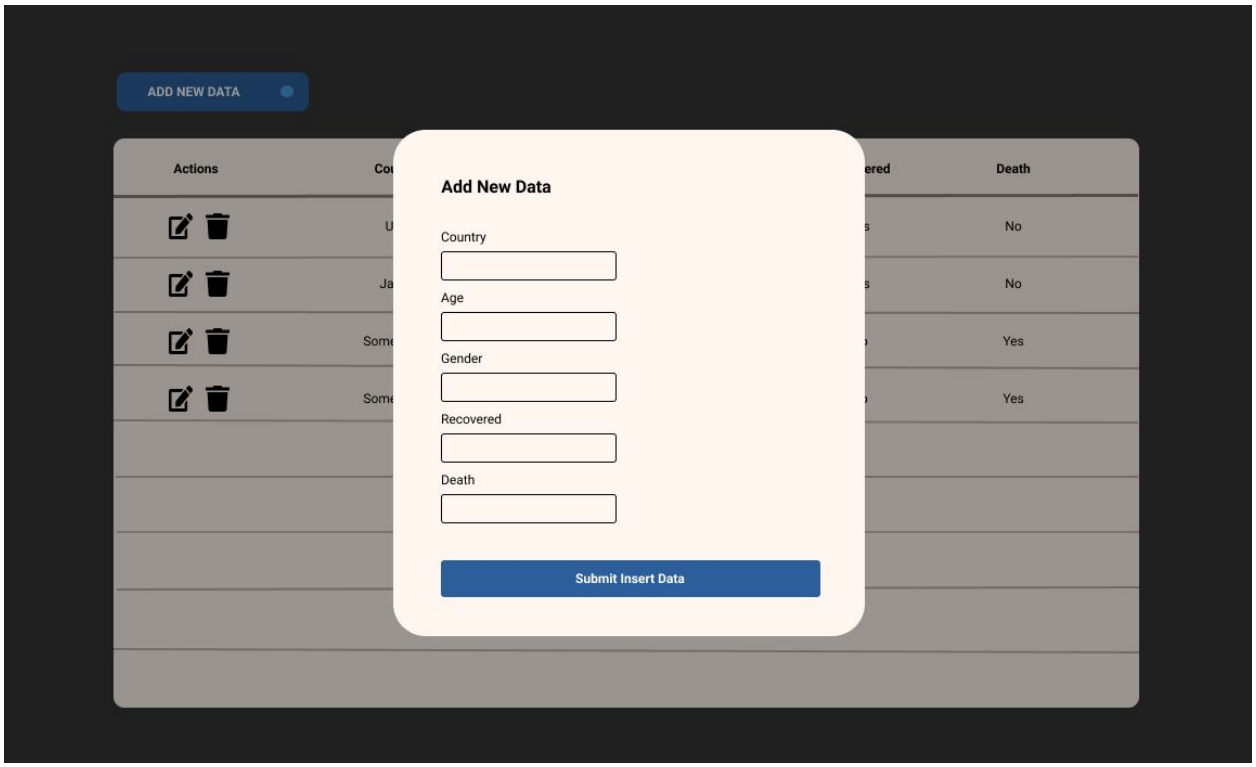
1. **10/15 Sprint**
    a. Input field for country, age, and sex
    b. Autocomplete feature for country input field
    c. A dropdown list of searchable sex options
    d. Shows confirmed, death, recovered number under the Input field

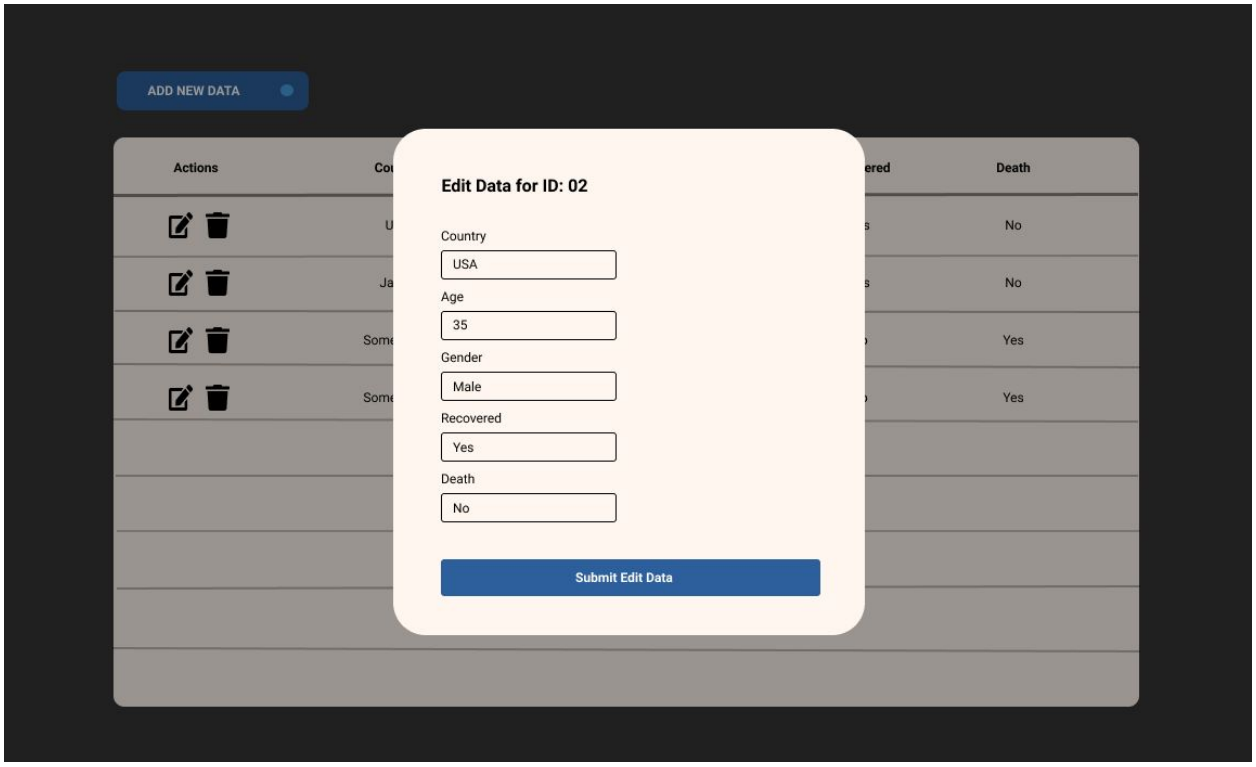
UI Design for Search operation for multiple fields in the data

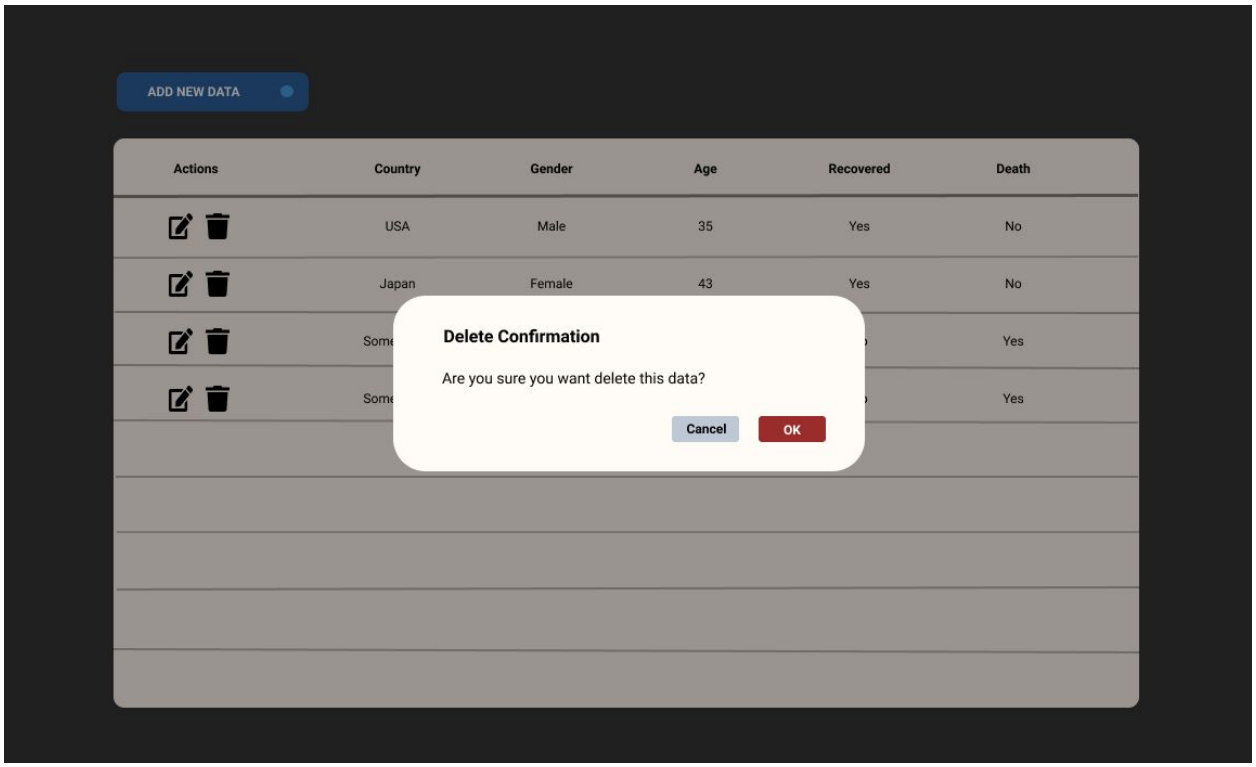


2. **10/22 Sprint**
    a. Add New Data button
    b. Table of all data
    c. Edit and Delete Buttons for each row of data


ADD NEW DATA
Actions
Country
Gender
Age
Recovered
Death
USA
Male
35
Yes
No
Japan
Female
43
Yes
No
Somewhere
Male
125
No
Yes
Somewhere
Male
125
No
Yes


ADD NEW DATA
Actions
Death
No
No
Yes
Yes
Add New Data
Country
Age
Gender
Recovered
Death
Submit Insert Data





User test cases:

1. **10/15 Test Cases**

a. As a user I should be able to tell which input field queries for what particular piece of data.
    i. Correct Output: Add labels inside the text fields and default values
b. As a user sometimes I don't want to be required to fill in all of the search input boxes.
    i. Correct Output: The search will only send what is inputted and the backend will only filter through based on what is sent. Empty inputs will be ignored as part of the filter
c. As a user I want to see data pertaining to the search I just requested in an easily digestible format.
    i. Correct Output: Charts and tables will be added to assist in data visualization
d. As a user I want to know if my submission went through
    i. Correct Output: Add an animated loading spinner while the backend filters through the data.

2. 10/22 Test Cases

a. As a user I accidentally click the delete button for a data point
    i. Correct Output: Deleting data will come with a popup confirmation modal
b. As a user I don't have all the data for all the possible inputs in the insert modal.
    i. Correct Output: Only certain fields will be required to make sure the dataset is filled with relevant data points.
c. As a user I don't want to have to search through thousands of data rows to find what I want to delete or edit
    i. A search box will be above the table to easily filter through the table in realtime
d. As a user I want to edit the data into all blank fields.
    i. Correct Output: Certain fields will be required to make sure the dataset is filled with relevant data points.

Taskboard:

1. 10/08 Sprint Done List

a. Finish Sprint Artifact Planning
(finished by Everyone)
b. Setup web and api servers
(finished by Dominic Lee)
c. Setup web request to the api server
(finished by Dominic Lee)
d. Upload Demo video
(finished by Biqian Cheng)

2. 10/15 Sprint Done List

a. Finish Sprint Artifact Planning
(finished by Everyone)

b. Add csv file of datasets. And create an api route that returns parsed csv files into an array of JSON objects.
(finished by Dominic Lee)
c. In the homepage add an input field for country, age, and sex
(finished by Biqian Cheng)
d. Once all search fields have content and have been validated for user errors, the search should submit automatically.
(finished by Shuang Zhou)
e. On search submit, query the backend for a response that returns all cases that match the search query.
    i. Ex. Search for country [United States], age [32], sex [Male] returns all 32 year old males that were confirmed with corona virus in the United States.

(finished by Shuang Zhou)
f. Display data in a visual format. Tables & charts.
(finished by Dominic Lee)
g. Upload Demo video
(finished by Biqian Cheng)

3. 10/22 Sprint Done List

a. Finish Sprint Artifact Planning
(finished by Everyone)
b. UI design for admin page
(finished by Biqian Cheng, verified by Dominic Lee)
c. Add more input fields for different categories in the home page such as status(confirmed, recovered, death).
(finished by Biqian Cheng)
d. Add a new admin page and a title showing what page you're on.
(finished by Dominic Lee)
e. Request all data from the parsed dataset on the backend api and use that data to populate a paginated table display the data
(finished by Dominic Lee)
f. Add insert data feature which includes an insert button, insert popup modal, and POST request to the backend upon insert submit.
(finished by Biqian Cheng)
g. Add an Edit and Delete icon button in each row of the data table.
(finished by Shuang Zhou)
h. Add an Edit popup modal displaying text fields for all editable data fields and a submit button that calls a PUT request to the backend
(finished by Shuang Zhou)
i. Create a DELETE request to the backend on delete icon click.
(finished by Shuang Zhou)
j. Write backend route for inserting (POST Request) new data
(finished by Dominic Lee)
k. Write backend route for delete data (DELETE Request)
(finished by Dominic Lee)

l. Write backend route for editing data (PUT Request)
   (finished by Dominic Lee)
m. Upload Demo video.
   (finished by Biqian Cheng)