

**Team Eclipse:** Amanda Zheng (PM), Tiffany Cao, Junhee Lee, Yifan Wang

**Period:** 1

**Assignment:** P04 - Let the Data Speak

**Due Date:** 5/4/2020

**Title:** Covid-19 Tracker

### **Individual Responsibilities**

Amanda Zheng (PM):

- Bar Charts/Pie Charts
- Welcome Page

Tiffany Cao (Frontend):

- Styling
- Creating the Database Layout

Junhee Lee (Backend):

- Regulating/ Controlling the movement of Data from Frontend to Backend
- API Management

Yifan Wang (Frontend):

- Data Querying Page
- Time Slider

### **Summary**

Our Covid-19 Tracker aims to give a comprehensive overview of the day-to-day changes of the spread of the coronavirus. Our focus is on a global scale, as well as a concentration on the United States. Starting from January to April 2020, the tracker can report visually the number of total cases, recoveries, and deaths per day on a map. The tracker also has an interactive, exploratory interface for users. They have the option of choosing which data they want to be included (they have the three choices listed before) and which countries or states they want to compare. The data for each state and/or country is represented with either a bar chart or a pie chart depending on the options chosen. Hovering over these charts will have a pop-up that shows information and numerical data. Sliders will be available for users to choose the dates, but a continuous, automatic animation of the daily changes can also be presented.

### **Project Timeline**

\*strikethrough == complete

Minimum Viable Product:

- Fifty states of the United States of America
- User can choose to view new cases, deaths, or recoveries of COVID-19 per day separately

- User can choose the day they wish to view
- Users can select multiple states' data to view at once

#### Ideal Product:

- Data of additional countries available (10 other countries)
- Have a pop-up when hovering over charts with more information and numerical data
- Time Slider
- User can choose to compare two or more components (out of new cases, deaths, and recoveries)
- User can choose multiple states/countries

#### Extra Features:

- Have a map of the United States with pie charts placed accordingly with the states
- Have a map of the world with pie charts placed accordingly with the countries
- Have different modes of visualization that the user can choose
- More countries data

#### **API**

- Country API: to obtain the flags to display next to the data for each country

#### **Front-end**

We would provide the user with the choice of looking at data specific to either the United States or multiple countries around the world and which type of data they want to be displayed. After selecting the data and pressing the “render” button, they would go to a page where all the data are displayed. If the user selected all three options (new cases, recoveries, and deaths), then a pie chart would be displayed for that specific state or country. If the user selected only one/two out of the three, a bar graph would be displayed comparing all the states/countries selected. We would include a slider at the top of the page where the user can see how the data changed across different dates. There would also be a “continuous” button when clicked would automatically animate the graphs, looping through all the dates available.

#### **Back-end**

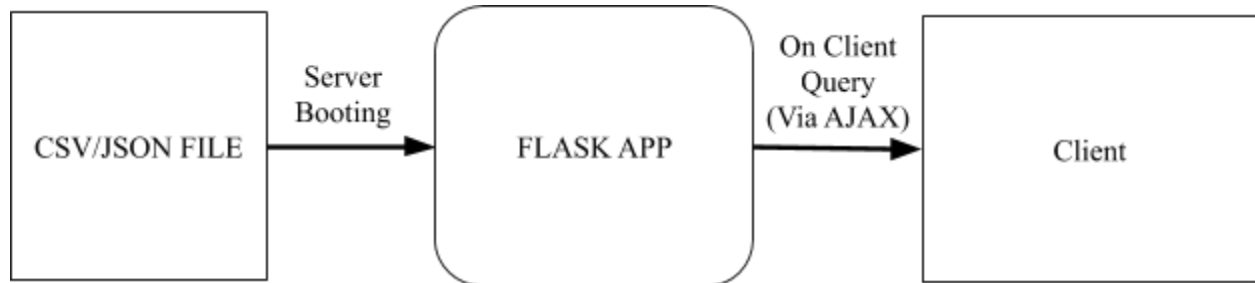
When the user initially visits the site, the backend returns a page with selections of the various datasets that we have from which the user can choose. When the dataset is queried through AJAX, the flask app filters the data and sends the appropriate data to the client.

#### **Frontend Framework - Bootstrap**

We chose Bootstrap over Foundation as our frontend framework because we like the appearance of the aesthetics more. It has more variability in its designs, giving us more freedom to customize

our project. We also have more experience with Bootstrap, so we are more comfortable utilizing it.

### Journey of the Data (Diagram):



### Database Layout

#### United States Data

- Each state will have three lists: one for total number of cases, one for total number of deaths, and one for the total number of recoveries. The indexes of these lists represent the dates, which we will keep in mind.
- All of the states' data will be stored as a large dictionary for the United States.

U.S. = [

Random State: {

Total Cases: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...],

Total Deaths: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...],

Total Recoveries: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...]

} ...

]

Excerpt of Table Representation for Random State:

	4/1	4/2	4/3	4/4	4/5	4/6
Cases	100	200	300	400	500	600
Deaths	12	34	33	40	50	67
Recoveries	1	2	3	4	3	10

#### Country Data

- Same layout as state data

	4/1	4/2	4/3	4/4	4/5	4/6
--	-----	-----	-----	-----	-----	-----

Cases	100	200	300	400	500	600
Deaths	12	34	33	40	50	67
Recoveries	1	2	3	4	3	10

Country A: {

Total Cases: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...],

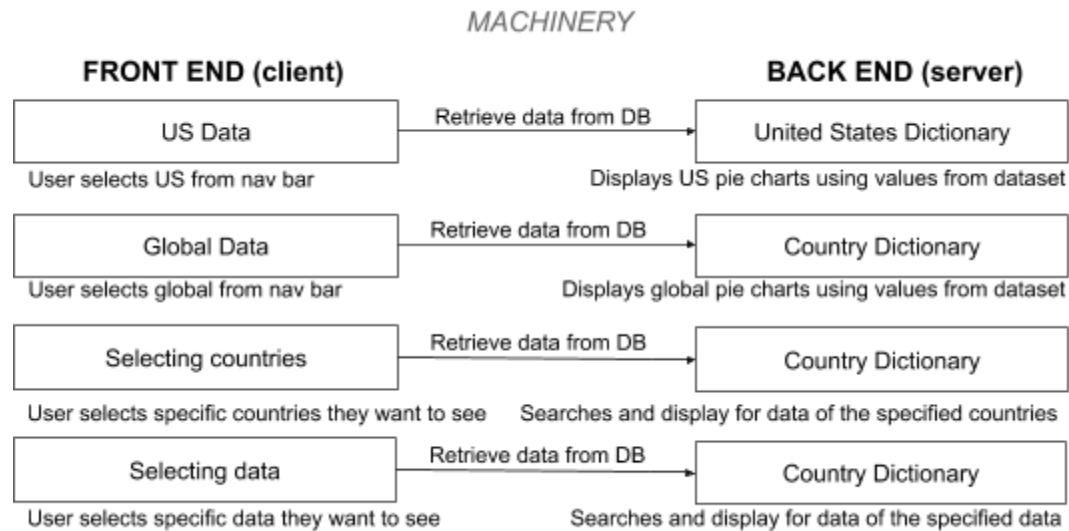
Total Deaths: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...],

Total Recoveries: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, ...]

}

\*Disclaimer: These are all lists/dictionaries, and not in a real database.

## Component Map



## Site Map

# Site Map

