**Team Eclipse:** Amanda Zheng (PM), Tiffany Cao, Junhee Lee, Yifan Wang
**Period:** 1
**Assignment:** P04 - Let the Data Speak
**Due Date:** 5/11/2020

**Title:** Covid-19 Tracker

**Individual Responsibilities**
Amanda Zheng (PM):

- Pie Charts Transition
- Animation Effects
- Welcome Page & Check Box and Event Coordination

Tiffany Cao (Frontend):

- Single Bar Charts Rendering and Transition
- Double Bar Charts Rendering and Transition
- Data Querying Page

Junhee Lee (Backend):

- Regulating/ Controlling the Movement of Data from Frontend to Backend
- Data Processing
- Slider coordination with Data

Yifan Wang (Frontend):

- Styling
- Pie Chart Rendering
- Pop Up Labels

**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 closer look 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 multiple graphs. 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 some 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~~
- Automatically sorting countries/states based on length of bar in bar chart

**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 selects 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 "transition" button when clicked would automatically animate the graphs, looping through all the dates available.

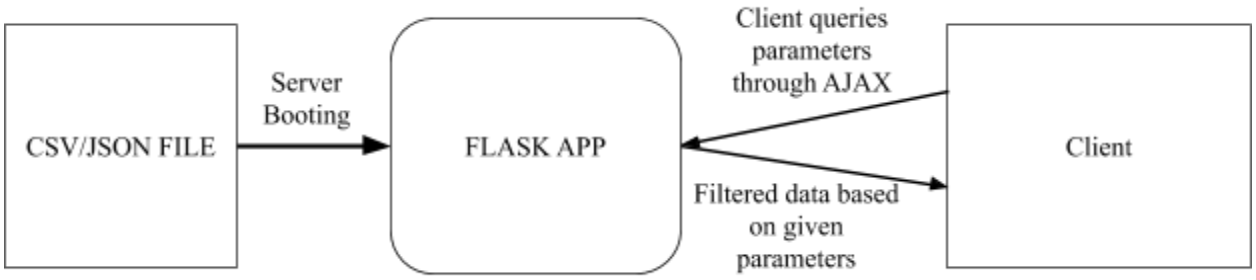
**Back-end**

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 parameters are passed to a flask route via AJAX and the slider value is given, the flask app filters the data, processing it into a dictionary consisting of the keys 'location', 'cases', 'recoveries', and 'deaths', 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-date combination has one entry in the list; although most of the data is not used by our site, they are vestigial structures from the csv file we imported.
- The data is stored as a list of lists.
  The relevant indices for the inner lists are date (0), state abbreviation (1), cases (2), recoveries (11), and deaths (14)
- Dates are stored as datetime.date objects

States = [

...

[datetime.date(2020, 01, 21), 'NY', 0, … , 0, … 0, ...]

...

]

Table Representation

| Date | State | Cases | ... | Recoveries | ... | Deaths |
|---|---|---|---|---|---|---|
| datetime.date(...) | NY | 14000 | ... | 50 | ... | 1000 |
| datetime.date(...) | WO | 240 | ... | 10 | ... | 50 |

## Country Data

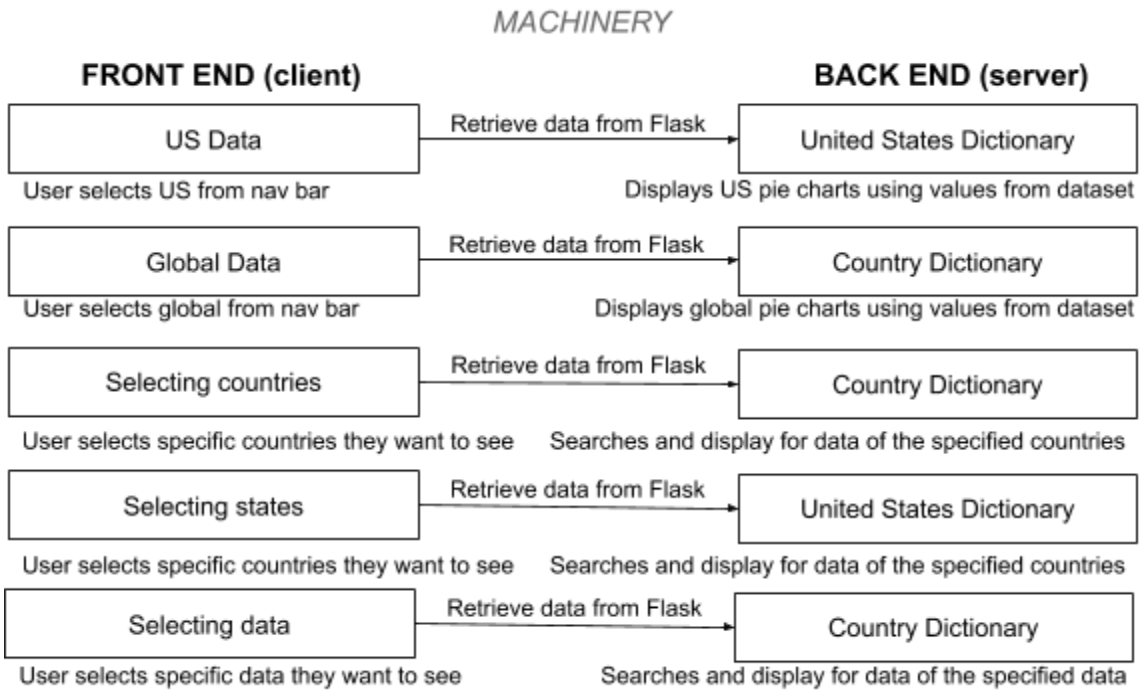
- Similar layout to State
- Relevant indices are date (0), country (1), cases (5), recovered (6), deaths (7)

| Date | Country | ... | Cases | Recovered | Deaths |
|---|---|---|---|---|---|
| datetime.date(...) | United Kingdom | ... | 500 | 100 | 15 |
| datetime.date(...) | China | ... | 10000 | 1 | 1500 |

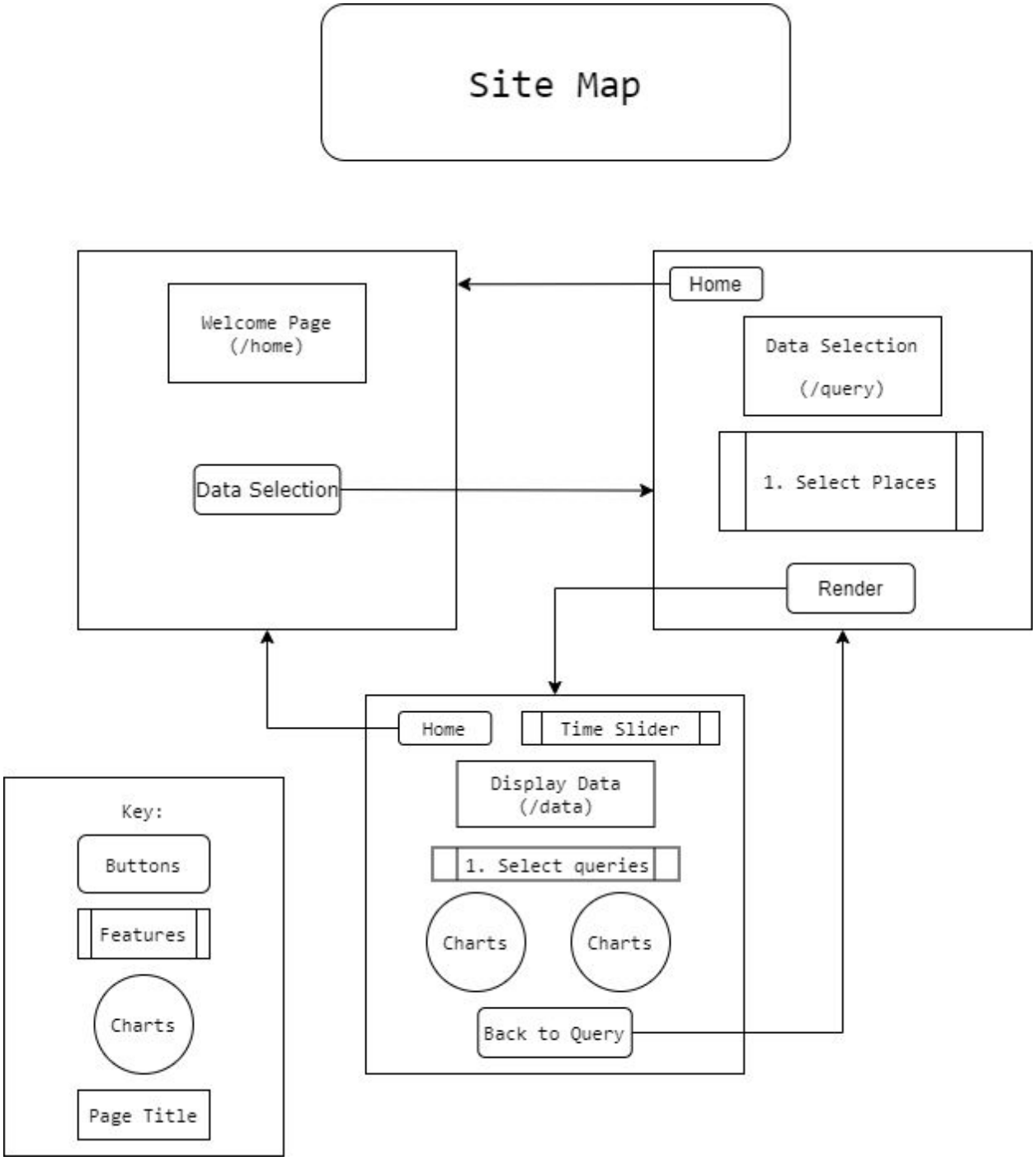
```
countries = [
        ...
        [datetime.date(2020, 05, 10), 'US', … , 150000, 100, 2000]
        ...
]
```

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

**Component Map**



**Site Map**


Site Map
Welcome Page
(/home)
Data Selection
Home
Data Selection
(/query)
1. Select Places
Render
Home
Time Slider
Display Data
(/data)
1. Select queries
Charts
Charts
Back to Query
Key:
Buttons
Features
Charts
Page Title