

## TEAM NAME: Mykolyangelo

**ROSTER:** Grace Mao (PM), Tammy Chen, Jun Tao Lei, Jackson Zou

**TITLE:** World Climate Change Analysis

### FRONTEND FRAMEWORK: Bootstrap

Bootstrap and Foundation both offer similar resources and abilities in terms of page organization, so because our group members are most familiar with Bootstrap, it was clear that we would lose nothing by choosing the one that we are most comfortable with.

### DESIGNATED ROLES:

Project Manager: Grace Mao

Frontend: Grace Mao, Tammy Chen

Backend: Jun Tao Lei, Jackson Zou

Project Manager: Grace Mao

- Check in with Mr. Mykolyk when necessary
- Organize the dataset to its essentials and sort out exceptions
- Split up work fairly and efficiently
- Update design doc and keep repo in check
- Transitions in D3 to add a line graph in country visualizations

Database Management: Jun Tao Lei

- Set up querying and management of data by pulling from CSV
- GraphQL interaction with Fetch API
- Blueprint connections with app via \_\_init\_\_.py
- Country ISO code comparison to fix exceptions within two datasets
- Interactive world globe set up complete with country selection

Data Retrieval: Jackson Zou

- Create queries to retrieve data for visualizations
- Easily display results on the page
- Assist in CSV management and filtration to the essential elements
- World line graph rendering and world statistics displayed on homepage

Frontend Setup: Tammy Chen

- Bootstrap usage and homepage organization
- HTML templates complete with Jinja and linked to Javascript files
- Accessible and easy-to-understand user interaction with our app
- Country line graph rendering and statistics displayed on specific page

## **CLIMATE CHANGE: IS IT REAL?**

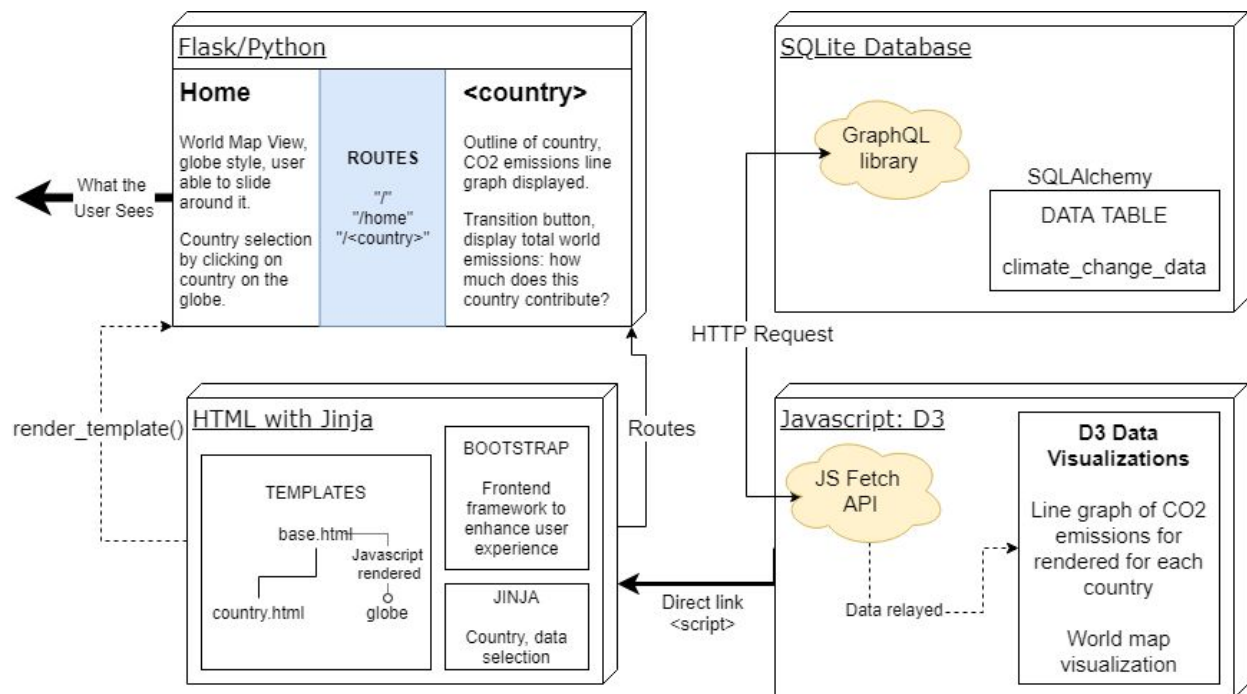
Our team's goal is to create a proper visualization for CO2 emissions and other climate change data we found on an international level. Our data is taken from the World Bank, and lists the greenhouse gas emissions in different countries over the years starting from 1960 to 2018. Find our data here: <https://data.worldbank.org/topic/climate-change>

With international data, the first visualization that users will see is a world globe. By scrolling or clicking around it, they'll be able to change the orientation and see different countries. There will also be displayed statistics about the world data, like total CO2 emissions and population. From here on, users will now explore different countries' individual data. To view a specific country, users can just click on a country on the globe.

After selection, the app will transition to just that certain country/region, and the page will render a line graph for gas emissions for that country over time. Basic statistics will be displayed (i.e population). Users will then be able to click to transition, in which it will show them the world total for these emissions layered on the same line graph. This will show how each country's emissions contribute to the world total, as some countries have a bigger influence than others; it puts things into perspective in terms of country size and population.

NOTE: With the use of topojson for the globe, we are using country ISO codes to transition between topojson and our dataset to avoid any spelling or naming convention differences. Countries that appear in topojson but not in our dataset (ex: Taiwan, small islands etc) will not be interactive on the world globe.

## COMPONENT MAP

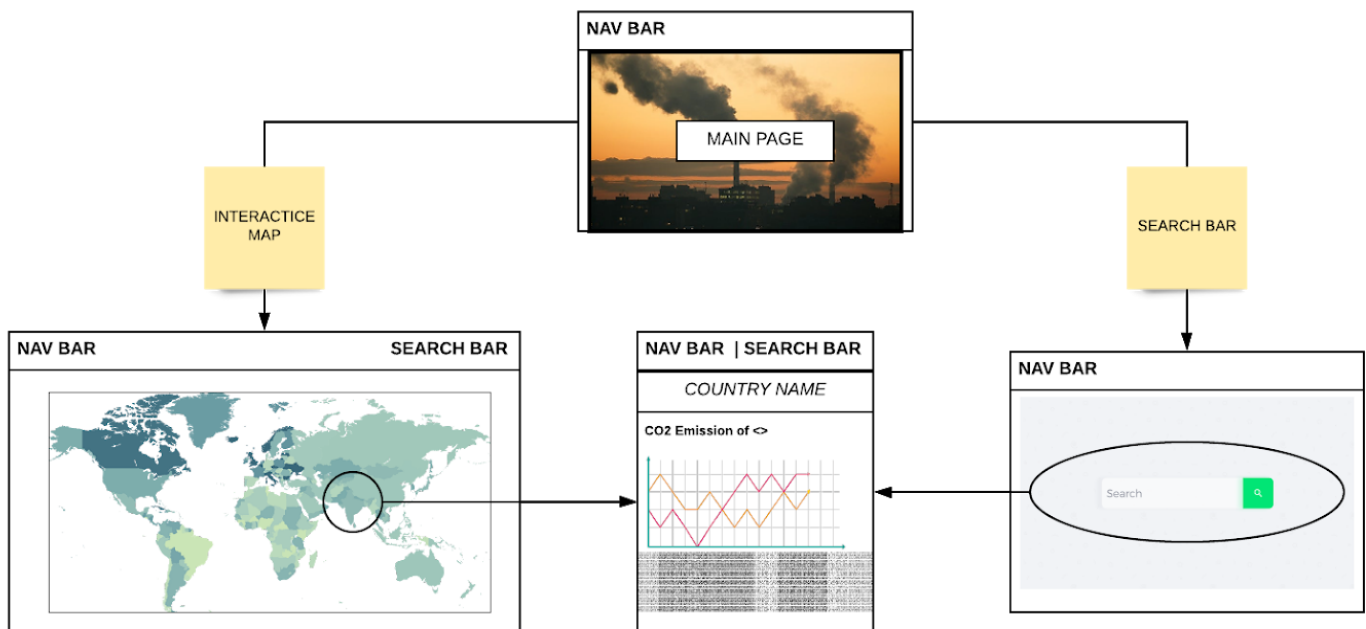
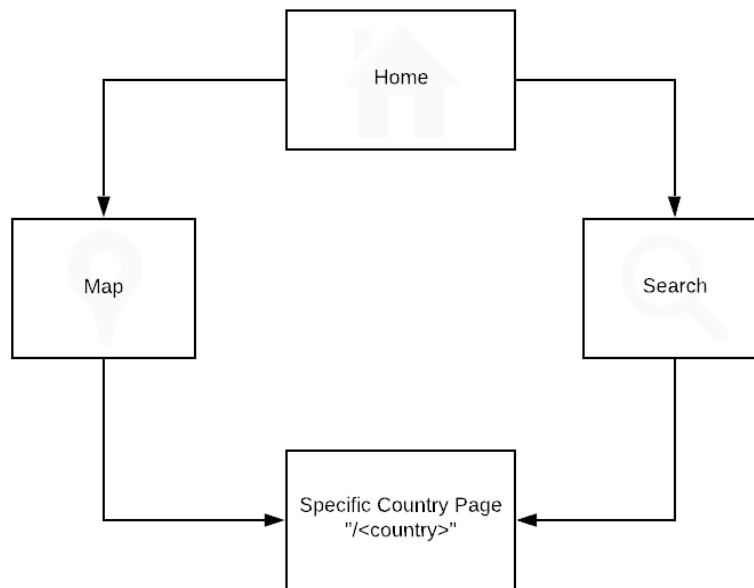


## DEVELOPMENT STAGES

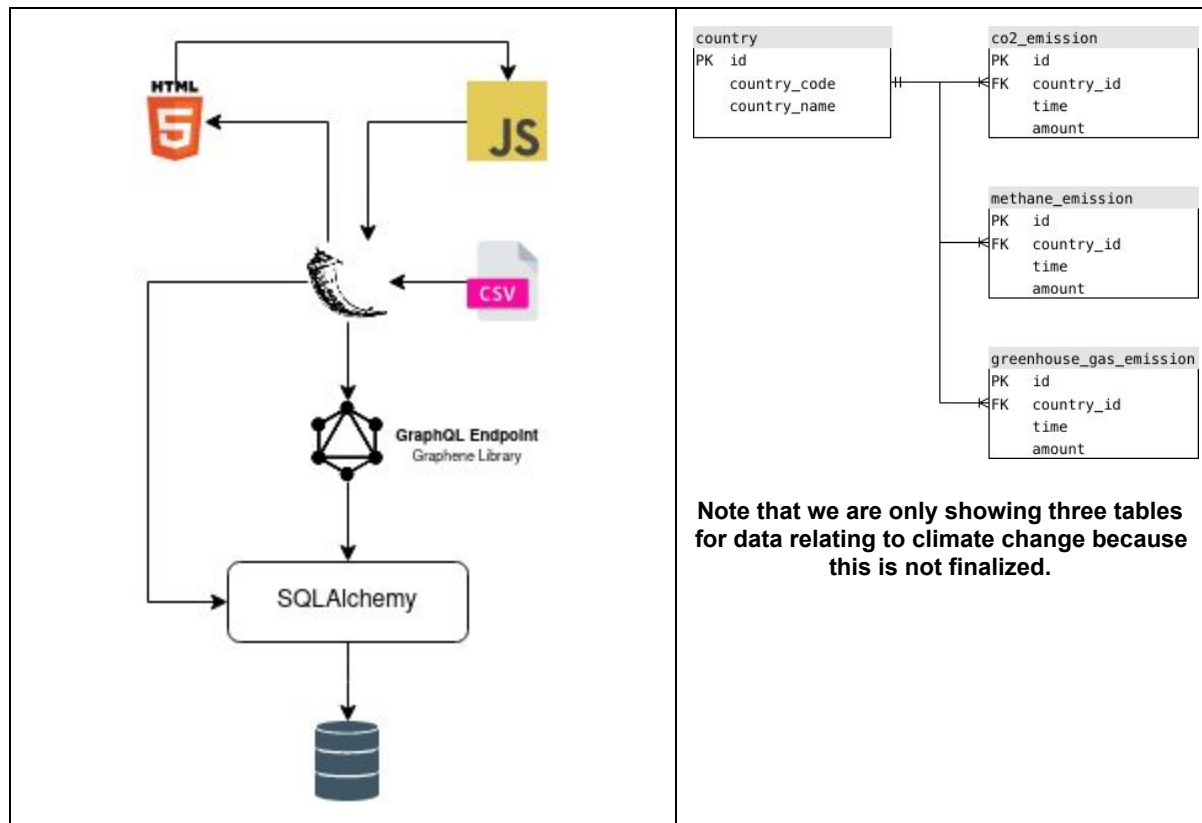
- 1) Parse data in CSV files ----- **Monday 5/4**
  - a) Move data into SQL database/table
  - b) Organize data based on relationships (country, year, etc.)
- 2) Create HTML templates and all flask endpoints ----- **Monday 5/4**
  - a) Figure out what's being displayed on different pages
  - b) Have placeholders where we will want to display information in the future
- 3) Pull information from databases on to web pages ----- **Wednesday 5/6**
  - a) Set up country selection
  - b) Display the basic statistics for each country: population, GDP, income etc
- 4) Frontend customization ----- **Sunday 5/10**
  - a) Build charts and graphs (D3)
    - i) World globe most complicated
    - ii) Line graphs of country and world ----- **Friday 5/8**
    - iii) Displayed statistics ----- **Thursday 5/9**
    - iv) Transition from country line graph to add world line
  - b) Beautify with Bootstrap and reorder elements if necessary

## SITE MAP

<https://www.lucidchart.com/invitations/accept/a4c40bee-4912-45b6-b7d5-8ea2027127a5>



## DATA ORGANIZATION



For this project, we would be importing a CSV file to a SQLite database. We would use SQLAlchemy as an object-relational mapping tool. This would allow us to quickly define one-to-many relationships and query the database.

In order to get data from the database, the front end JavaScript will make a HTTP request to the GraphQL endpoint under /graphql through the JavaScript Fetch API. This would be better than loading data directly from Flask/Jinja as we could customize the query to give us the data we want rather than all the data.

Our world map data will be taken from topojson, and the country names will be adjusted and filtered between the two datasets to correlate each country to its correct data.

An example of a global map using topojson can be found here:

<https://jorin.me/d3-canvas-globe-hover/>

Here is available source code for the package:

<https://github.com/topojson/world-atlas>