

## **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

- HTML
  - Base web page template
  - Bootstrap will be used as the frontend CSS framework
- Javascript
  - D3 will be used for building graphs/charts for visualization of information

Backend: Jun Tao Lei, Jackson Zou

- Database Management
  - Parse data from csv and categorize it properly to be stored in SQL.
  - SQL will be used for ease of access and speed
- Data Retrieval
  - Javascript fetch will be used to access information in databases
  - This data will need to be loaded and joined into D3

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

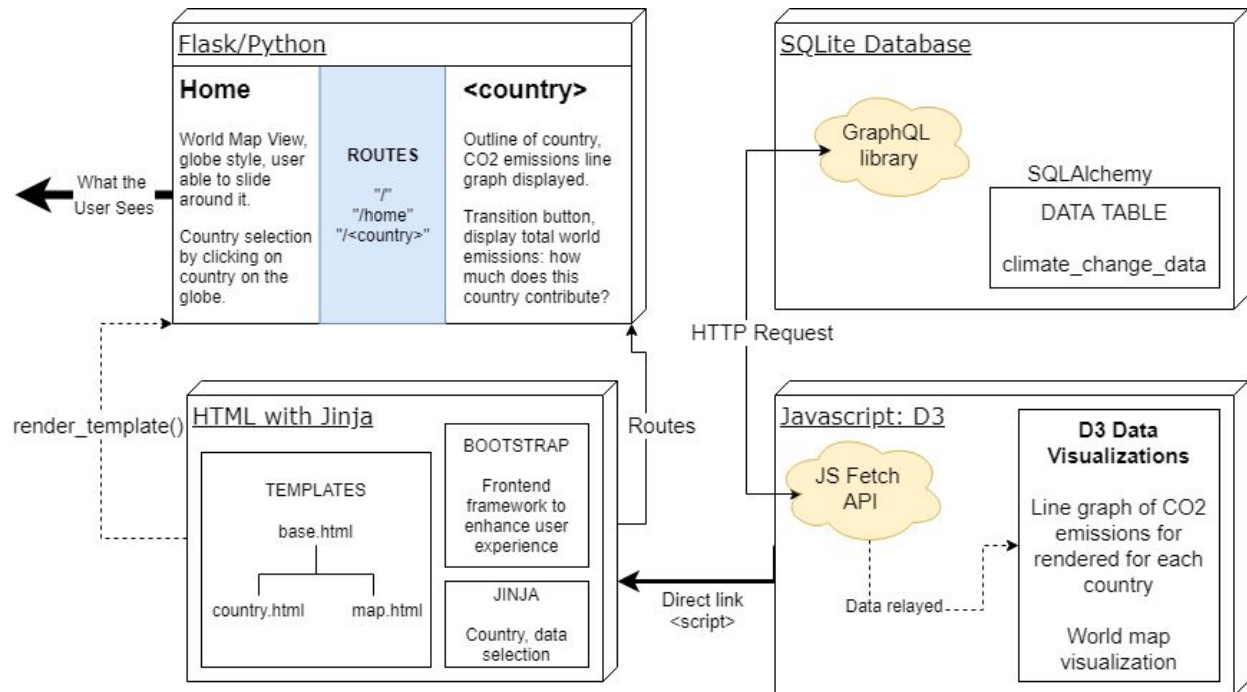
Our team's goal is to create a proper visualization for CO2 emissions and other climate change data we found on an international level. Our data is taken from the World Bank, and lists the greenhouse gas emissions in different countries over the years starting from 1960 to 2018. Find our data here: <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. 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. With extra time, we may also implement a random button that leads them to a random country in case users just want to explore.

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.

## COMPONENT MAP

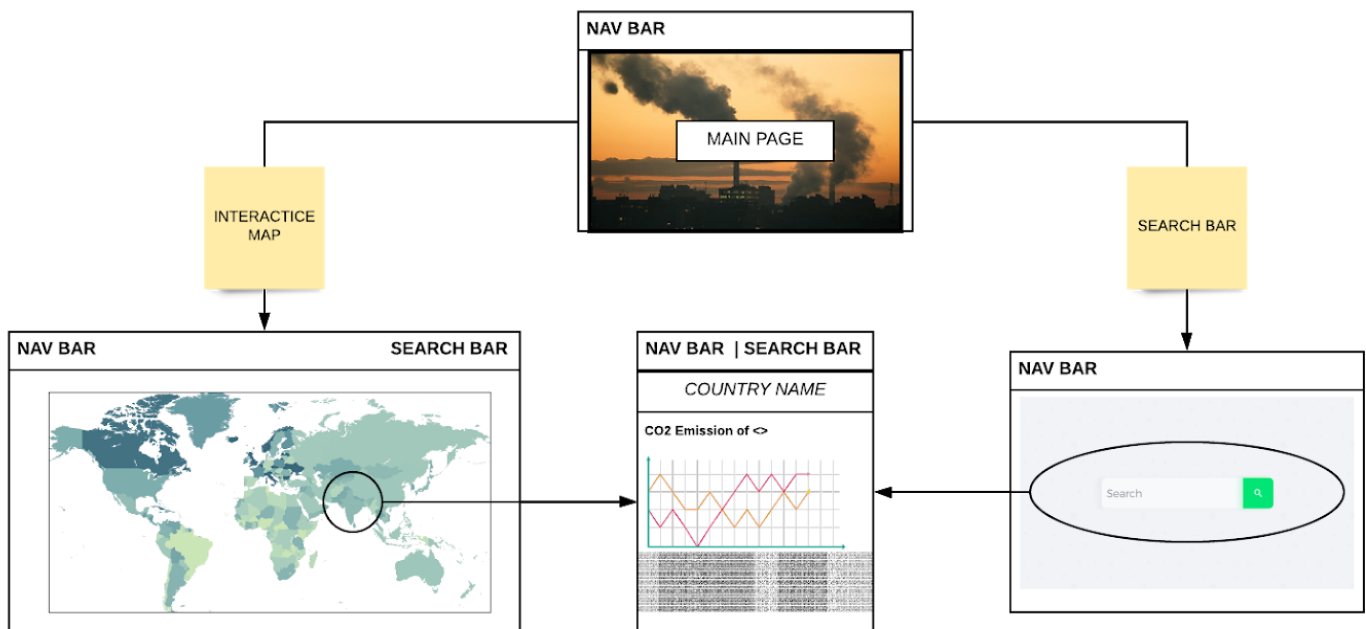
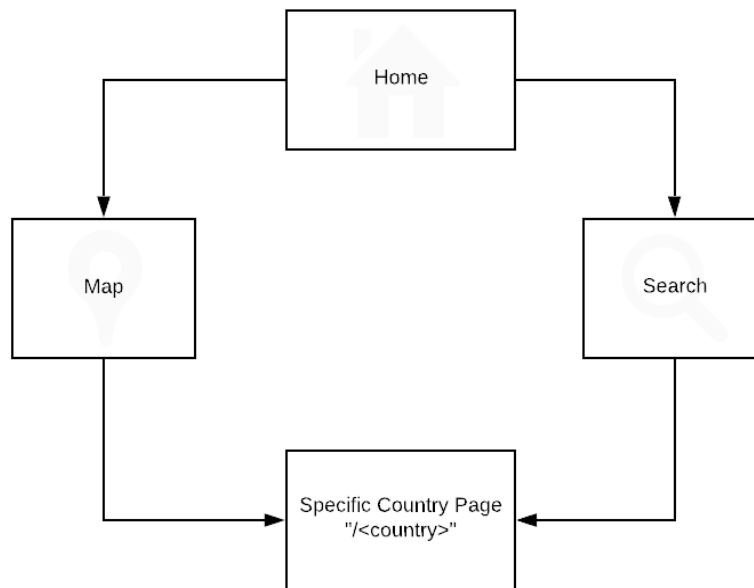


## DEVELOPMENT STAGES

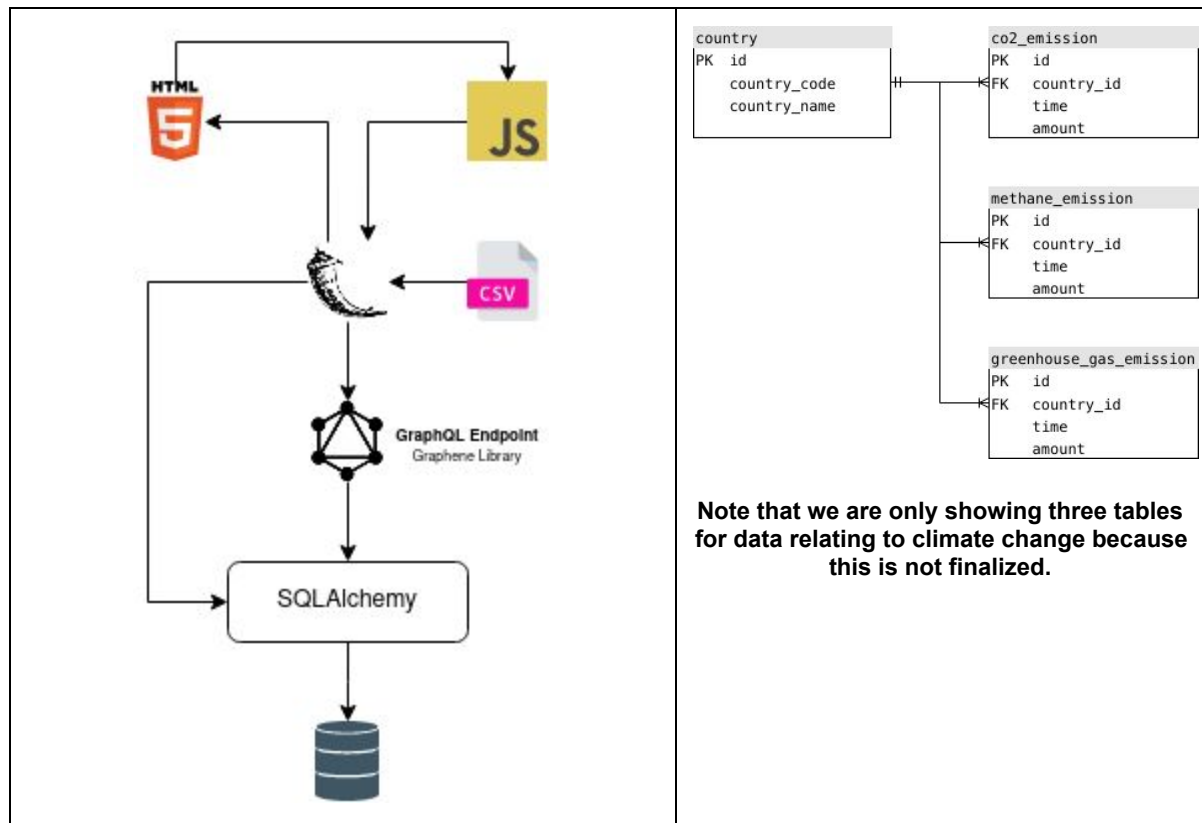
- 1) Parse data in CSV files ----- **Monday 5/4**
  - a) Move data into SQL database/table
  - b) Organize data based on relationships (country, year, etc.)
- 2) Create HTML templates and all flask endpoints ----- **Monday 5/4**
  - a) Figure out what's being displayed on different pages
  - b) Have 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
  - c) No D3 needed at the moment, make sure that information pulled is accurate
- 4) Frontend customization ----- **Sunday 5/10**
  - a) Build charts and graphs (D3)
  - 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>