Project 3
Group 9
Tesla Superchargers
August 15, 2024

Steven Madden
Amber Venes
Mrunmai Gadbail
Alexandra Reid

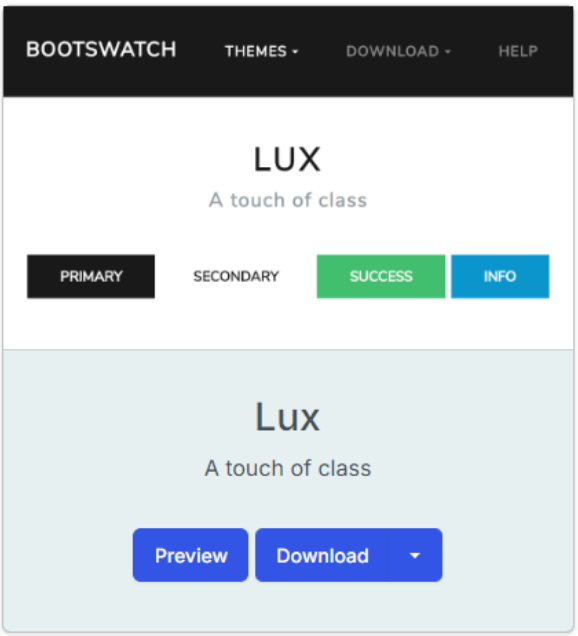
Tesla Supercharger Writeup

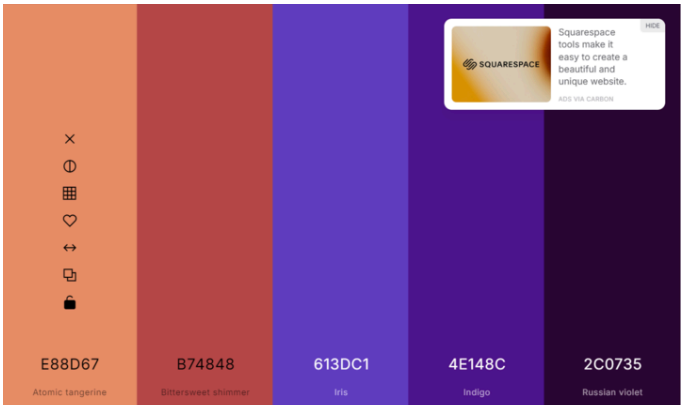
Technology is constantly advancing at an impressive rate and this can be seen within our own communities. This is especially evident in our vehicles that we drive everyday. Tesla is an incredibly successful tech company that has entered the automotive market and challenged the “everyday driver” with their electric powered vehicles. Tesla vehicles have become impressively popular across the U.S and even other countries across the world. So much so, that they have created supercharging stations in several locations in order to conveniently charge your Tesla. This brings into question, where is the best place to own a Tesla based on charger location? What areas should be avoided when owning a Tesla? And finally what areas have the most and least growth since superchargers have opened?

Our dataset contains information about exactly where Tesla supercharging stations have been opened since their date of creation from 2012 to 2023. It contains street address, city, state, zip, country, number of stalls, kilowatts, GPS, elevation, and the date it was opened. With this information we have created a website with a map and visualizations in order to manipulate and depict the data to gain a better understanding and answer our questions. The purpose of this project is to be able to take data and create a website that anyone can use to further their knowledge on their desired subject. Also to be able to use location and create a map that is user friendly and easily consumable to any viewer.

To start our project we first started with cleaning the data. This required splitting the columns, removing any nulls, and a datachange. The most important task was to take the GPS column, which held both the Latitude and Longitude coordinates, and split it into two columns, one dedicated to Latitude and Longitude respectively. A small number of nulls were dropped, and the team did find some cell entries with question marks in them. After some discussion it was decided to see if the jupyter notebook we were working in, simply could not read certain symbols from other alphabets. We waited to see if the question marks would go away when implemented in our html.

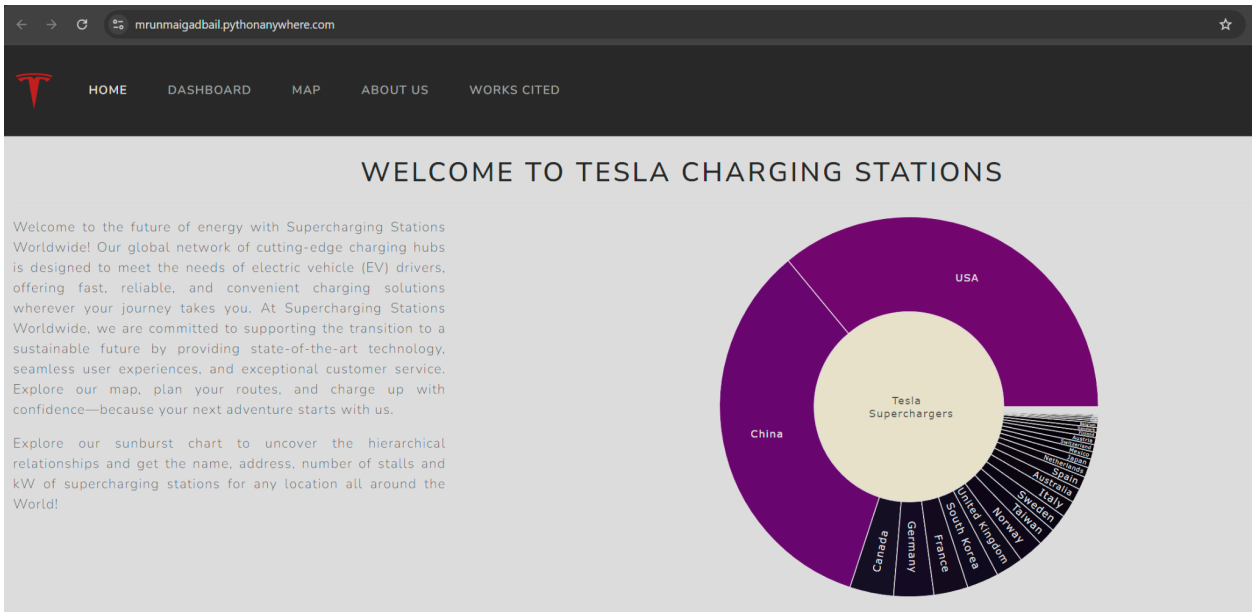
We then came up with possible design concepts for our website and a color theme to help better depict our data story. We started by using https://bootswatch.com/, to choose a theme for the website, ultimately deciding on LUX.





We decided to use reds as it is the main color of Tesla and purples because it is a good contrast. We kept the look of the website itself to be mainly black and white in order to make the visualizations pop and grab attention.

The website contains attributes such as a home page, dashboard, map, about us, and the works cited page. Each tab contains useful information regarding our dataset on Tesla superchargers, project, and our group. When our link is used what the user will first see is our home page.

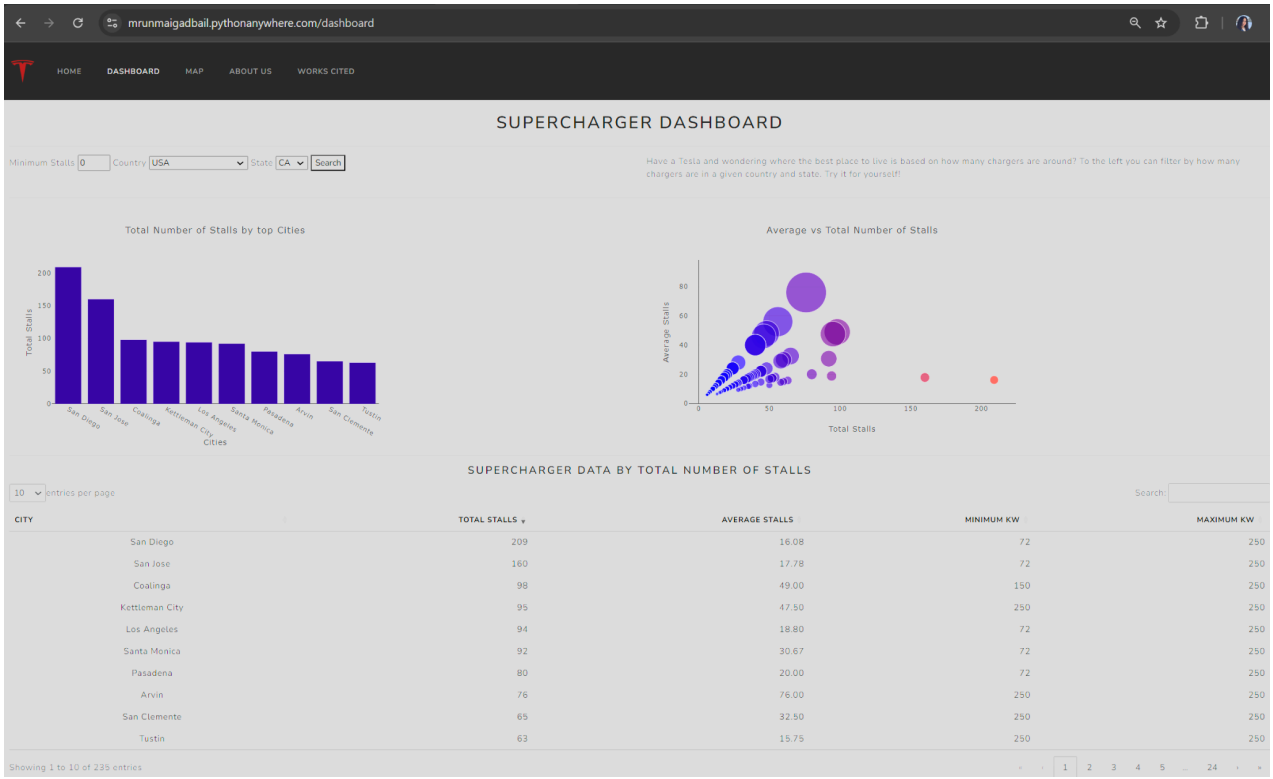


The home page gives users a brief description of what to expect from our website. It also contains an interactive sunburst chart that depicts all of the countries, cities, street address, number of stalls, and the kilowatts.

```
14
15      # Quary to get sunburst chart data
16          def get_sunburst(self):
17              query = f"""
18              SELECT
19              Country,
20              State,
21              City,
22              Supercharger,
23              Street_Address,
24              kW,
25              Stalls,
26              Latitude,
27              Longitude
28              FROM supercharge_locations;
29
30              """
31              df = pd.read_sql(text(query), con = self.engine)
32      # sunburst chart was showing an warning as same city in different states and same state name in different countries
33      # joined city with state and state with country to make unique values
34              df['City'] = df['City'] + "_" + df['State']
35              df['State'] = df['State'] + "_"+ df['Country']
36              data = df.to_dict(orient="records")
37              return(data)
```

The challenge with generating the sunburst came from layering the query properly so the desired information was returned the further a user goes into the sunburst chart.

The next tab after the home page is the dashboard. This contains a brief description of the dashboard, two visualizations and a chart. All can be interacted with and manipulated by filters to show users information based on location.



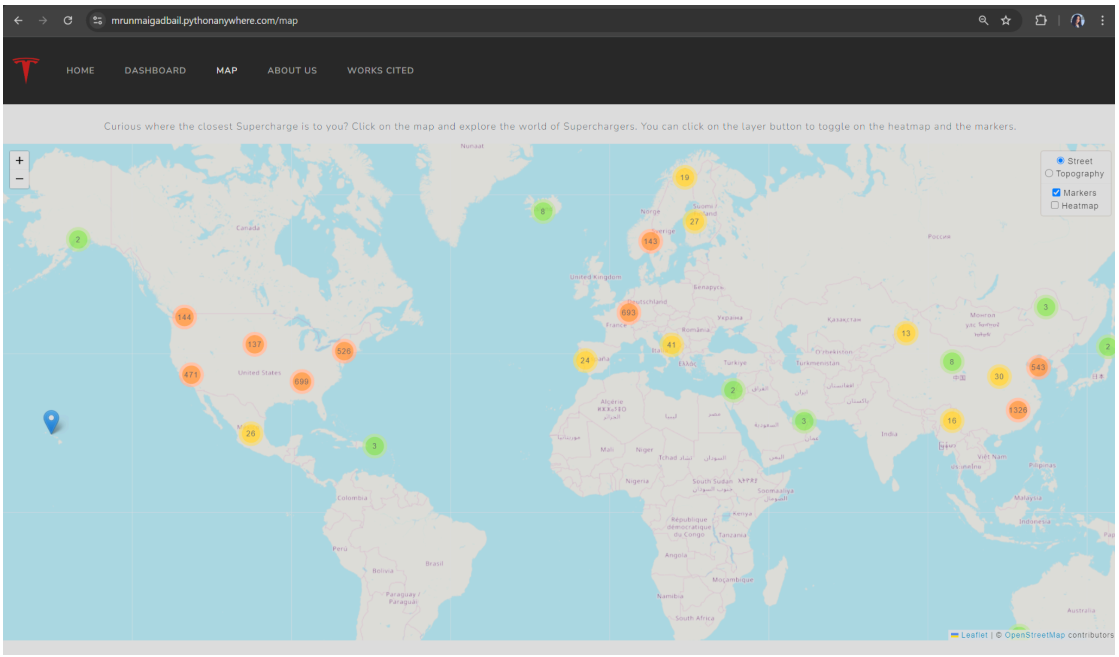

```
function do_work() {
    // extract user input
    let min_stalls = d3.select("#stall_filter").property("value");
    min_stalls = parseInt(min_stalls);
    let country = d3.select("#country_filter").property("value");
    let state = d3.select("#state_filter").property("value");

    // We need to make a request to the API
    let url = `/api/v1/get_dashboard/${min_stalls}/${country}/${state}`;
    d3.json(url).then(function (data) {
        // create the graphs
      make_bubble(data.dashboard_data, country, state);
      make_table(data.dashboard_data, country, state);
      make_bar_chart(data.dashboard_data, country, state);
      });
  }
```

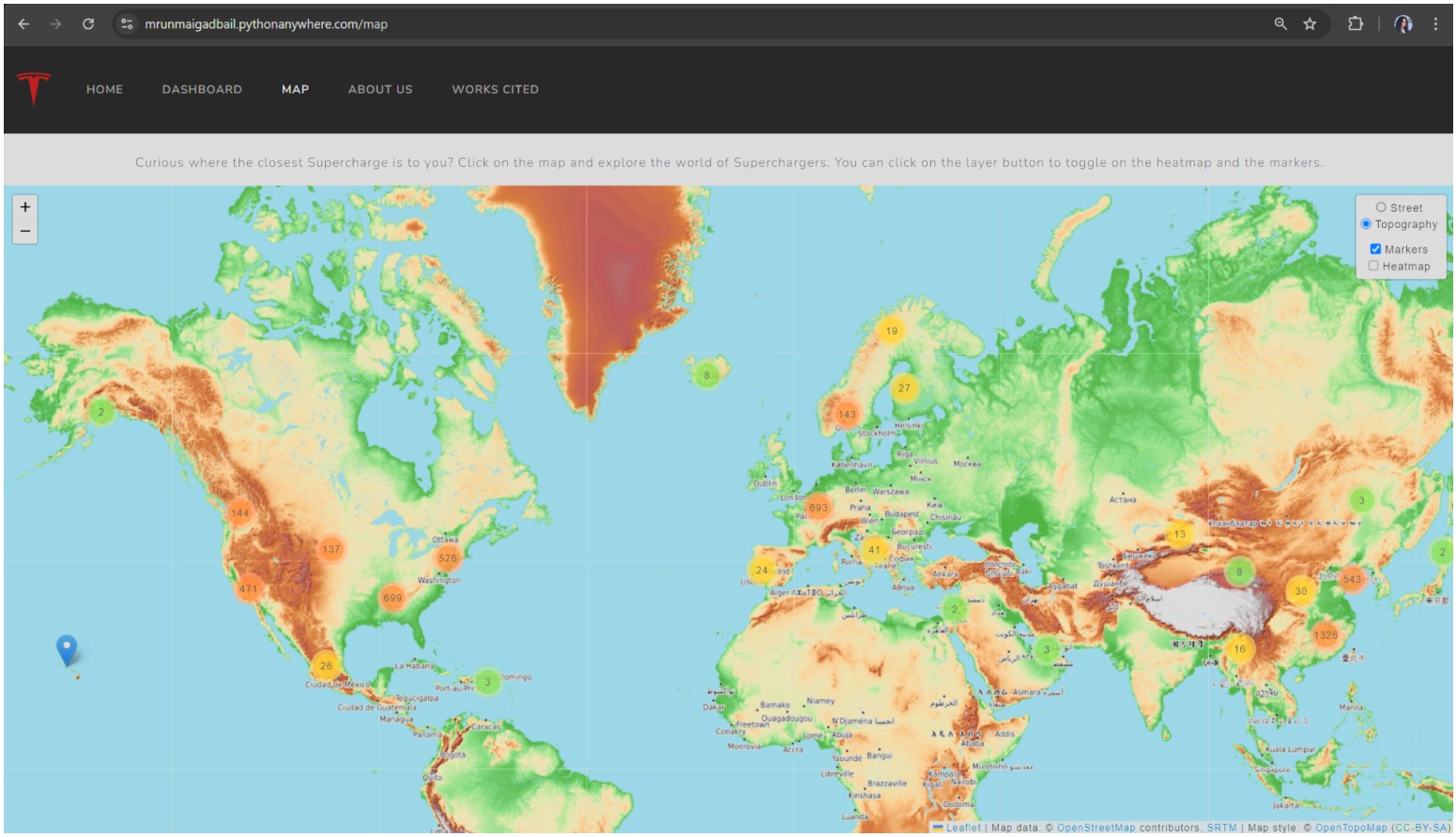
The dashboard was first populated with our three visualizations: the bar chart, the bubble chart, and our information table. After we made sure our queries returned our visualizations, we added filters to the dashboard so users could interact with the

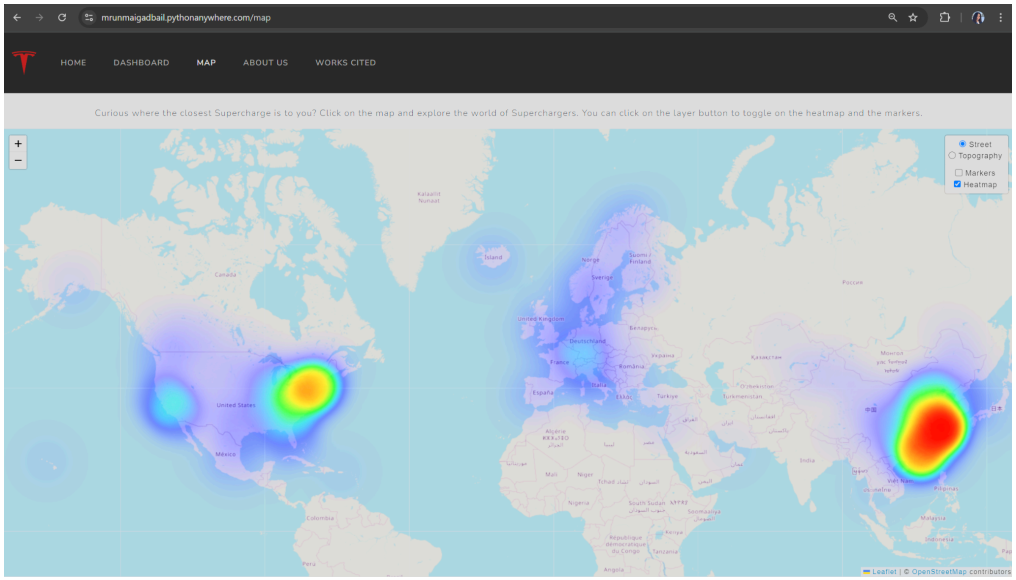
website. Users can filter by country and state, although state acts as a region filter outside of the United States.

After the Dashboard tab is the Map. The Map tab shows users a global map with markers that show the location of every super charger. The map itself starts with a broad overview of all global locations in an area. As you zoom in further, the marker clusters separate into smaller clusters, until eventually breaking into single markers. When you click on a marker, it will list the supercharger location name, address, the number of stalls, and the kWs of the stalls (indicating how fast it would charge).

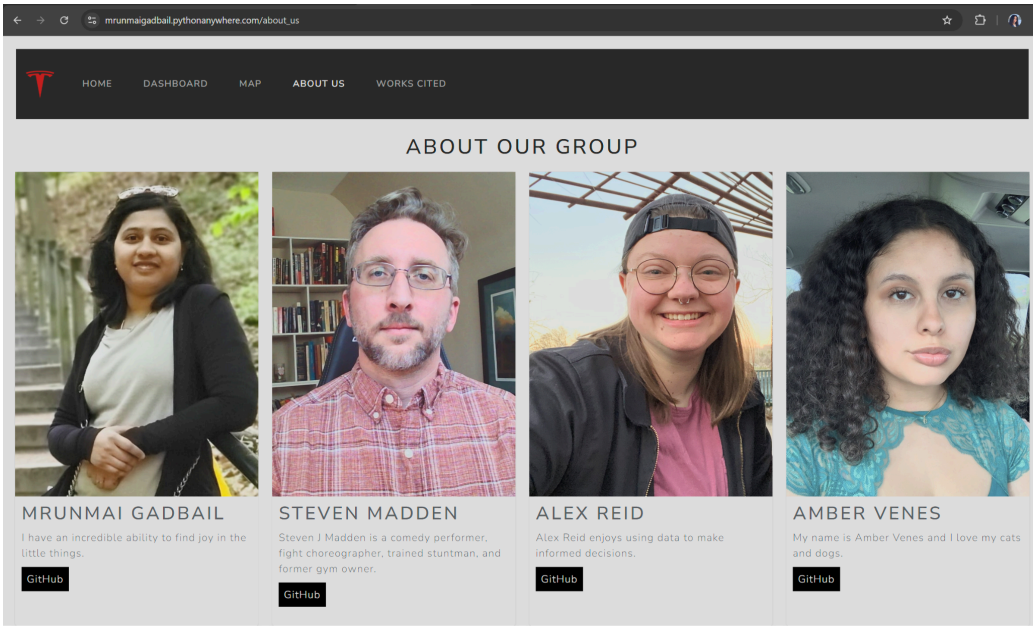


Users also have the option of a heatmap or to show the topography if desired.





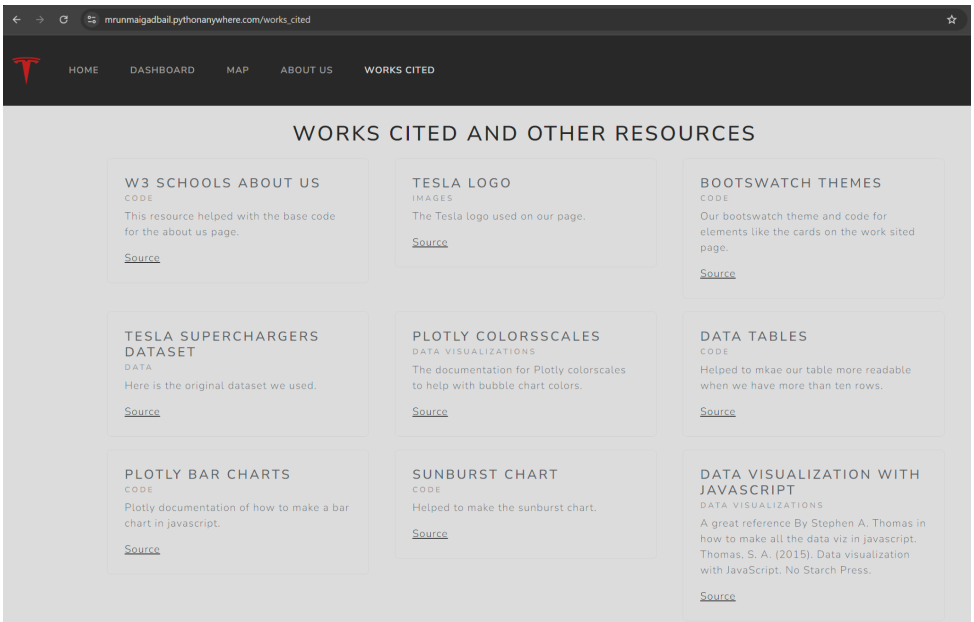
The next tab available is the “About Us” tab. In this tab users get to know our group a little better with pictures depicting each of us, a brief blurb, and a link to our github.



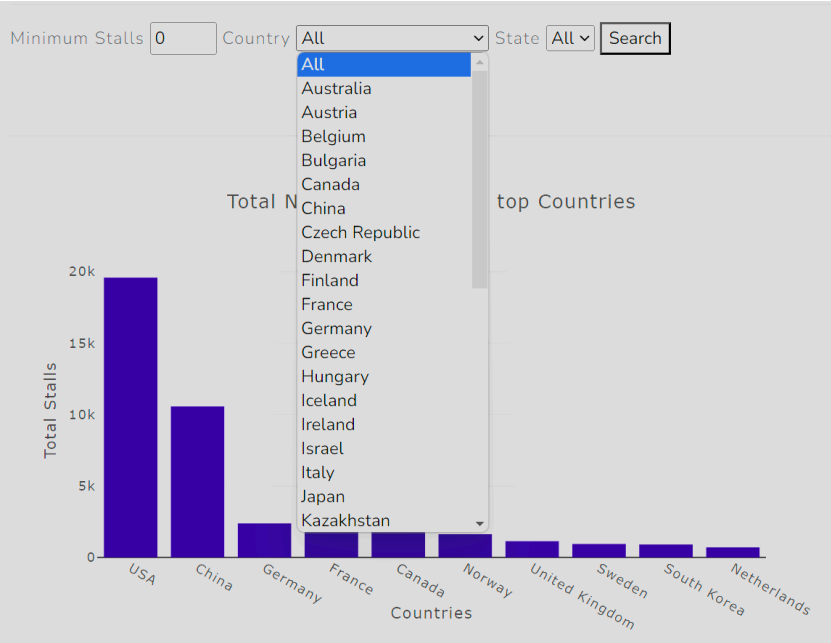

```
<div class="column">
  <div class="card">
    <img src="./static/images/alex.jpg" alt="Alex" style="width:100%; height:500px; object-fit: cover;">
    <div class="container">
      <h2>Alex Reid</h2>
      <!-- <p class="title">Art Director</p> -->
      <p>Alex Reid enjoys using data to make informed decisions.</p>
      <!-- <p>example@example.com</p> -->
      <p><a href= "https://github.com/agreid057" class="button">GitHub</a></p>
      <!-- <p><a href= "https://www.linkedin.com/in/alexandra-g-reid/" class="button2">Linkedin</a></p> -->
    </div>
  </div>
</div>
```

The about us was built in a very straightforward manner through the html.

Finally, our last tab is the “Works Cited” tab that contains all the sources used to create the website.



The first visualization in our dashboard is a bar chart that shows the total number of stalls in descending order. Users can pick a country and the state to show the top 10 cities based on the number of stalls in descending order, or they can also compare the number of stalls simply based on country by using a drop down tab listing all available options.

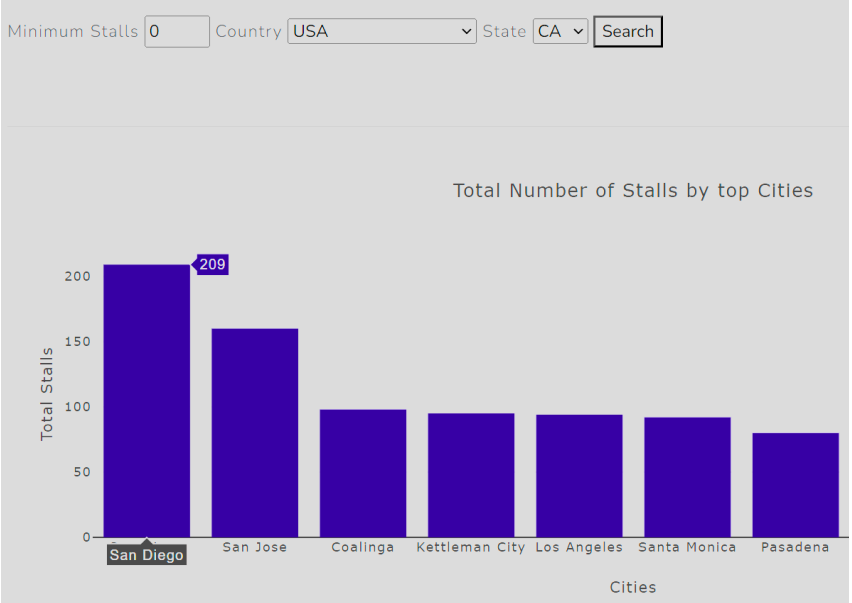


You can see here how we coded our interactive filters.

```
57      #Dashborad data routes
58      @app.route("/api/v1/get_dashboard/<min_stalls>/<country>/<state>")
59      def get_dashboard(min_stalls, country, state):
60          min_stalls = int(min_stalls) # cast to int
61          dashboard_data = sql.dashboard_data(min_stalls, country, state)
62          data = {
63              "dashboard_data": dashboard_data,
64              }
65          return(jsonify(data))
66
67
68      #Country filter data route
69      @app.route("/api/v1/get_Country_filter_Data")
70      def get_Country_filter_Data():
71          Countries = sql.get_Country_filter_Data()
72          data = {
73              "Countries": Countries
74              }
75          return jsonify(data)
76
77      #state filter data route
78      @app.route("/api/v1/get_State_filter_Data/<country>")
79      def get_State_filter_Data(country):
80          States = sql.get_State_filter_Data(country)
81          data = {
82              "States": States
83              }
84          return jsonify(data)
```

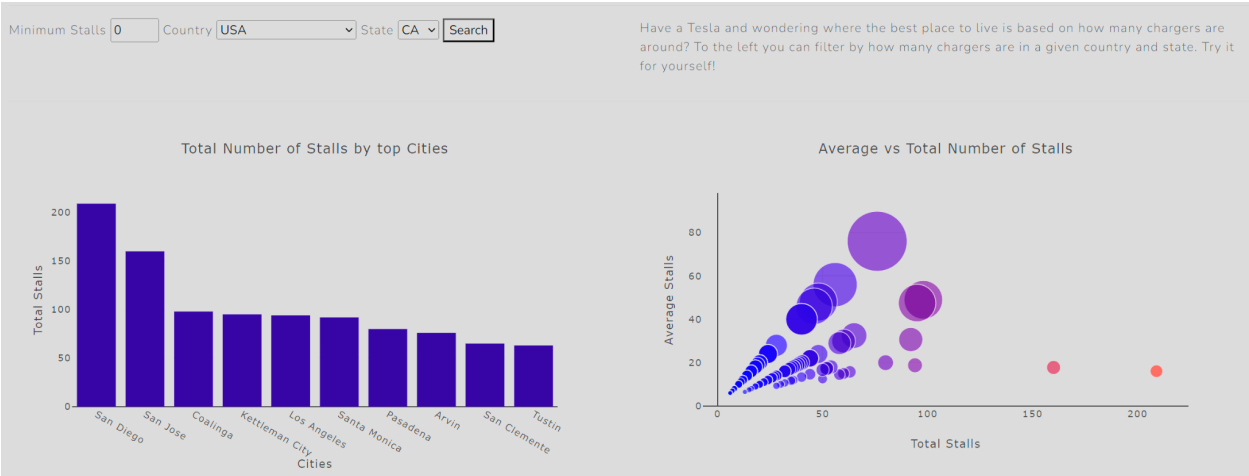
We felt a bar chart would best depict the total number of stalls and to help us answer our question, where is the best location to own a Tesla? To answer this, we first set our filter “Country” to show “All” in order to show the top 10 countries across the world based on the number of stalls. By doing this it is clear the USA has the most number of stalls. Then we used the “Country” filter and set it to “USA” in order to see the top 10 states based on the number of stalls.

After this, we can see the state California contains the most number of stalls. The final step, we changed the “State” filter to “CA” in order to show the top cities based on the number of stalls.



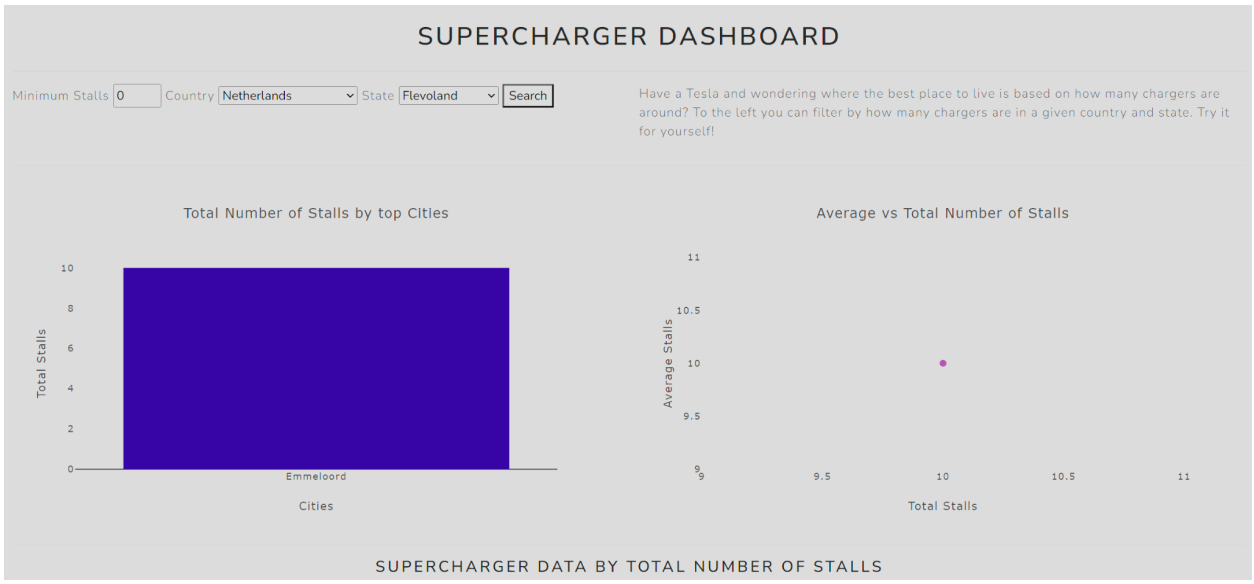
By looking at this graph, one can conclude that San Diego would be the best place to own a Tesla in California.

Our second visualization is a bubble chart that shows users the growth of stalls in a location by average vs the total number of stalls.



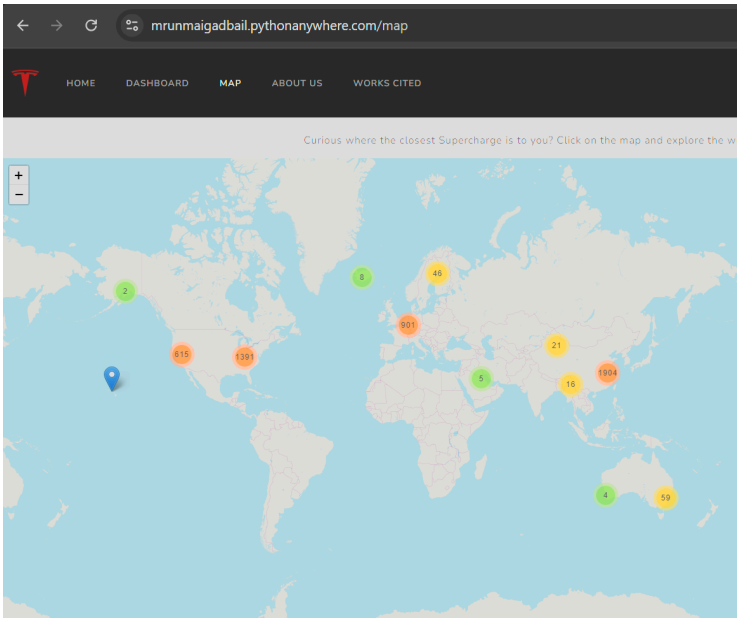
We used this chart to answer the question of what areas have the most or least amount of growth of charger location? Because California is the top place by number of stalls it is safe to assume the growth rate would be good as well. By setting the location to USA, California, Users can see a well developed bubble chart depicting california. By

using the same logic, users can list the top 10 countries, then refer to the bar chart, and then pick the country in 10th place. Users will then discover that the 10th place is the Netherlands. This step may be repeated with the “state” filter selecting the 10th place, in this case being Flevoland.



Users can see by analyzing this graph Flevoland can be assumed to be one of the locations with the least amount of growth.

Our final question, what areas should you avoid when owning a Tesla? This question is most easily answered by taking a look at our map with the markers. Two entire continents, South America and Africa, lack superchargers, as well as the country of India would be terrible places to own a Tesla. However, given a country like India, with such a large population, it is possible to guess a company like Tesla would be interested in the region for potential growth.



Some limitations the data left us with, these vehicles have a rather high price point, making building superchargers in lower populated regions less likely. Superchargers also seem to run slower in areas of higher altitude and colder climates, however, Tesla still seems to build in these climates regardless, making those two fields of interest hard to predict future development.

Unfortunately, the team ran into a problem trying to predict future supercharger sites. Whatever predictions we might have made about future supercharger locations, were rendered pointless by the fact that Tesla laid off over 500 employees, including their entire supercharger division. With no one now incharge of, or working on supercharger sites, the future growth of locations has halted entirely.

A few problems we did run into with the data were name of states and cities are the same- name in city is same as city in other country example: Paris, Texas. We solved this problem with the following code:

```
df = pd.read_sql(text(query), con = self.engine)
df['City'] = df['City'] + "_" + df['State']
df['State'] = df['State'] + "_"+ df['Country']
```

This caused a problem with the sunburst chart chart as well. We had issues with the same city name in different states and same state name in different countries. So before creating the json file we joined ct_state and state_country to add unique values to the json file.

```
const countryLabel = `${entry.Country} `;
const stateLabel = `${entry.State.substring(0, entry.State.indexOf("_"))}`;
const cityLabel = `${entry.City.substring(0, entry.City.indexOf("_"))}`;
```

So the new label before creating the sunburst chart was fixed by removing everything after the underscore

After analyzing all of the interactive visualizations, we were able to answer our questions. California is the best place to live when owning a Tesla. California has the most growth and Flevoland has one of the least amount of growth. Africa and South America should be avoided when owning a Tesla because there are no superchargers located anywhere on the continents. In the end of the project we have learned how to take data from a dataset and put it into a website that anyone can use to easily absorb

information from visualizations that are customizable by filters in order to gather specifically wanted information.