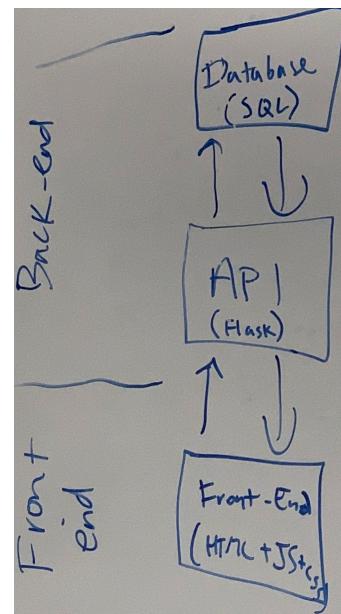


From: Piruz Alemi & Zhongping Yang
To: Team #2
Subject: The Architecture of a Full Stack Application- Project #2
Date: March 9th, 2020

Greetings to All. The architecture of Full Stack Application grew out of our discussion with Json_TA, Illya, Ju Chen Issac Mollinedo & Lori Harris and their guidance, resolving some intractable errors for us. We owe them a debt of gratitude. Thank you for making our project a success in record time of 7 days in the middle of a pandemic crisis and shutdowns.

1. Briefly the full architecture consisted of a front end (Leaflet) + a Backend {SQL/MONGO DB} + Flask, as was first noted by Illya:



2. What we needed was to use Flask as an intermediary step between SQL/MongoDB of our presidential elections, to safeguard our data or use a micro framework like Flask to cut the overhead, instead of accessing the full database directly.

Following code from presidents.js in line 247 connects to Flask url.

In the first place, to run presidents.js we had to activate

python -m http.server

```
244 // http://127.0.0.1:5000/api/v1/presidents
245 // console.log(http://127.0.0.1:5000/api/v1/presidents)
246 urlFlask = "http://127.0.0.1:5000/api/v1/presidents"
247
248 //d3.json("data_candidates_file.json", function(dataRows) {
249   d3.json(urlFlask, function(dataRows) [
250     //var stationStatus2 = statusRes2[0].state;
251     console.log("dataRows Length #####", dataRows.length)
252
253
254
255
256   for (i = 0; i < dataRows.length; i++) {
257
258     president = dataRows[i]
259     state = president.state
260     candLat = president.Latitude
261     candLong = president.Longitude
262     if (candLat == null || candLong == null)
263   
```

The above command would open <http://0.0.0.0:8000/index.html>

The 8000 port, whereby we could activate the index.html that would run president.js or one of its javascripts bkups.

FLASK:

However to run index.html that runs presidents.js we still needed to have made the flask active beforehand. To activate the Flask component, we did the following:

- a. Opened a new terminal window
- b. CD to where the Flask program (api.py) resided

```
Alemi_Project2_Advance          Movies
Applications                      Music
CU-NYC-DATA-PT-10-2019-U-C      Option Pricing
Columbia                          OptionPricing
Creative Cloud Files              Pictures
D3_Advance_Challenge             Piruz Alemi Report on Leaflet.pdf
D3_Challenge                      Plotly-Challenge
Desktop                           Project-1
Doc                               Public
Documents                         SQL_ALCHEMY_Flask_Challenge
Downloads                         SQL_Challenge
ETL_Challenge                     X
IMG_0291.JPG                      __pycache__
IMG_0292.JPG                      bin
IMG_0293.JPG                      code
InteractiveVisual                 code2
JavaScript                        d3_interactive-Dashboard
Law                               opt
Leaflet                           python_API_Challenge2
Leaflet copy                      static
Leaflet_Alemi_challenge           templates
Leaflet_Challenge-master          web-scraping-challenge
LearnPython
(base) Piruzs-MBP:~ piruzalemi$ cd Alemi_Project2_Advance/
```

- c. Run python api.py

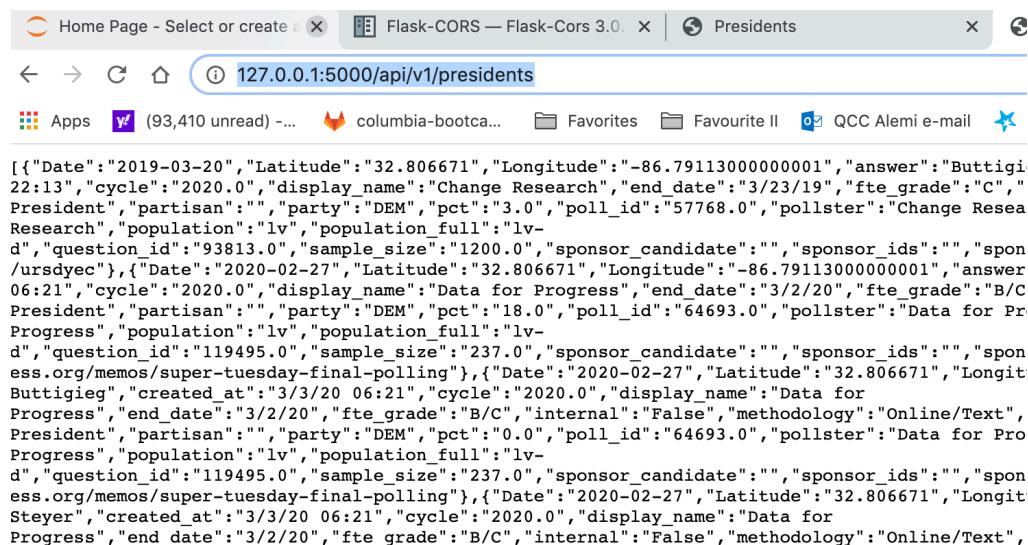
This was the Flask component that would access the MongoDB and could filter any data we needed, as in collection.find{}

The terminal command would verify api.py is syntax error free and create a path between api.py Routes to MongoDB.

Program was short and brief, that after some tweeking on MongoDB collections and Return displays, it worked!

```
api.py -- AlemiSample (Workspace)
JS presidents.js    api.py    ×
Alemi_Project2_Advance > api.py > ...
1 import pymongo
2 from flask import Flask, jsonify
3 from flask_cors import CORS
4
5 #establish mongo connection
6 client = pymongo.MongoClient("mongodb://localhost:27017/")
7 db = client["presidents"]
8 collection=db.presidents_coordinates_poll
9
10 app = Flask(__name__)
11 CORS(app)
12
13
14 @app.route("/api/v1/<presidents>")
15 def get_api_lang(presidents):
16     #data = list(db[presidents].find({}, {'_id': False}))
17     data = list(collection.find({}, {'_id': False}))
18     #return jsonify(data)
19     return jsonify(data)
20     #return "12345"
21
22 if __name__ == "__main__":
23     app.run()
```

The Flask output was then tested, with and without filterings and displayed as in:
<http://127.0.0.1:5000/api/v1/presidents>



3. These breakdowns also helped in division of labor, minimizing down-time and “localizing” the errors and working in teams and identifying source of errors.
-

On Some Proto-Types used and Data Cleaning:

- A. If you go to Git-Lab/ClassWork 17/logic.js we have a good protocol of a program that reads from a url (line #2) (see below) which creates a jsonified file.
 - B. Once the file is jsonified, via d3.json(url), we can access the key-value pairs and extract the needed data as input to Leaflet/front-end program, which in our case was called earthquake.js or the presidential candidates as in presidents.js
 - C. Presidents-Coordinates_polling data that was created by AlemiMerge2.js program
 - D. We filtered this data, in 3 ways:
 - a. We set some filters in the api.py under Flask as in say a selection of a candidate(S)
- ```
E. def get_api_lang(presidents):
F. #data = list(db[presidents].find({}, {'_id': False}))
G. #data = list(collection.find({}, {'_id': False}))
H. data = list(collection.find({"answer": "Sanders"}, {'_id': False}))
I. #return jsonify(data)
J. return jsonify(data)
```

From MongoDB, the collections was accessed, and in its document we filtered on say “Sanders”

- a) Filtering in api.py under Flask
  - b) Filtering in MongoDB
  - c) Filtering in Preident.js as in suppression of Null values of long and lat in coordinates.
- K. Recall, The Alemimerge2.js program that read a CSV file of presidential polling data and left joined it with State-GPS Coordinates files, based on State Codes. In addition it created an SQL data base.

- L. We also imported the CSV file into MONGODB. In this case we created presidents DB and added presidents\_coordinates\_poll.csv file to its collection with 18,296 documents, as shown:

| Collection Name             | Documents | Avg. Document Size | Total Document Size | Num. Indexes | Total Index Size | Properties |
|-----------------------------|-----------|--------------------|---------------------|--------------|------------------|------------|
| presidents_coordinates_poll | 7,649     | 960.4 B            | 7.3 MB              | 1            | 81.9 KB          |            |
| presidents_poll             | 18,296    | 892.1 B            | 16.3 MB             | 1            | 180.2 KB         |            |
| presidents_poll3            | 0         | -                  | 0.0 B               | 1            | 4.1 KB           |            |

The data now included the coordinates of the State, where we could tell Leaflet to add the marker vectors on its map layers.

The screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with navigation options: HOST (localhost:27017), CLUSTER (Standalone), EDITION (MongoDB 4.2.2 Community), and a search bar labeled 'Filter your data' containing a list of databases: Nasa, SPY, admin, animals, classDB, config, craigslist\_app, craigslist\_db, local, myNewDB, nhl\_db, presidents, presidents\_coordinates..., and ...

The main right panel is titled 'presidents.presidents\_coordinates\_poll'. It has tabs for Documents, Aggregations, Schema, and Explain Plan. The 'Documents' tab is selected. Below it is a 'FILTER' button and an 'ADD DATA' button with a plus icon. There are also 'VIEW' and other view mode buttons.

The document details pane shows a single document with the following fields:

```

_id: ObjectId("5e697d553ea616286f3821f2")
state: "Alabama"
Latitude: "32.806671"
Longitude: "-86.79113000000001"
question_id: "93813.0"
poll_id: "57768.0"
cycle: "2020.0"
pollster_id: "1365.0"
pollster: "Change Research"
sponsor_ids: ""
sponsors: ""
display_name: "Change Research"
pollster_rating_id: "48.0"
pollster_rating_name: "Change Research"
fte_grade: "C"
sample_size: "1200.0"
population: "1v"
population_full: "1v-d"
methodology: "Online"
office_type: "U.S. President"
start_date: "3/20/19"
end_date: "3/23/19"
sponsor_candidate: ""
internal: "False"
partisan: ""

```

At the bottom of the document details pane, there's a button labeled 'SHOW 12 MORE FIELDS' with a downward arrow.

---

In our case line 2 becomes the Flask URL that was accessing/interfacing with DB: SQL/MONGODB. Some of the data in fields of interest lik Longitude & Latitude had nulls that we excluded.

```

logic.js 2.72 KB
Edit Web IDE Replace Delete

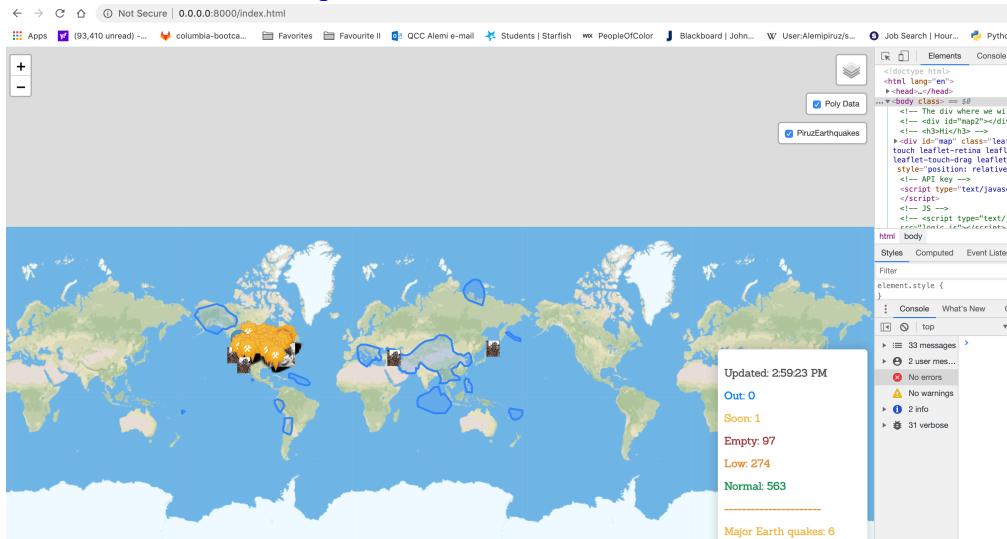
1 // Store our API endpoint inside queryUrl
2 var queryUrl = "https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson&starttime=2014-01-01&endtime=" +
3 "2014-01-02&maxlongitude=-69.52148437&minlongitude=-123.83789062&maxlatitude=40.74894534&minlatitude=25.16517337";
4
5 // Perform a GET request to the query URL
6 d3.json(queryUrl, function(data) {
7 // Once we get a response, send the data.features object to the createFeatures function
8 createFeatures(data.features);
9 });
10
11 function createFeatures(earthquakeData) {
12
13 // Define a function we want to run once for each feature in the features array
14 // Give each feature a popup describing the place and time of the earthquake
15 function onEachFeature(feature, layer) {
16 layer.bindPopup("<h3>" + feature.properties.place +
17 "</h3>
<p>" + new Date(feature.properties.time) + "</p>");
18 }
19
20 // Create a GeoJSON layer containing the features array on the earthquakeData object
21 // Run the onEachFeature function once for each piece of data in the array
22 var earthquakes = L.geoJSON(earthquakeData, {
23 onEachFeature: onEachFeature
24 });
25
26 // Sending our earthquakes layer to the createMap function
27 createMap(earthquakes);
28 }
29
30 function createMap(earthquakes) {
31
32 // Define streetmap and darkmap layers
33 var streetmap = L.tileLayer("https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}", {
34 attribution: "Map data © OpenStreetMap contributors, CC-BY-SA",
35 maxZoom: 18,
36 id: "mapbox.streets",
37 accessToken: API_KEY
38 });
39
40 var darkmap = L.tileLayer("https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}", {
41 attribution: "Map data © OpenStreetMap contributors, CC-BY-SA",
42 maxZoom: 18,
43 id: "mapbox.dark"
44 });

```

## Flask\_CORS

Note worthy was the use of CORS that resolved the conflicts between two separate session URLs

We had to install Flask CORS <https://flask-cors.readthedocs.io/en/latest/index.html>



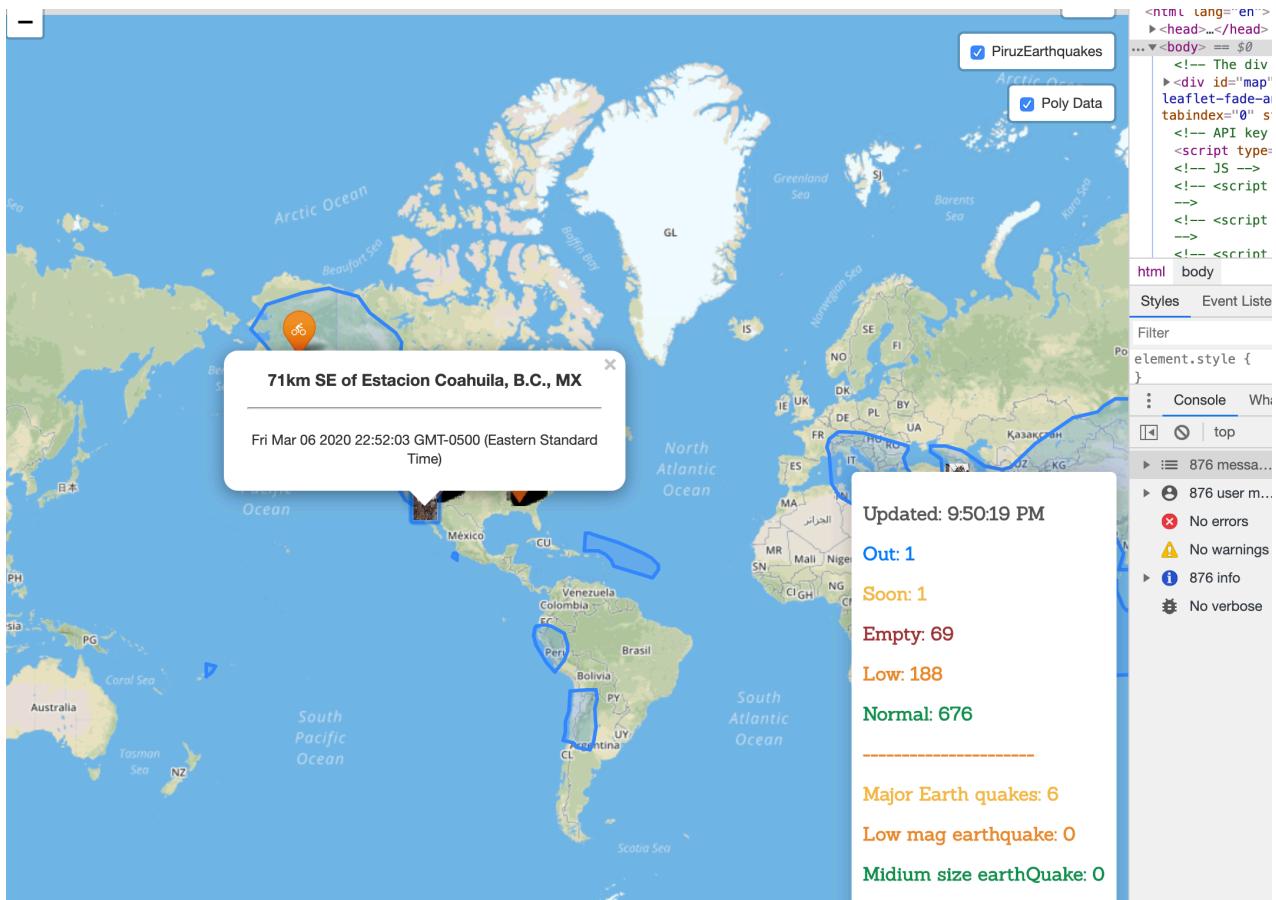
But we could also run without CORS, as in the change of the architecture, where First Flask is run. Running flask via render template required some quick restructuring as in:

```
@app.route("/")
def index():
 return render_template("index.html")
```

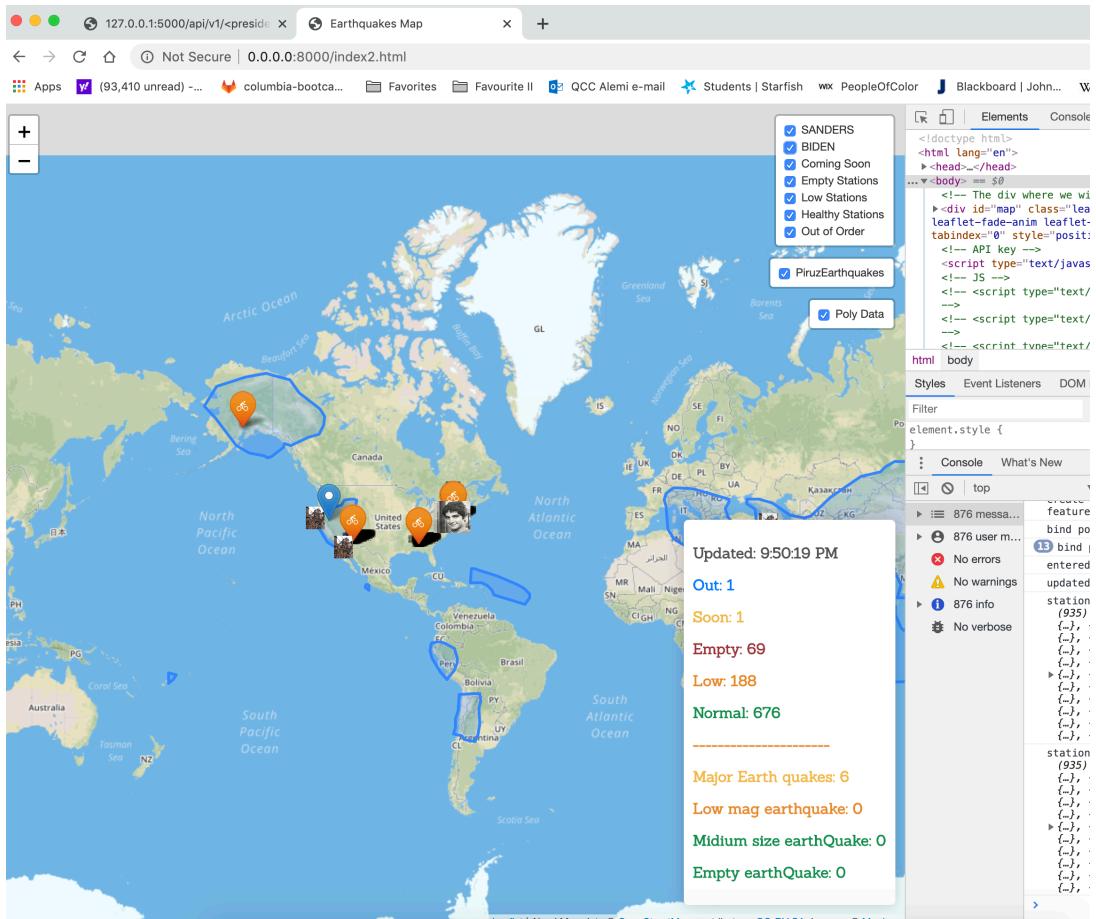
Helpful Links:

- <https://docs.mongodb.com/manual/reference/method/db.collection.find/>
- <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- <https://stackoverflow.com/questions/33962616/setting-full-screen-leaflet-js-map-using-bootstrap>
- <https://getbootstrap.com/docs/4.3/getting-started/introduction/>
- <https://embed.plnkr.co/plunk/VyCze8>
- <https://icons8.com/icon/pack/cultures/doodle>
- [https://i.vimeocdn.com/portrait/3134773\\_600x600.webp](https://i.vimeocdn.com/portrait/3134773_600x600.webp)
- 

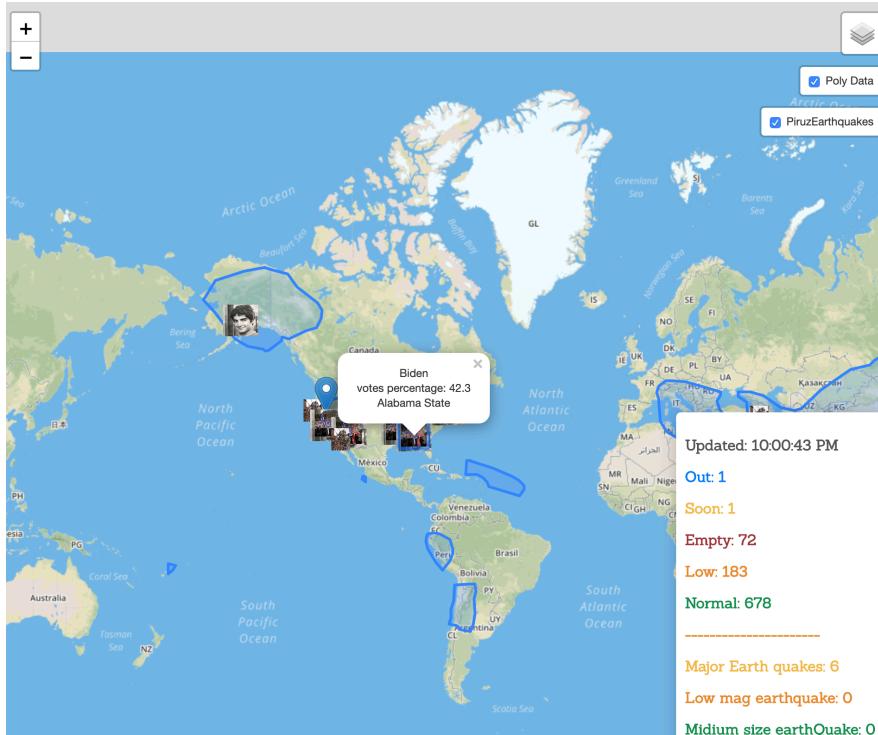
## Results:

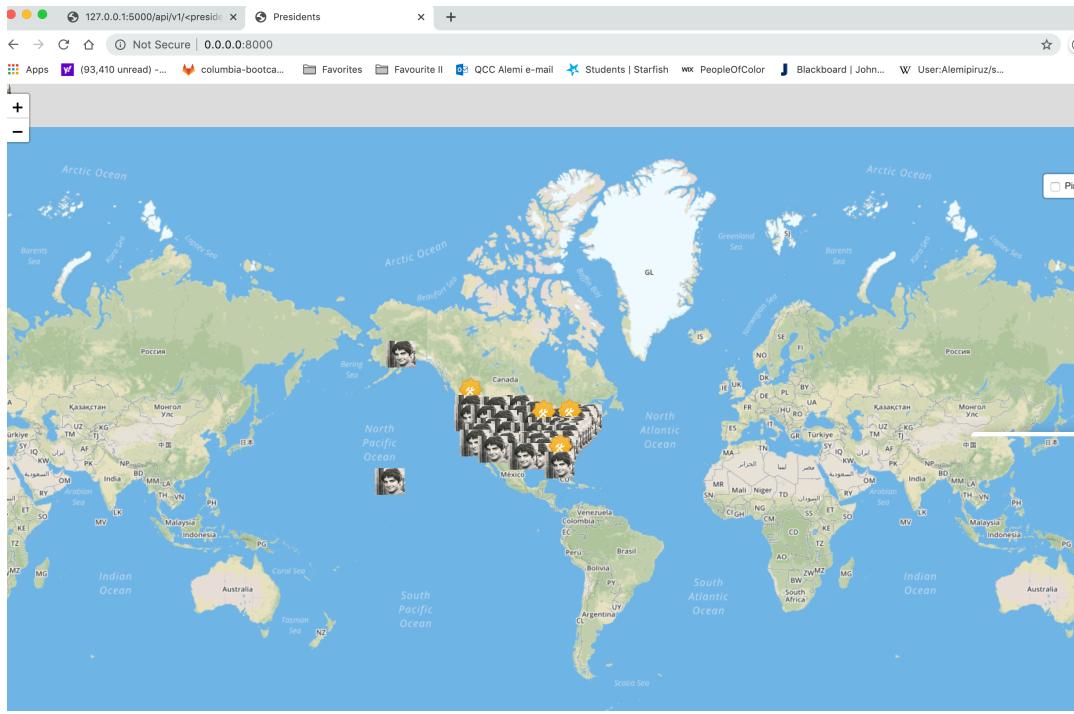


Captures the earthquakes location, date and time + its Poly data

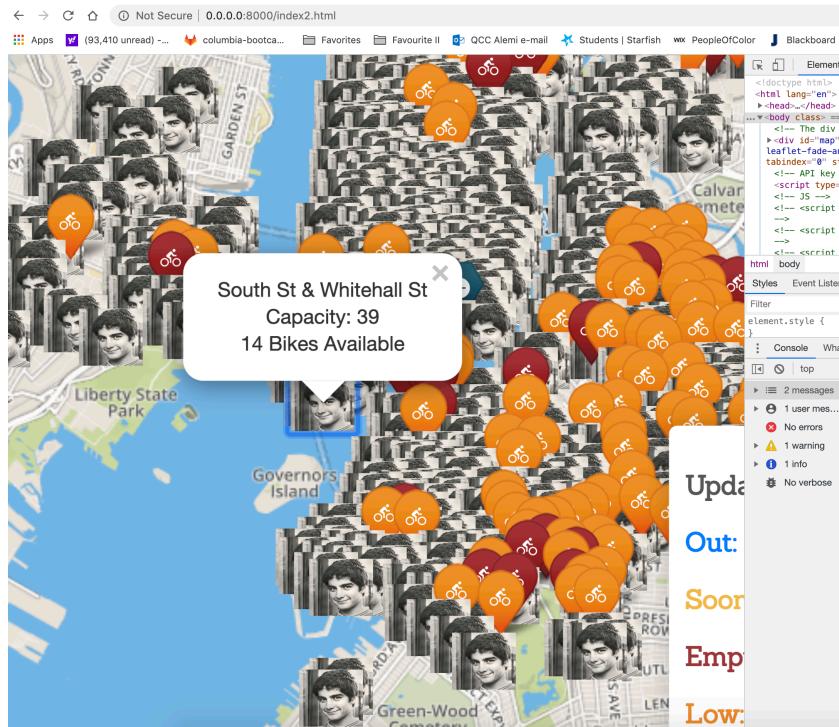
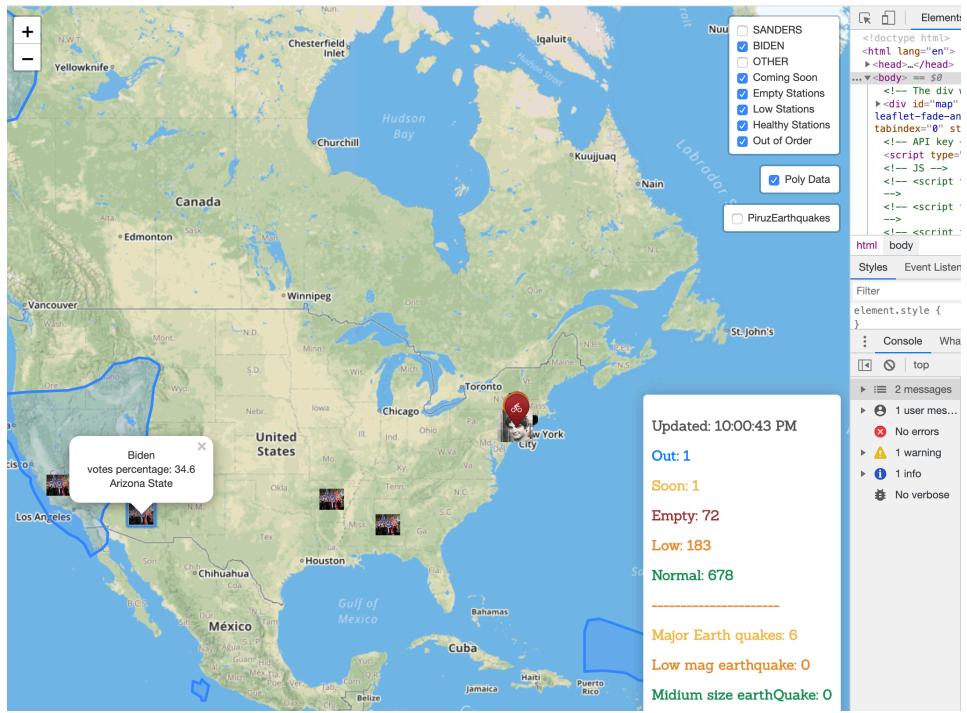


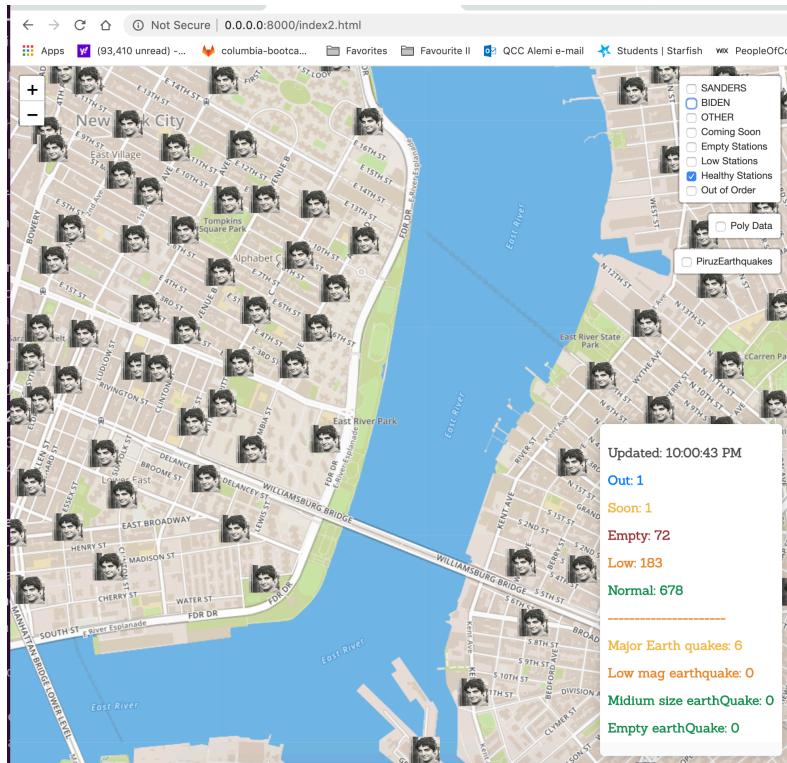
It also captures the Votes distribution:





Selecting on Biden only:





## Flask Output:

```
127.0.0.1:5000/api/v1</presid... x Earthquakes Map x ORC java - Google Search x Flask-CORS — Flask-Cors 3.0 x Flask-CORS — Flask-Cors 1.3 x +
```

App (93,410 unread) ~ 🌐 columbia-bootca... Favorites Favourite II QCC Alemi e-mail Students | Starfish PeopleOfCo

Updated: 10:00:43 PM

**Out: 1**

**Soon: 1**

**Empty: 72**

**Low: 183**

**Normal: 678**

---

**Major Earthquakes: 6**

**Low mag earthquake: 0**

**Medium size earthQuake: 0**

**Empty earthQuake: 0**

[{"date": "2019-03-20", "Latitude": "32.806671", "Longitude": "-86.79113000000001", "answer": "Buttigieg", "candidate\_id": "13345.0", "candidate\_name": "Pete Buttigieg", "created\_at": "3/26/19 22:13", "cycle": "2020.0", "display\_name": "Change Research", "end\_date": "3/23/19", "fie\_grade": "C", "internal": "False", "methodology": "Online", "nationwide\_batch": "False", "notes": "", "office\_type": "U.S. House", "population": "1v", "population\_full": "1v", "polister\_id": "64493.0", "polister\_rating\_id": "522.0", "polister\_rating\_name": "Change Research", "population\_size": "1v", "population\_full\_size": "1v", "question\_id": "93813.0", "sample\_size": "120", "sponsor\_candidate": "", "sponsor\_id": "13289.0", "stage": "Primary", "start\_date": "3/20/19", "state": "Alabama", "tracking": "False", "url": "https://www.dataforprogress.org/memos/super-tuesday-final-polling"}, {"date": "2019-03-20", "Latitude": "32.806671", "Longitude": "-86.79113000000001", "answer": "Bloomberg", "candidate\_id": "13289.0", "candidate\_name": "Michael Bloomberg", "created\_at": "3/3/20 06:21", "cycle": "2020.0", "display\_name": "Data for Progress", "end\_date": "3/2/20", "fie\_grade": "B/C", "internal": "False", "methodology": "Online/Text", "nationwide\_batch": "False", "notes": "", "office\_type": "U.S. House", "population": "1v", "population\_full": "1v", "polister\_id": "1515.0", "polister\_rating\_id": "522.0", "polister\_rating\_name": "Data for Progress", "population\_size": "1v", "population\_full\_size": "1v", "question\_id": "118495.0", "sample\_size": "237.0", "sponsor\_candidate": "", "sponsor\_id": "13289.0", "stage": "Primary", "start\_date": "3/2/20", "state": "Alabama", "tracking": "False", "url": "https://www.dataforprogress.org/memos/super-tuesday-final-polling"}, {"date": "2020-02-27", "Latitude": "32.806671", "Longitude": "-86.79113000000001", "answer": "Buttigieg", "candidate\_id": "13345.0", "candidate\_name": "Pete Buttigieg", "created\_at": "3/3/20 06:21", "cycle": "2020.0", "display\_name": "Data for Progress", "end\_date": "3/2/20", "fie\_grade": "B/C", "internal": "False", "methodology": "Online/Text", "nationwide\_batch": "False", "notes": "", "office\_type": "U.S. House", "population": "1v", "population\_full": "1v", "polister\_id": "1515.0", "polister\_rating\_id": "522.0", "polister\_rating\_name": "Data for Progress", "population\_size": "1v", "population\_full\_size": "1v", "question\_id": "118495.0", "sample\_size": "237.0", "sponsor\_candidate": "", "sponsor\_id": "13289.0", "stage": "Primary", "start\_date": "3/2/20", "state": "Alabama", "tracking": "False", "url": "https://www.dataforprogress.org/memos/super-tuesday-final-polling"}, {"date": "2020-02-27", "Latitude": "32.806671", "Longitude": "-86.79113000000001", "answer": "Steyer", "candidate\_id": "13327.0", "candidate\_name": "Tom Steyer", "created\_at": "3/3/20 06:21", "cycle": "2020.0", "display\_name": "Data for Progress", "end\_date": "3/2/20", "fie\_grade": "B/C", "internal": "False", "methodology": "Online/Text", "nationwide\_batch": "False", "notes": "", "office\_type": "U.S. House", "population": "1v", "population\_full": "1v", "polister\_id": "64493.0", "polister\_rating\_id": "522.0", "polister\_rating\_name": "Data for Progress", "population\_size": "1v", "population\_full\_size": "1v", "question\_id": "118495.0", "sample\_size": "237.0", "sponsor\_candidate": "", "sponsor\_id": "13327.0", "stage": "Primary", "start\_date": "3/2/20", "state": "Alabama", "tracking": "False", "url": "https://www.dataforprogress.org/memos/super-tuesday-final-polling"}, {"date": "2019-03-20", "Latitude": "32.806671", "Longitude": "-86.79113000000001", "answer": "Gabbard", "candidate\_id": "13343.0", "candidate\_name": "Tulsi Gabbard", "created\_at": "3/26/19 22:13", "cycle": "2020.0", "display\_name": "Change Research", "end\_date": "3/23/19", "fie\_grade": "C", "internal": "False", "methodology": "Online", "nationwide\_batch": "False", "notes": "", "office\_type": "U.S. House", "population": "1v", "population\_full": "1v", "polister\_id": "1515.0", "polister\_rating\_id": "522.0", "polister\_rating\_name": "Change Research", "population\_size": "1v", "population\_full\_size": "1v", "question\_id": "118495.0", "sample\_size": "237.0", "sponsor\_candidate": "", "sponsor\_id": "13343.0", "stage": "Primary", "start\_date": "3/20/19", "state": "Alabama", "tracking": "False", "url": "https://www.dataforprogress.org/memos/super-tuesday-final-polling"}, {"date": "2019-03-20", "Latitude": "32.806671", "Longitude": "-86.79113000000001", "answer": "Warren", "candidate\_id": "13258.0", "candidate\_name": "Elizabeth Warren", "created\_at": "3/26/19 22:13", "cycle": "2020.0", "display\_name": "Change Research", "end\_date": "3/23/19", "fie\_grade": "C", "internal": "False", "methodology": "Online", "nationwide\_batch": "False", "notes": "", "office\_type": "U.S.

Thank you! & Have a wonderful day.