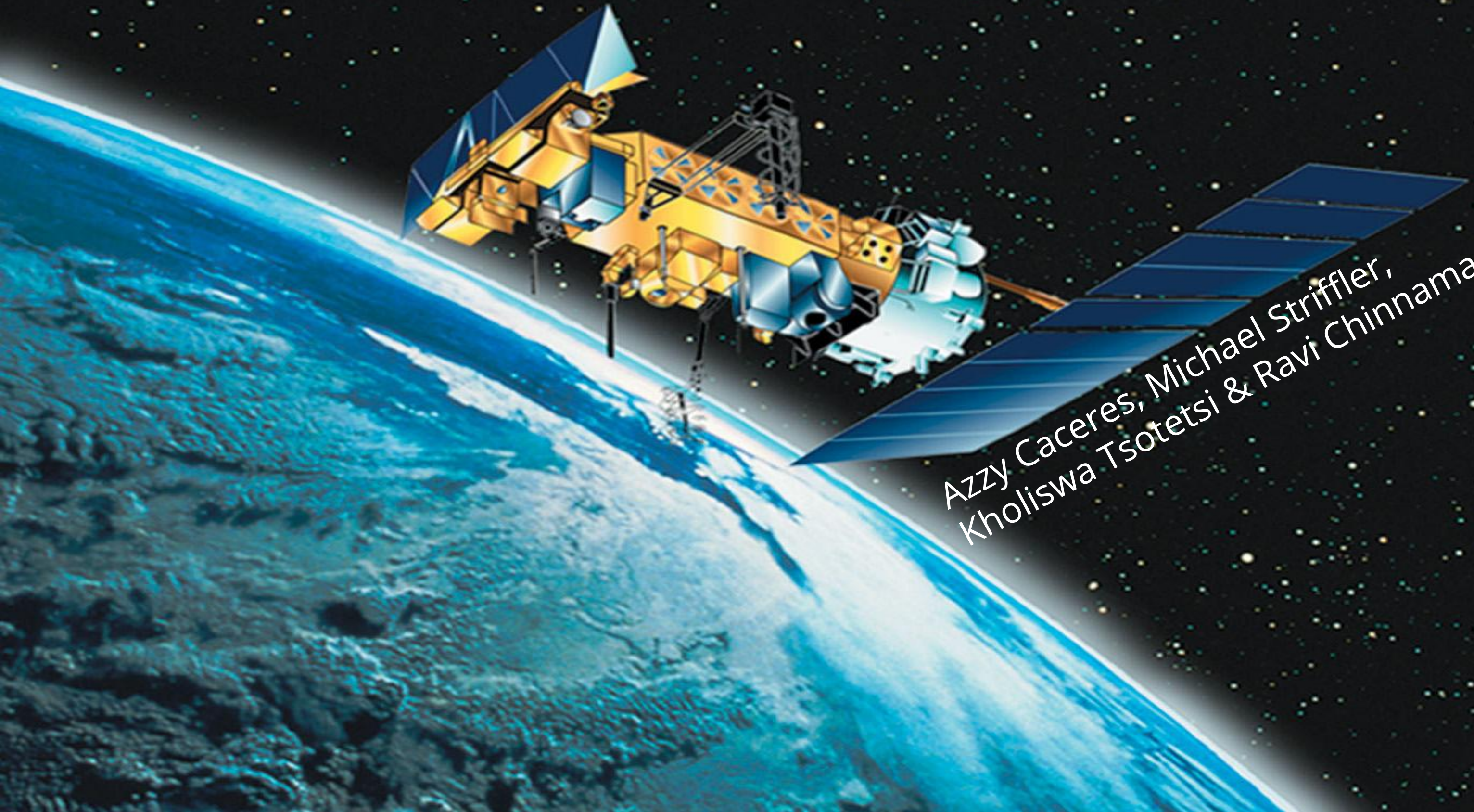
Satellite Exploration

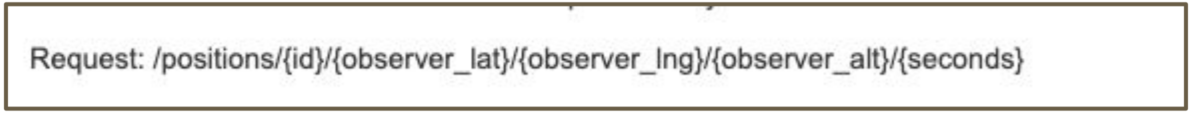


Azzy Caceres, Michael Striffler,
Kholiswa Tsotetsi & Ravi Chinnamanaidu

Data Source: N2YO

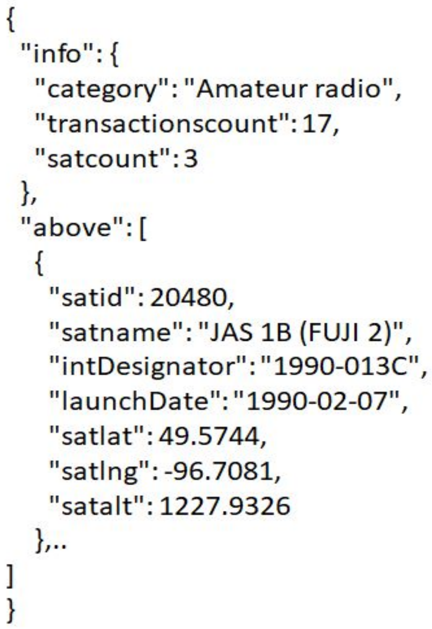
Provides the details of the satellites launched from various countries, the site provides details of the satellites launched, countries that launched, number of satellites for each country, satellites categories, satellite name and other updated satellite launch programs.

API Call:



Request: /positions/{id}/{observer_lat}/{observer_lng}/{observer_alt}/{seconds}

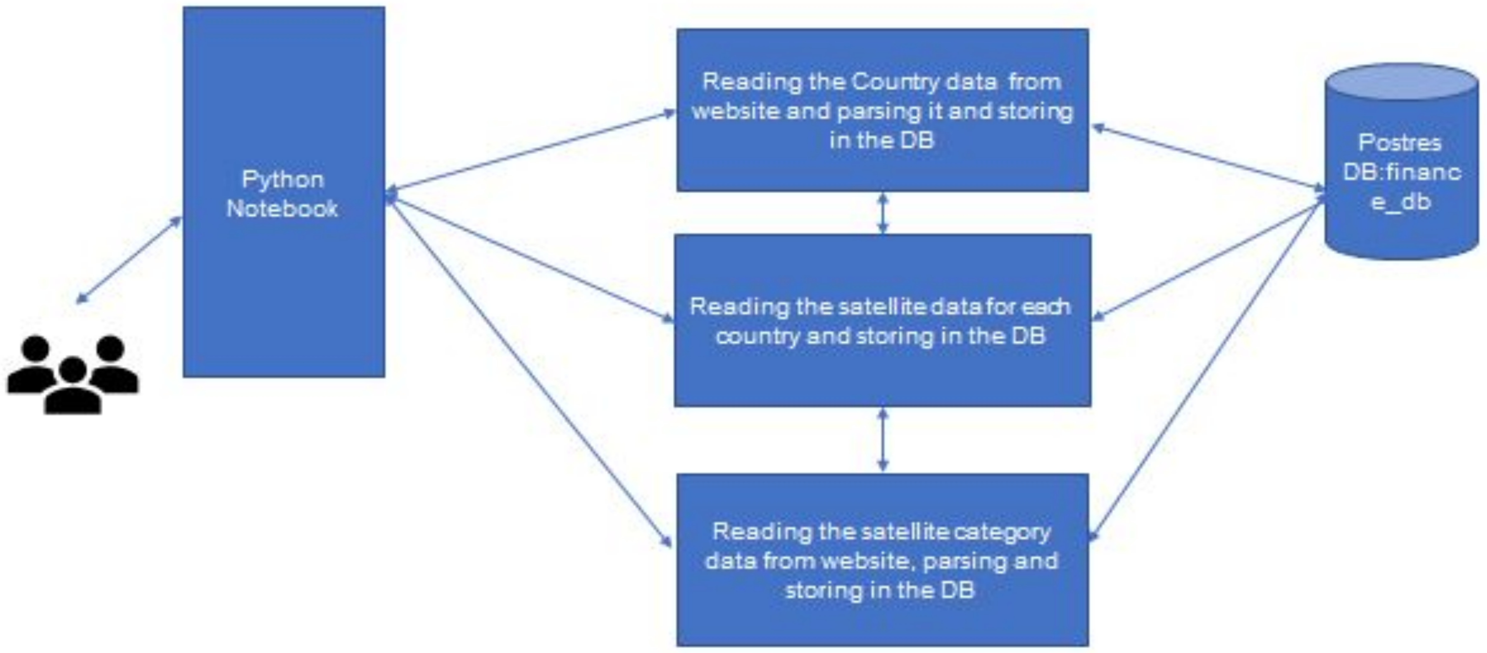
Returned JSON Response:



```
{
 "info": {
  "category": "Amateur radio",
  "transactionscount": 17,
  "satcount": 3
 },
 "above": [
  {
   "satid": 20480,
   "satname": "JAS 1B (FUJI 2)",
   "intDesignator": "1990-013C",
   "launchDate": "1990-02-07",
   "satlat": 49.5744,
   "satlng": -96.7081,
   "satalt": 1227.9326
  },..
]
}
```

Satellite Acquisition Overview

From Web Scraping to Storing in SQL Database



Python Notebook

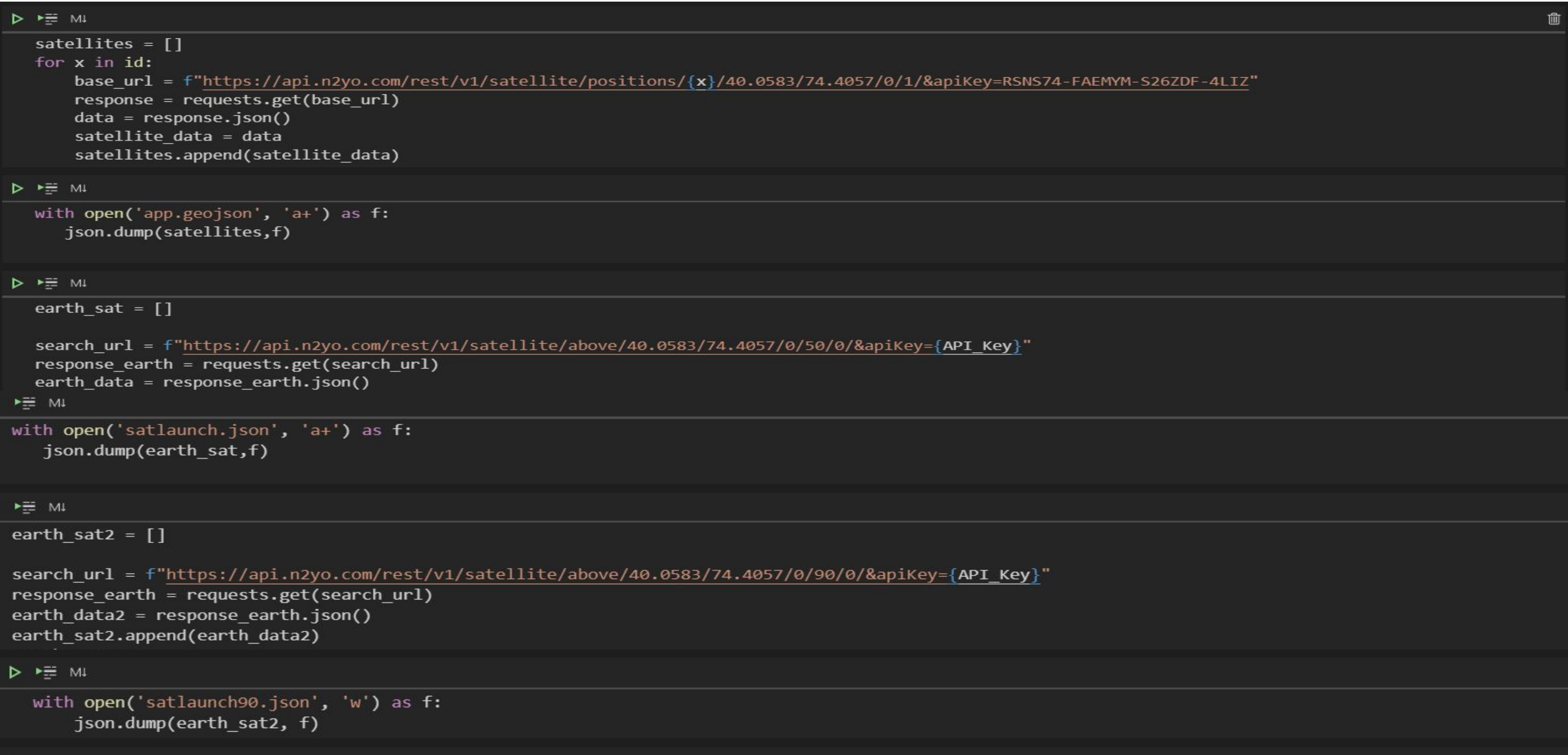
Reading the Country data from website and parsing it and storing in the DB

Reading the satellite data for each country and storing in the DB

Reading the satellite category data from website, parsing and storing in the DB

Postres DB:finance_db

Reading API to Create JSON and GeoJSON Files



```
satellites = []
for x in id:
    base_url = f"https://api.n2yo.com/rest/v1/satellite/positions/{x}/40.0583/74.4057/0/1/&apiKey=RSNS74-FAEMYM-S26ZDF-4LIZ"
    response = requests.get(base_url)
    data = response.json()
    satellite_data = data
    satellites.append(satellite_data)
```

```
with open('app.geojson', 'a+') as f:
   json.dump(satellites,f)
```

```
earth_sat = []

search_url = f"https://api.n2yo.com/rest/v1/satellite/above/40.0583/74.4057/0/50/0/&apiKey={API_Key}"
response_earth = requests.get(search_url)
earth_data = response_earth.json()
```

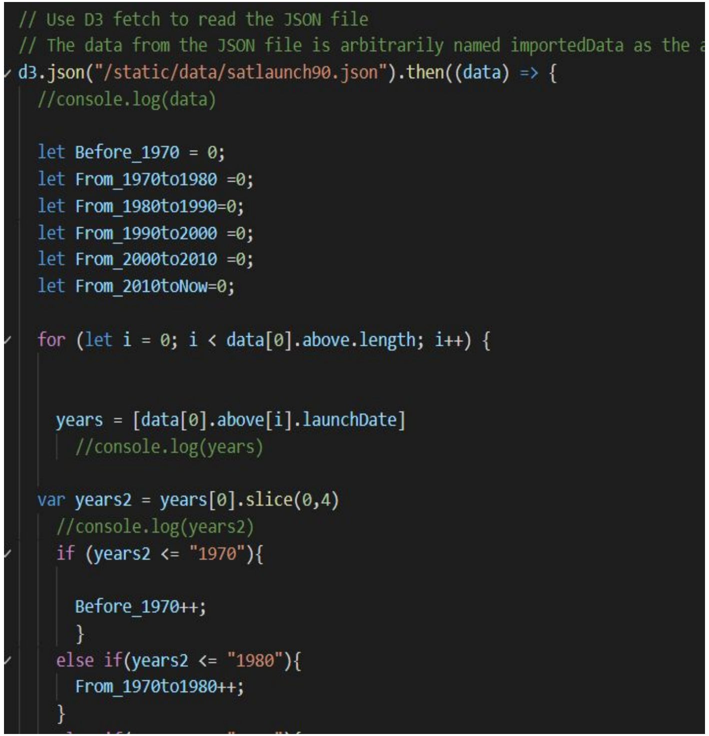
```
with open('satlaunch.json', 'a+') as f:
    json.dump(earth_sat,f)
```

```
earth_sat2 = []

search_url = f"https://api.n2yo.com/rest/v1/satellite/above/40.0583/74.4057/0/90/0/&apiKey={API_Key}"
response_earth = requests.get(search_url)
earth_data2 = response_earth.json()
earth_sat2.append(earth_data2)
```

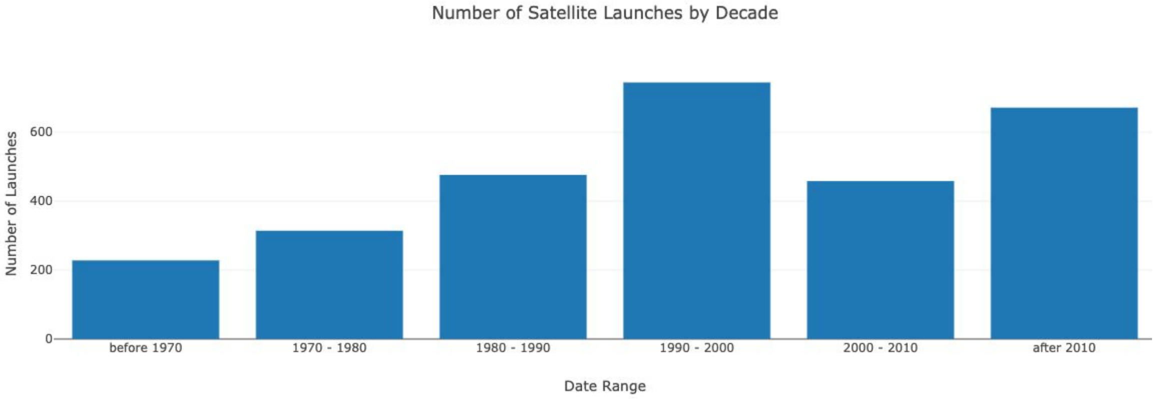
```
with open('satlaunch90.json', 'w') as f:
    json.dump(earth_sat2, f)
```

Visualization: Bar Chart Launch Dates

***Libraries Used:** Plotly, D3*

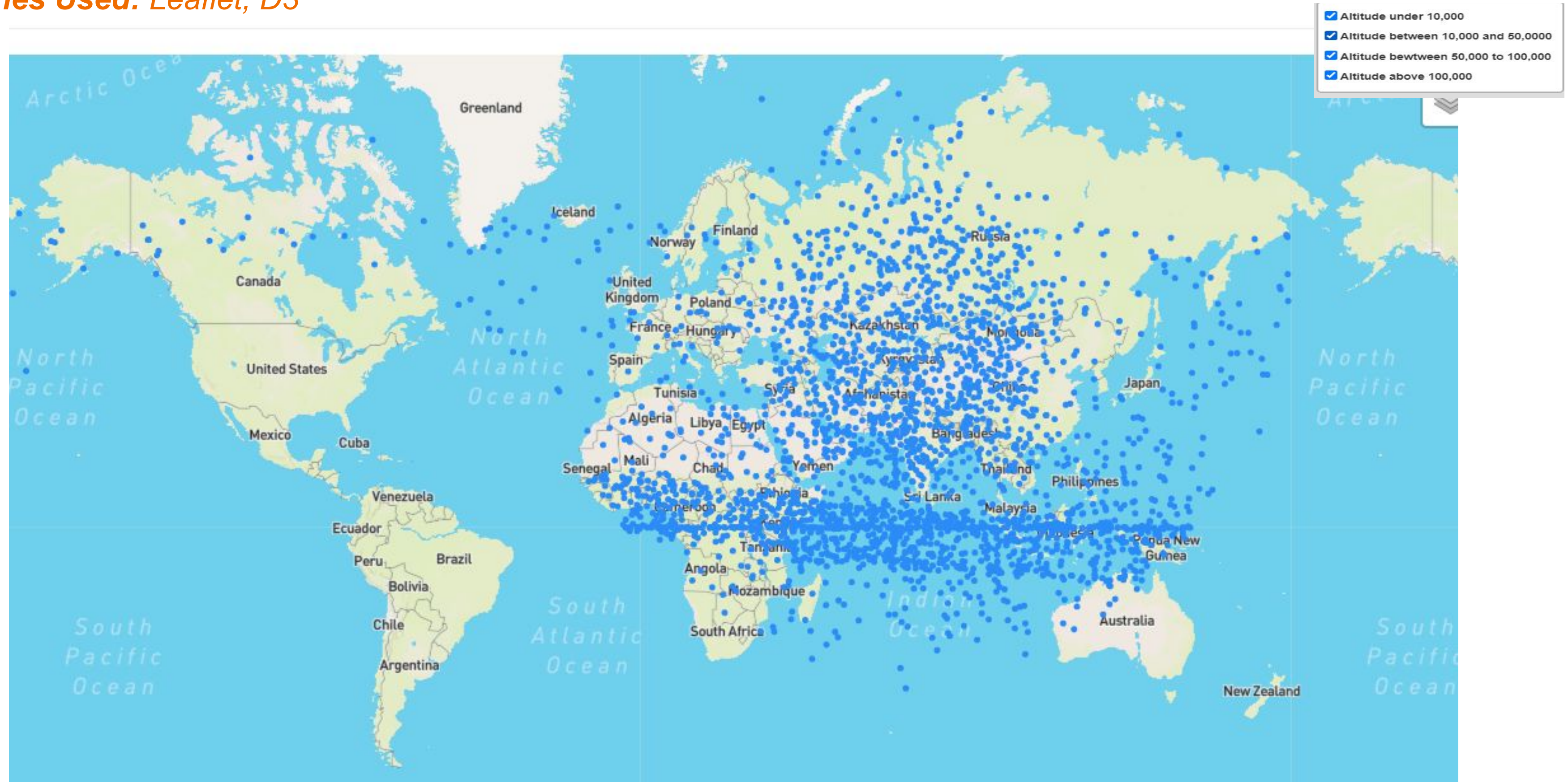


```
// Use D3 fetch to read the JSON file
// The data from the JSON file is arbitrarily named importedData as the a
d3.json("/static/data/satlaunch90.json").then((data) => {
  //console.log(data)

  let Before_1970 = 0;
  let From_1970to1980 =0;
  let From_1980to1990=0;
  let From_1990to2000 =0;
  let From_2000to2010 =0;
  let From_2010toNow=0;

  for (let i = 0; i < data[0].above.length; i++) {


    years = [data[0].above[i].launchDate]
      //console.log(years)

  var years2 = years[0].slice(0,4)
    //console.log(years2)
    if (years2 <= "1970"){

      Before_1970++;
      }
    else if(years2 <= "1980"){
      From_1970to1980++;
    }
```



Visualization: Altitude Map of Current Satellites

Libraries Used: *Leaflet, D3*



☑ Altitude under 10,000

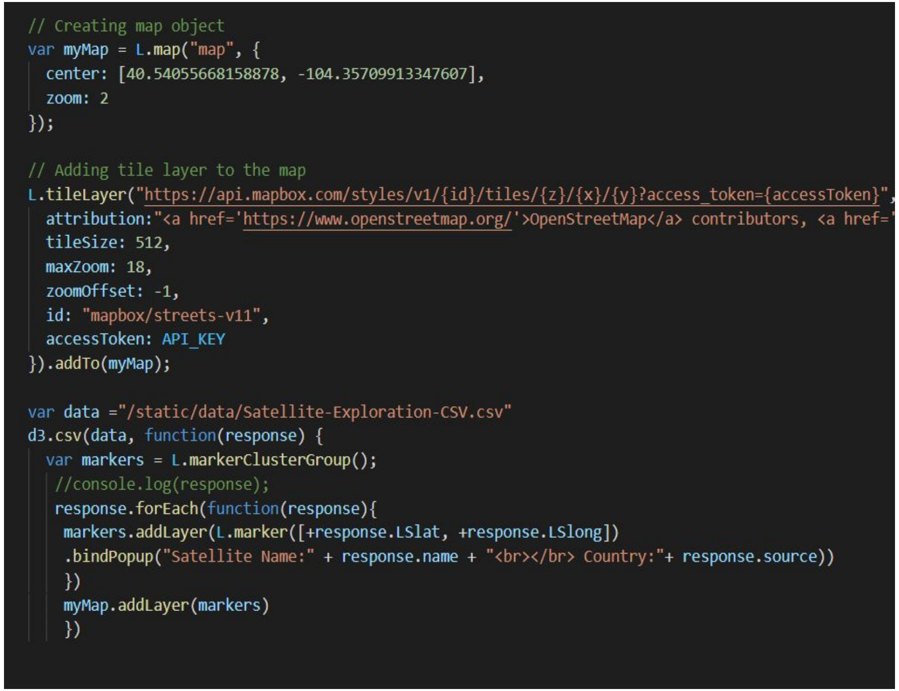
☑ Altitude between 10,000 and 50,0000

☑ Altitude bewtween 50,000 to 100,000

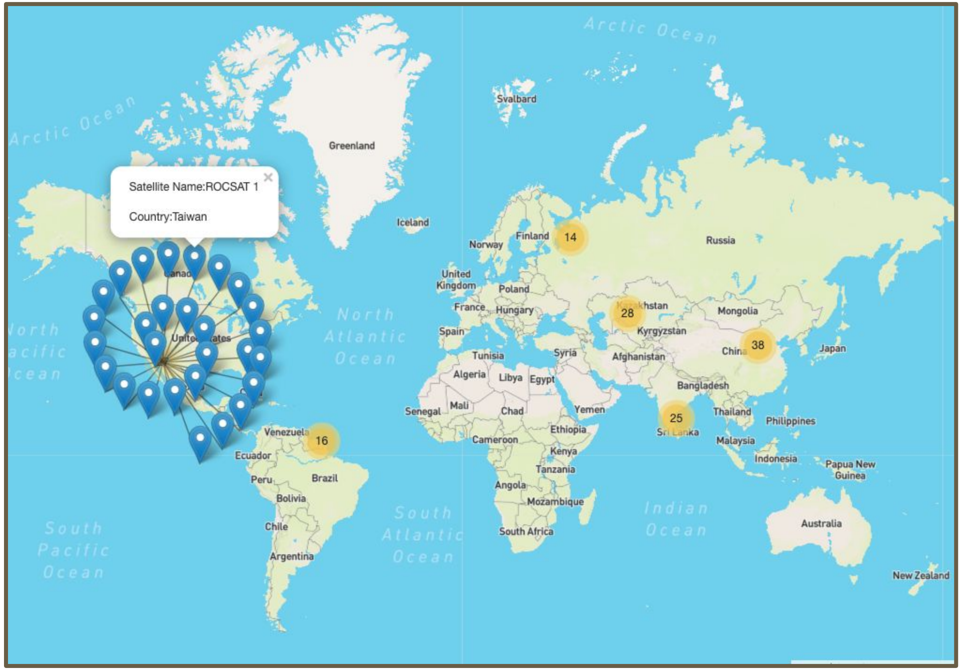
☑ Altitude above 100,000

Visualization: Cluster Map of Satellite Landing Sites

Libraries Used: *Leaflet, D3*



```
// Creating map object
var myMap = L.map("map", {
  center: [40.54055668158878, -104.35709913347607],
  zoom: 2
});

// Adding tile layer to the map
L.tileLayer("https://api.mapbox.com/styles/v1/{id}/tiles/{z}/{x}/{y}?access_token={accessToken}",
  attribution:"<a href='https://www.openstreetmap.org/'>OpenStreetMap</a> contributors, <a href='
  tileSize: 512,
  maxZoom: 18,
  zoomOffset: -1,
  id: "mapbox/streets-v11",
  accessToken: API_KEY
}).addTo(myMap);

var data ="/static/data/Satellite-Exploration-CSV.csv"
d3.csv(data, function(response) {
  var markers = L.markerClusterGroup();
   //console.log(response);
   response.forEach(function(response){
    markers.addLayer(L.marker([+response.LSlat, +response.LSlong])
    .bindPopup("Satellite Name:" + response.name + "<br></br> Country:"+ response.source))
    })
    myMap.addLayer(markers)
    })
```

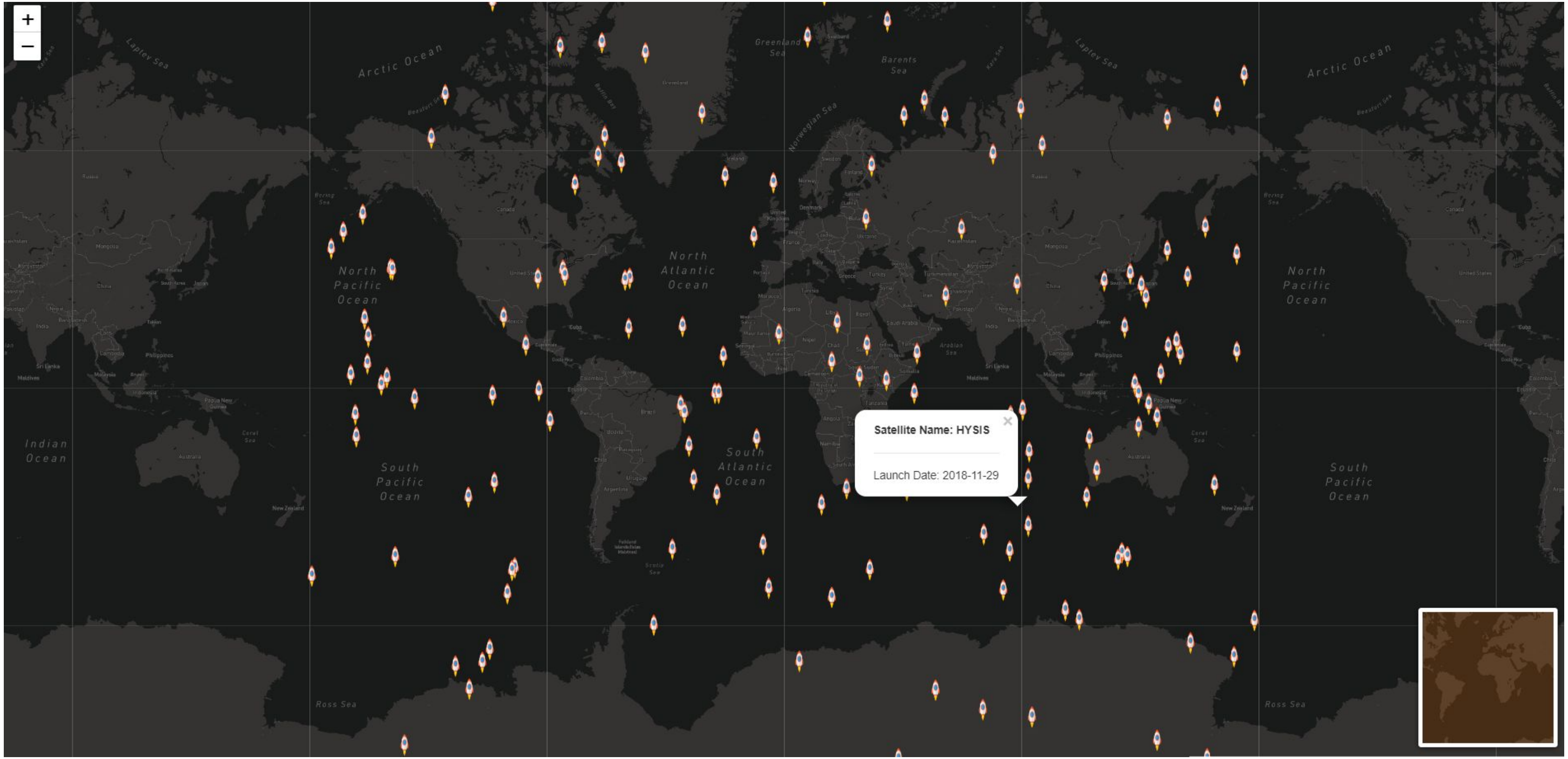


Satellite Name:ROCSAT 1

Country:Taiwan

Visualization: Current Earth Resources Satellites in Orbit

Libraries Used: *Leaflet, D3*



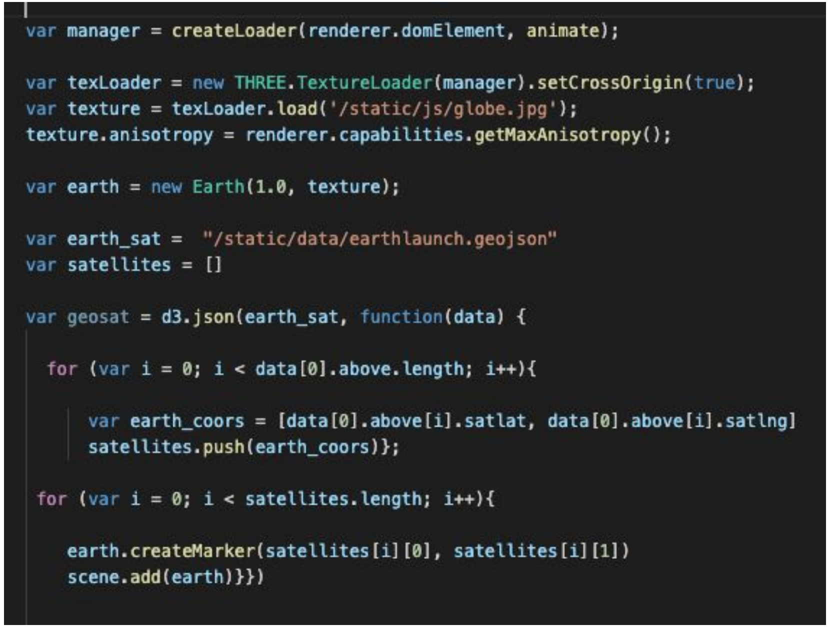
Satellite Name: HYSIS

Launch Date: 2018-11-29

Visualization: Current Earth Resources Satellites in Orbit

Libraries Used: *ThreeJS, D3*

Code Snapshot



```
var manager = createLoader(renderer.domElement, animate);

var texLoader = new THREE.TextureLoader(manager).setCrossOrigin(true);
var texture = texLoader.load('/static/js/globe.jpg');
texture.anisotropy = renderer.capabilities.getMaxAnisotropy();

var earth = new Earth(1.0, texture);

var earth_sat =  "/static/data/earthlaunch.geojson"
var satellites = []

var geosat = d3.json(earth_sat, function(data) {

  for (var i = 0; i < data[0].above.length; i++){

      var earth_coors = [data[0].above[i].satlat, data[0].above[i].satlng]
      satellites.push(earth_coors)};

 for (var i = 0; i < satellites.length; i++){

    earth.createMarker(satellites[i][0], satellites[i][1])
    scene.add(earth)}})
```