

Weather at some world cities

Final Project Data Programming

By:

Kernel Prieto, Jhoana Romero, Babitha Lewis and Bridget Ndubuisi

Outline



Our team



Topic introduction and prerequisites



Summary



Tools and development



Our final web application



References





Kernel Prieto-Data Analyst

Link: <https://www.linkedin.com/in/kernel-enrique-prieto-moreno-5b7a6a220/>



Jhoana Romero - Project Manager

Link: <https://www.linkedin.com/in/jhoanaromero/>



Babitha Lewis - Developer

Link: <https://www.linkedin.com/in/babitha-soniya-lewis-05638710b>



Bridget Ndubisi – Business Analyst

Link: No link

Our team

Topic Introduction



Weather in your city

[Guide](#) [API](#) [Pricing](#) [Maps](#) [Our Initiatives](#) [Partners](#) [Blog](#) [Marketplace](#) [Sign in](#) [Support](#)

Weather API

[Home](#) / [Weather API](#)

Please, [sign up](#) to use our fast and easy-to-work weather APIs for free. In case your requirements go beyond our freemium account conditions, you may check the entire list of our [subscription plans](#). You can read the [How to Start](#) guide and enjoy using our powerful weather APIs right now.

URL: <https://openweathermap.org/api>

Temperature
Humidity
Wind Pressure



Mexico city (Mexico)



Pasto (Colombia)

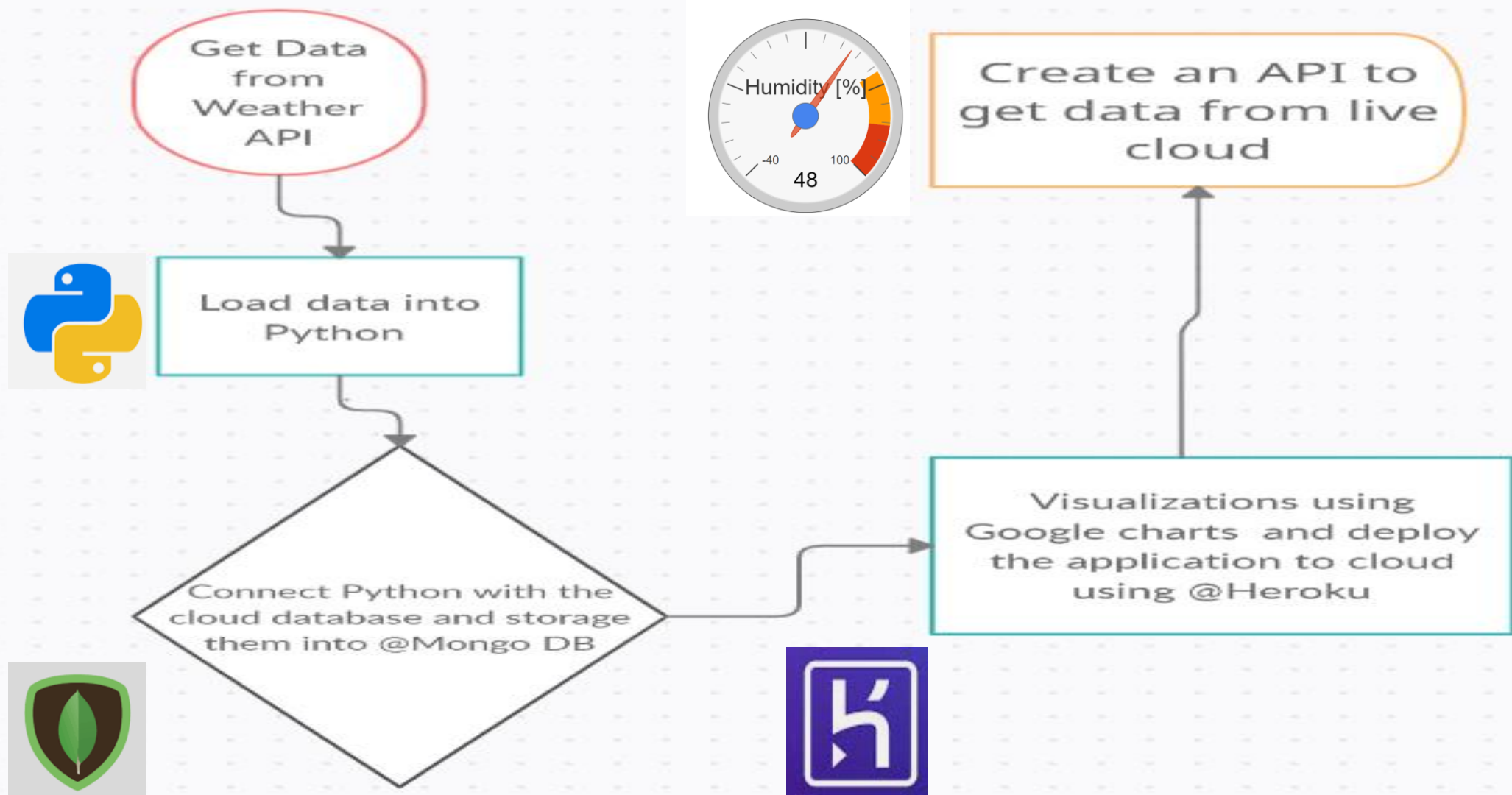


Manchester (UK)



Sapporo (Japan)

Summary



Prerequisites



Python



Visual Studio Code



Flask



MongoDB



Heroku



Basic understanding of Python and HTML



Basic understanding of Git

Tools and development

Starting MongoDB: Create a MongoDB Atlas Cluster

The screenshot displays the MongoDB Atlas web interface. The top navigation bar includes the organization name 'Kernel Enrique's Org...', 'Access Manager', 'Billing', 'All Clusters', 'Get Help', and a user profile dropdown. The left sidebar shows the navigation menu with sections: 'DEPLOYMENT' (Databases, Data Lake), 'DATA SERVICES' (Triggers, Data API), and 'SECURITY' (Database Access, Network Access, Advanced). The 'Databases' section is expanded, showing a list of databases under 'WeatherDB': Manchester, MexicoCity, Pasto, and Sapporo. The main content area is titled 'Collections' and shows 'DATABASES: 1' and 'COLLECTIONS: 4'. It includes a '+ Create Database' button, a search bar for namespaces, and a table of collections for 'WeatherDB'. The table has columns for Collection Name, Documents, Documents Size, Documents Avg, Indexes, Index Size, and Index Avg. A 'CREATE COLLECTION' button is visible in the top right of the collection list.

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
Manchester	8	3.78KB	484B	1	36KB	36KB
MexicoCity	99	48.08KB	498B	1	36KB	36KB
Pasto	11	5.61KB	522B	1	36KB	36KB
Sapporo	4	1.89KB	483B	1	36KB	36KB

Python
implementation:
Connect with
MongoDB Atlas
using python
'*pymongo*' module

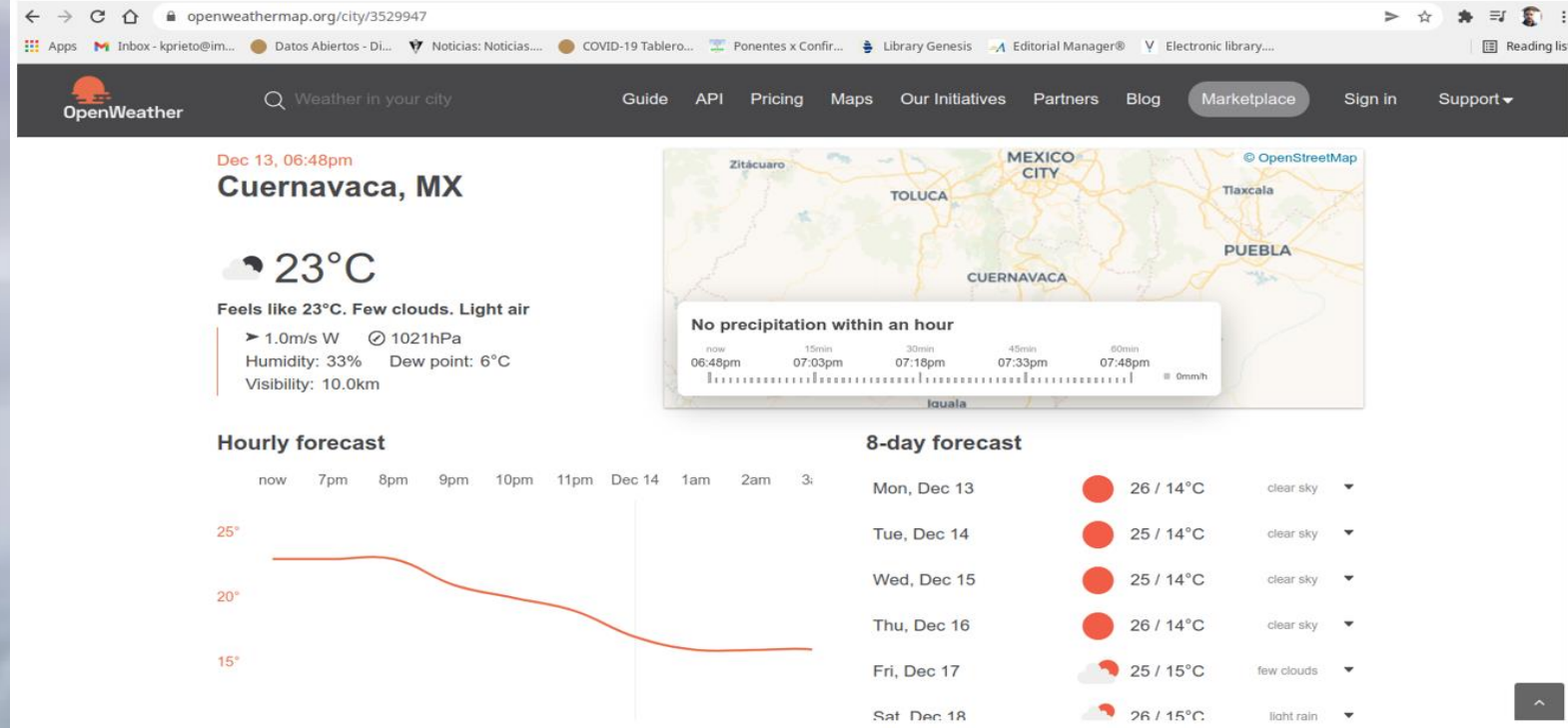
```
import requests
import time

from pymongo import MongoClient
from flask import Flask, render_template, jsonify, request

app = Flask(__name__)

client = MongoClient("mongodb+srv://Dexter:passwordABC@cluster0.o
db = client.get_database('WeatherDB')
```


Request and upload weather data to the live database



```
@app.route('/mexicocity')
def mexicocity():
    mexicocity = r1.json()
    cityName = mexicocity['name']
    country = mexicocity['sys']['country']
    localtime = mexicocity['timezone']
    cityLongitud = mexicocity['coord']['lon']
    cityLatitud = mexicocity['coord']['lat']
    temp_K = mexicocity['main']['temp'] - 273
    humidity = mexicocity['main']['humidity']
    pressure = 0.0295 * mexicocity['main']['pressure']
    wind_speed = mexicocity['wind']['speed']

    return render_template('MexicoCity.html', **locals())
```

```
r1=requests.get("https://api.openweathermap.org/data/2.5/weather?q=Mexico%20City&APPID=2ee475268973f586496f80acda284557")
r2=requests.get("https://api.openweathermap.org/data/2.5/weather?q=Pasto&APPID=2ee475268973f586496f80acda284557")
r3=requests.get("https://api.openweathermap.org/data/2.5/weather?q=Manchester&APPID=2ee475268973f586496f80acda284557")
r4=requests.get("https://api.openweathermap.org/data/2.5/weather?q=Sapporo&APPID=2ee475268973f586496f80acda284557")

if r1.status_code == 200:
    mexicocity = r1.json()
    print(mexicocity)
    db.MexicoCity.insert_one(mexicocity)
else:
    exit()

if r2.status_code == 200:
    pasto = r2.json()
    print(pasto)
    db.Pasto.insert_one(pasto)
else:
    exit()

if r3.status_code == 200:
    manchester = r3.json()
    print(manchester)
    db.Manchester.insert_one(manchester)
else:
    exit()

if r4.status_code == 200:
    sapporo = r4.json()
    print(sapporo)
    db.Sapporo.insert_one(sapporo)
else:
    exit()
```



Visualization: Gauge @Google Charts

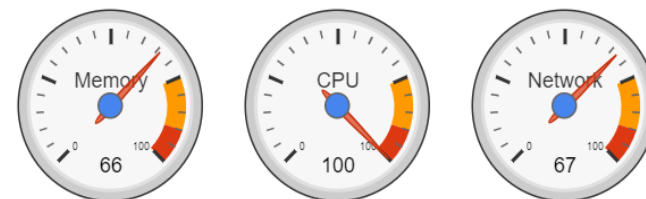
Visualization: Gauge

Overview

A gauge with a dial, rendered within the browser using SVG or VML.

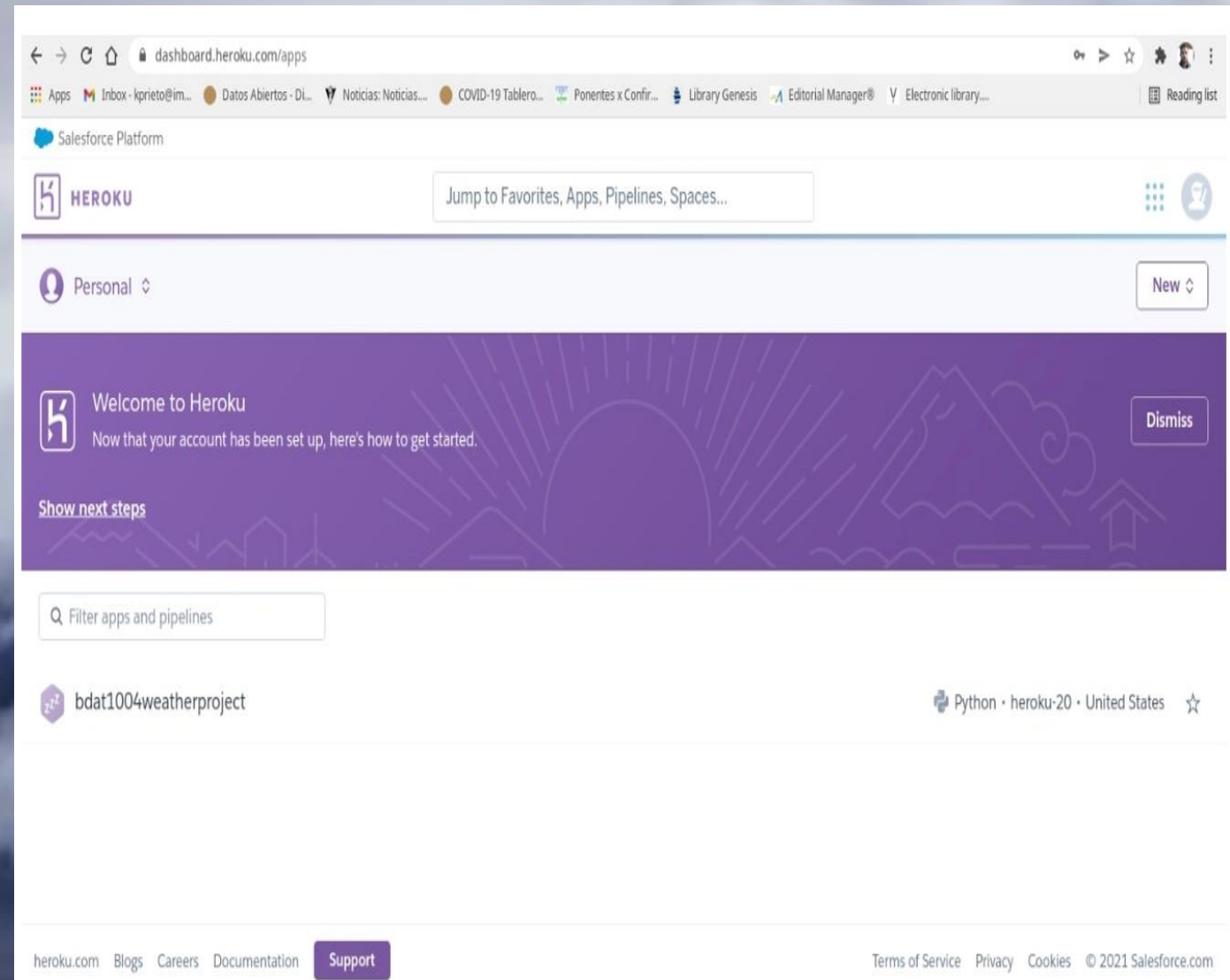
Example

Computational resources



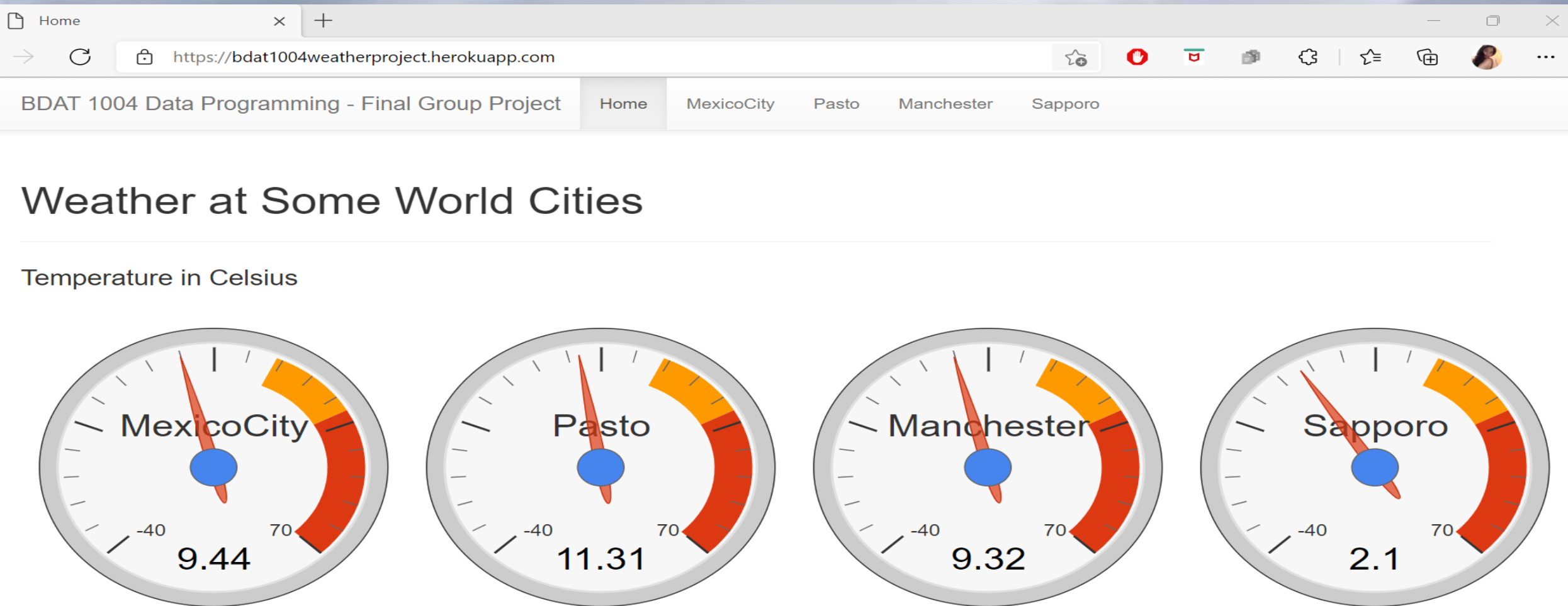
```
function drawChart() {  
  var data = google.visualization.arrayToDataTable([  
    ['Label', 'Value'],  
    ['Humidity [%]', {{humidity}}],  
    ['Wind [m/s]', {{wind_speed}}],  
    ['Pressure [Hg]', {{pressure}}],  
  ]);  
  
  var options = {  
    width: 1100, height: 450,  
    min: -40,  
    max: 100,  
    redFrom: 80, redTo: 100,  
    yellowFrom: 60, yellowTo: 80,  
    minorTicks: 5  
  };  
  
  var chart = new google.visualization.Gauge(document.getElementById('chart_div'));  
  
  chart.draw(data, options);  
}
```

@Heroku: The Python app that we have developed is residing on our local machine. But to make it available to end-users, we need to deploy it to either an on-premise server or to a cloud service. Heroku is one such cloud service provider.

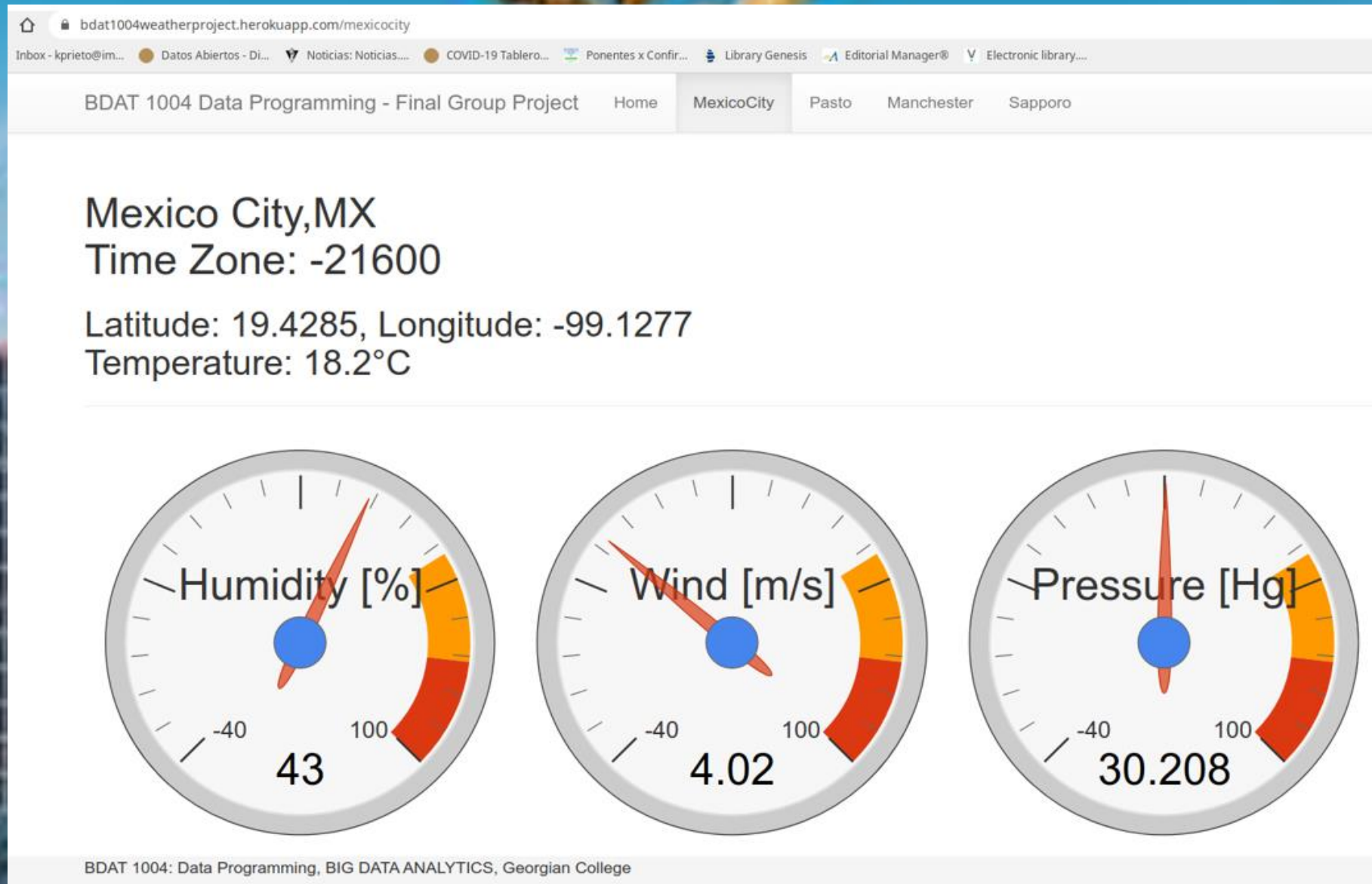


Web Application Home Page

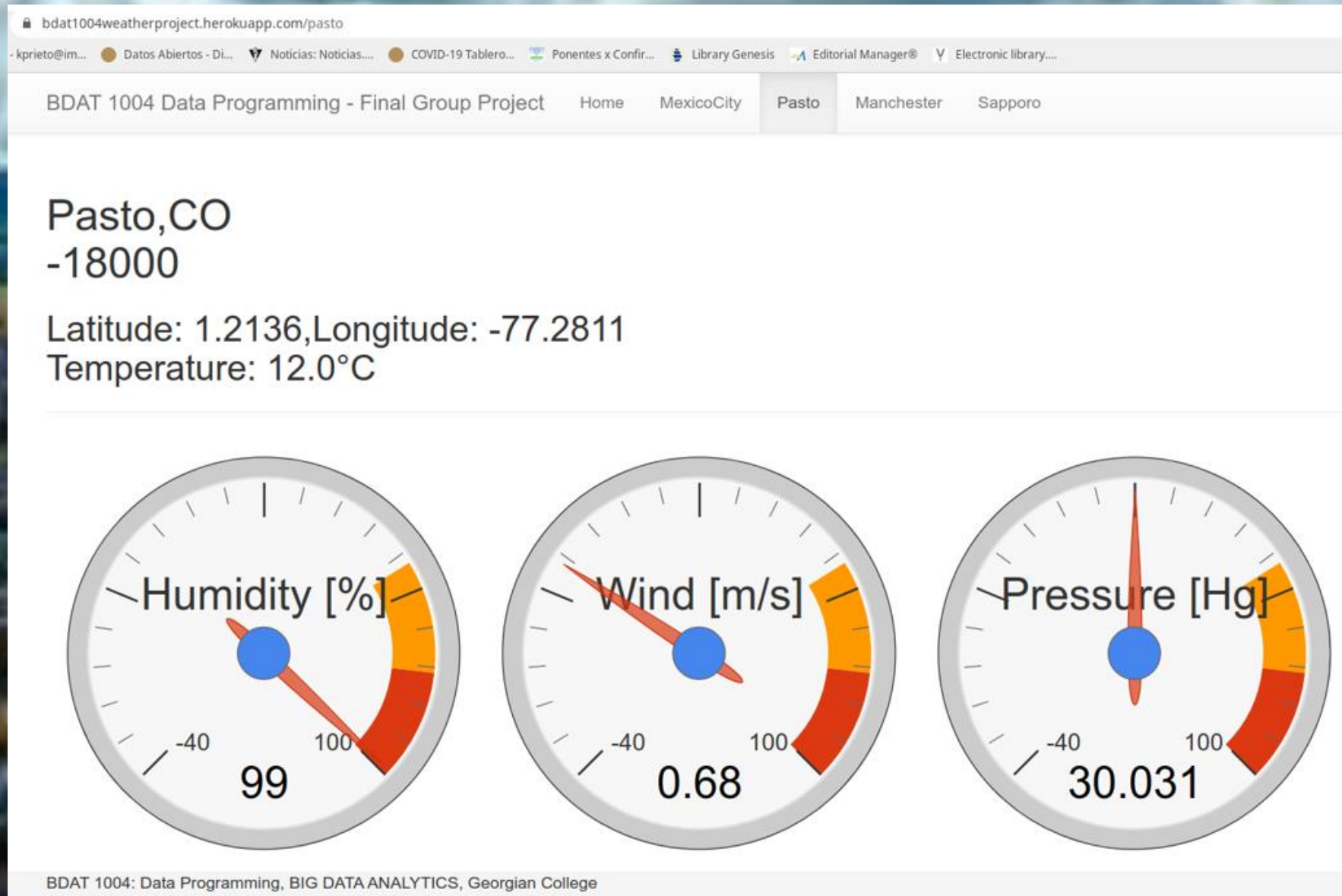
[Home \(bdat1004weatherproject.herokuapp.com\)](https://bdat1004weatherproject.herokuapp.com)



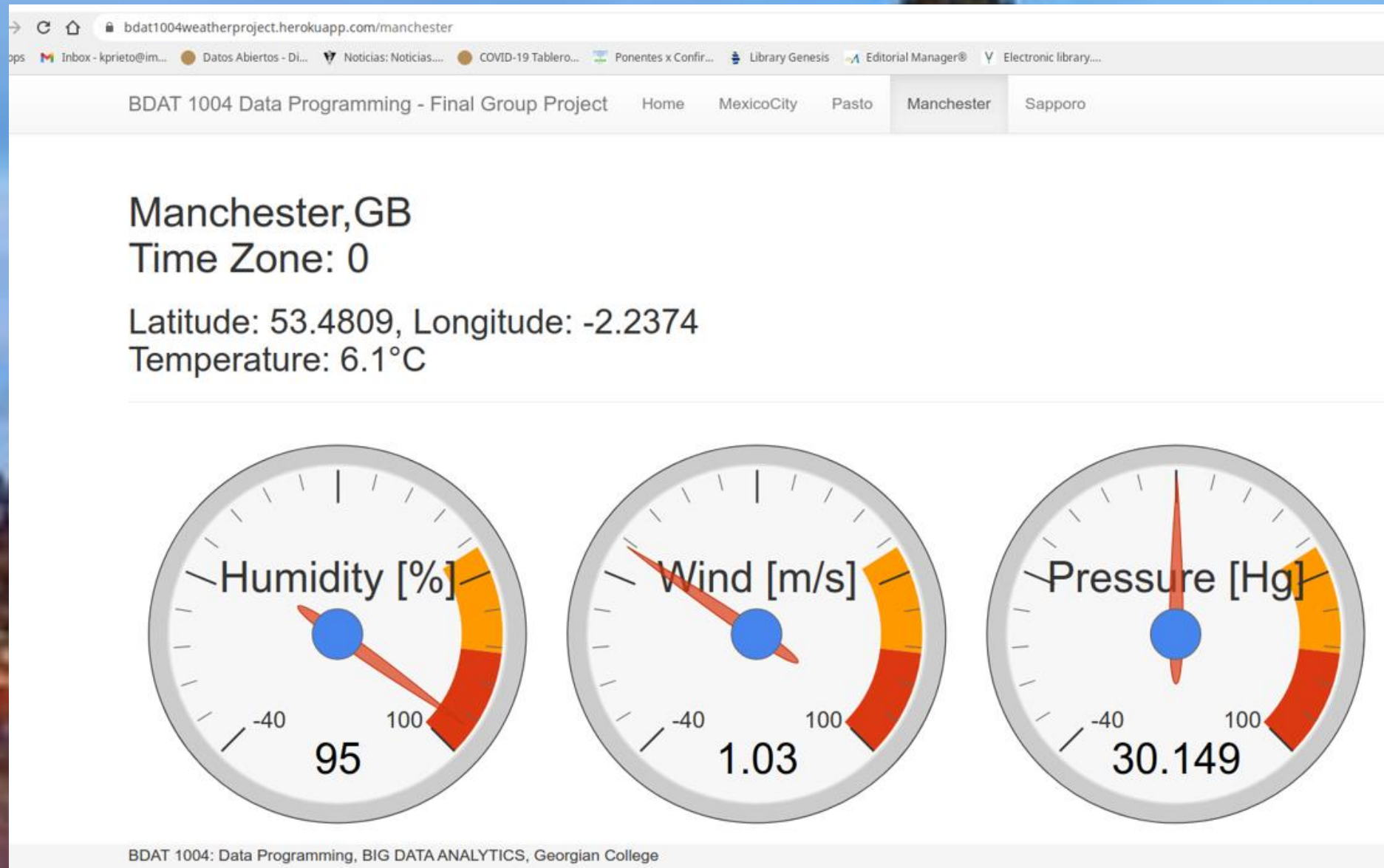
Mexico city (Mexico)



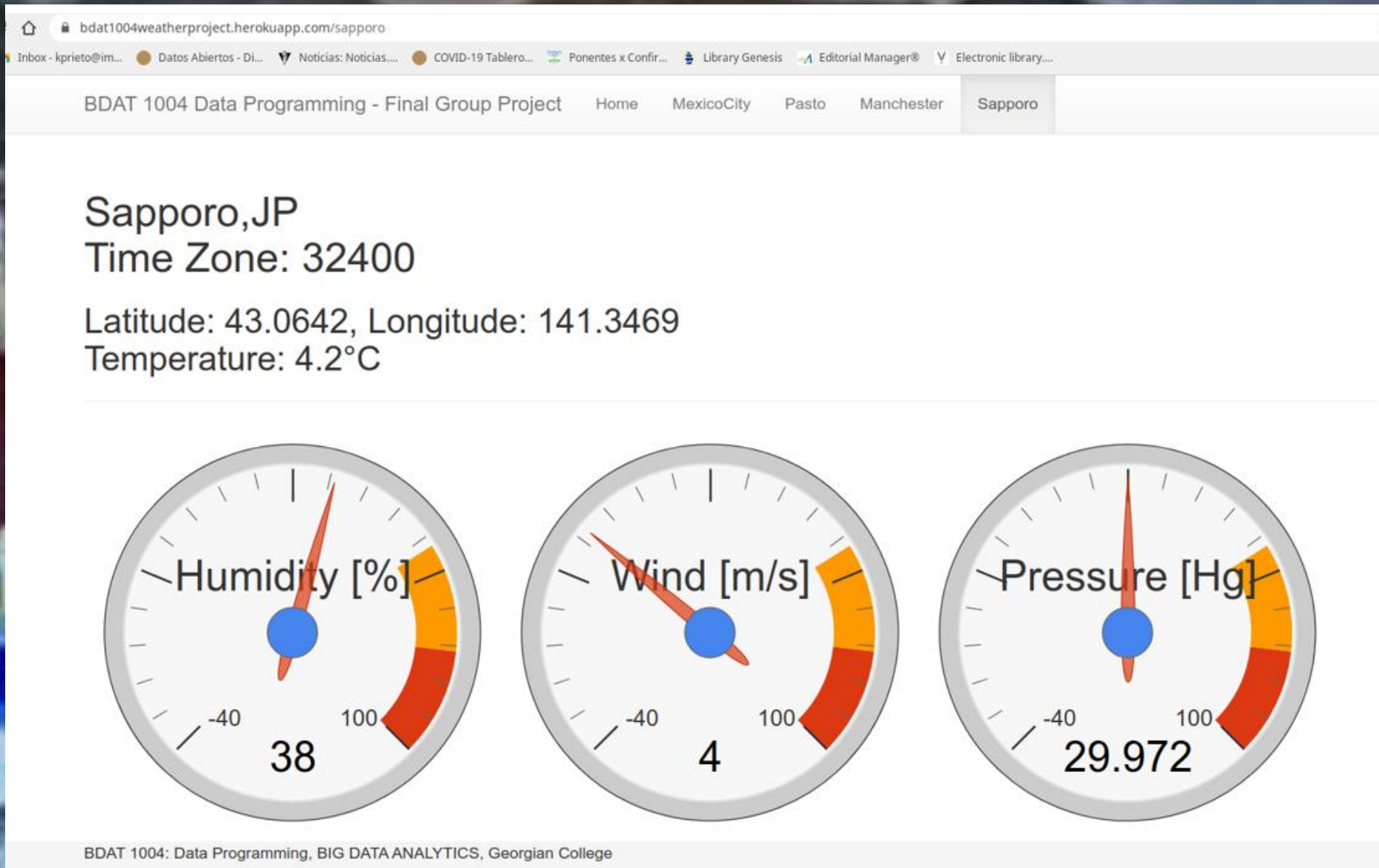
Pasto (Colombia)



Manchester (UK)



Sapporo (Japan)



References (acknowledgment)



1. Data programming course, Georgian College, Week 7 - Web frameworks and Google Charts , Ethan Davis Chanel

https://www.youtube.com/watch?v=6allmBXV2U4&ab_channel=EthanDavis



2. Turn your web scraper into a web app with Python and Flask, John Watson Rooney Chanel

https://www.youtube.com/watch?v=ukVjagiXJzw&ab_channel=JohnWatsonRooney



3. Review scraper from scratch till deployment, iNeuron

https://canvas.instructure.com/courses/2068451/files/96615045/download?download_frd=1