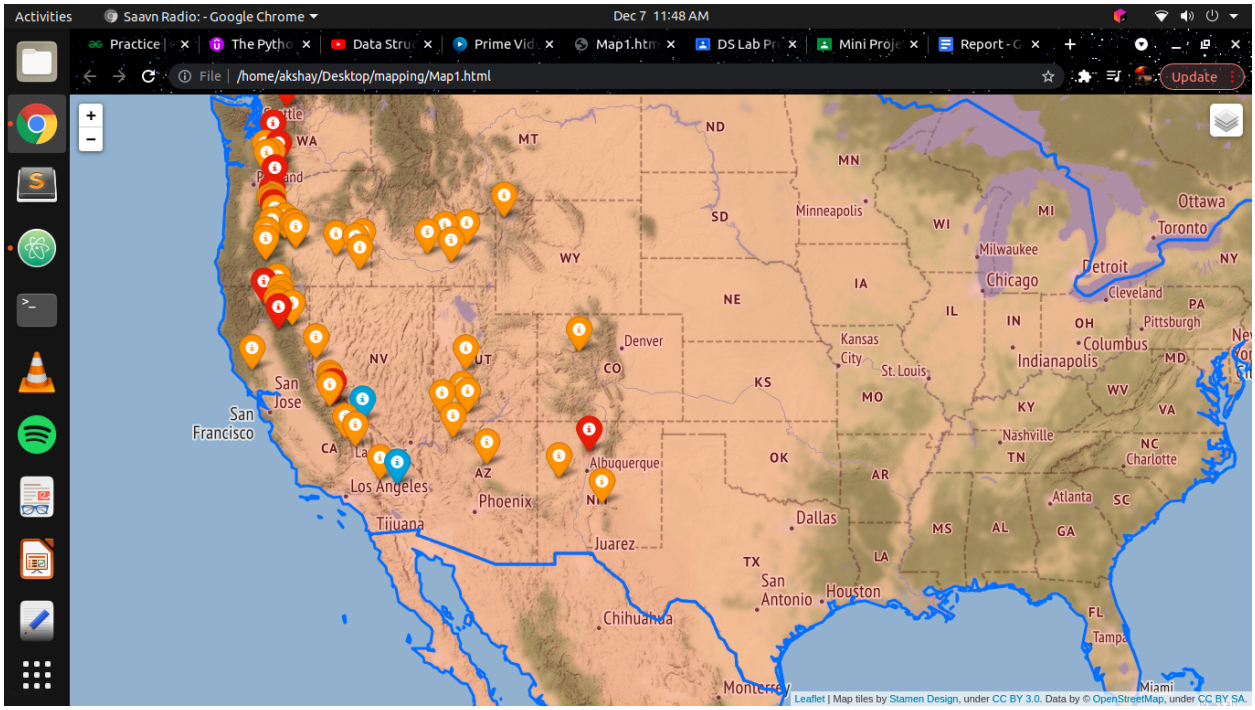
CTPY(COMPUTATIONAL THINKING WITH PYTHON)

PROJECT SYNOPSIS

WEB MAPPING: INTERACTIVE MAPPING OF POPULATION AND VOLCANOES



Submitted by-

KHUSHI KUMARI (ENG21AM3017)

Submitted to-

Professor Krishna Sowjanya

III semester, under the supervision of

Professor Krishna Sowjanya, Department of CSE. Dayananda Sagar University.

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that Khushi Kumari (ENG21AM3017) have satisfactorily completed her mini Project as prescribed by the University for the 3rd semester B.Tech. program in Computer Science & Engineering during the year 2020-21 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date:

Supervisor(s) Profesor Krishna Sowjanya ma'am

Department of Computer Science & Engineering

DECLARATION

I hereby declare that the work presented in this minor project entitled "**WEB MAPPING**" submitted to **DAYANANDA SAGAR UNIVERSITY, Bengaluru** is a record of work implemented under the guidance of **Professor Krishna Sowjanya,** Department of CSE, Dayananda Sagar University.

This project is submitted in the partial fulfilment of the requirements and conductive criticism for the award of III semester minor-project**.** The results embodied in this project have not been submitted elsewhere for the same.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be complete without the mention of people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are very much thankful to **Mrs.Krishna Sowjanya,** Professor, Department of CSE, Dayananda Sagar University, for her helpful tips and suggestions in carrying out this mini project successfully.

INTRODUCTION:

What is web mapping?

A web map is an interactive display of geographic information, in the form of a web page. Web maps are interactive. The term interactive implies that the viewer can interact with the map. This can mean selecting different map data layers and features like viewing, zooming in and out into a particular part of the map.

Web maps are also said to be powered by the web, rather than just digital maps on the web. This means that the map is usually not self-contained; in other words, it depends on the internet. At least some of the content displayed on a web map is usually loaded from other locations on the web, such as a tile server.

LIBRARIES USED TO CREATE WEB MAP:

- FOLIUM

Folium is **a Python library used for visualizing geospatial data**. It is easy to use and a powerful library.

- JSON

JSON **stands for JavaScript Object Notation**. JSON is a lightweight format for storing and transporting data. JSON is often used when data is sent from a server to a web page. JSON is "self-describing" and easy to understand.

- HTML

HTML (**Hypertext Markup Language**) is the code that is used to structure a web page and its content.

- PANDAS

Pandas is a library used for data manipulation and analysis. It is used to extract the data and store it in the desired format.

STEPS TO CREATE A WEB MAP:

1. CREATING AN HTML MAP WITH PYTHON

https://att-c.udemycdn.com/2021-02-22_10-12-55-43b7eef1125e7debad8a484653be815c/original.zip?response-content-disposition=attachment%3B+filename%3DWebmap_datasources.zip&Expires=1638876474&Signature=GaM4nktX11IMJQJFdffTuqfsyPRdRciZCQfK9S2fd2wKj4crzCRFMFwHwmRpSUp81RWyZjc1iLHLerATVA1DU~ZoaBg4MjiiTs-JacCsEjWxcJndCTxHXrWHWlV5NIJW4EaOf8-03ZmsKWWjaXucUxR8v2AHI9HZlyRp~24H~t~Qq3T9SPwcMtR4C6v3NQ07aqFqiQpy5fxNSH~NyydOtyncZqT6qHfkyE8BSuKKCuJi6mieiL9UqqxzFzc161K3J59x1lqJh~XDk8q8Q1shW1bMvkFrQNqqS6wwyYq-pBJDDVNUXqx-jqntlR43KVE6m5zcGq11W732VRtl9Fq7cg__&Key-Pair-Id=APKAITJV77WS5ZT7262A

2. ADDING A MARKER TO THE MAP (adding markers on the volcanoes using a text file called volcanoes which consists geographic data about volcanoes)

https://att-c.udemycdn.com/2020-04-01_11-00-58-afb9373c2527072ddc531ab2832f24f5/original.txt?response-content-disposition=attachment%3B+filename%3DVolcanoes.txt&Expires=1638534776&Signature=bu~JLSRZxigHszKOHrdLB8lZsHvk2isNIFL5-utwyXK2RPa2Y7WGgX4ywxFIhMmFK7-wCQjiMJUI3meDdUj5Ke~FYD5iqYiqIbrrUyDTcqfwvl~oXf9TE2TEJ80tTcpcqAqQbFou-tKGAI5qrjXdrMNQRK5DG~60Rmhq6MofcBr9JXYedvKseQ0azxUpkMl~FPU1NRC142kzZVX7CUPyKXaOnklwhLfUCIOBGpdHsK-0VJwrPYC9UaF4qkBTKuselk4UdG1S03OrdKbPPZ5TVt2enqOuOI5n3YCNUQbphVjcyhq6~OhhSALlvmpWeieCVnTFIWj1dPYokaQlwx8q3Q__&Key-Pair-Id=APKAITJV77WS5ZT7262A

3. USING FOR-LOOPS WHILE ADDING MULTIPLE MARKERS.

4. USING FILE PROCESSING WHILE ADDING MARKERS FROM FILES.

5. USING STRING MANIPULATION WHILE ADDING TEXT ON THE MAP POPUP WINDOW.

6. USING FUNCTIONS WHILE GENERATING A COLOR GENERATION FUNCTION ON MARKERS.

7. USING POPULATION JSON DATA.

8. USING JSON DATA WHILE ADDING A POPULATION MAP LAYER FROM THE DATA.

9. STYLIZE THE POPULATION LAYER.

10. ADD A CONTROL PANEL LAYER.

CODE:

```
import folium

import pandas

data = pandas.read_csv("Volcanoes.txt")

lat = list(data["LAT"])

lon = list(data["LON"])

elev = list(data["ELEV"])

def color_producer(elevation):

    if elevation < 1000:

        return 'blue'

    elif 1000 <= elevation < 3000:

        return 'orange'

    else:

        return 'red'

'''
```

```
def countrycolor(data):
    pop = data["properties"]["POP2005"]
    if pop < 10000000:
        return{"colorFill" : "green"}
    elif 10000000 <= pop <= 40000000:
        return{"colorFill" : "orange"}
    else:
        return{"colorFill" : "red"}
'''
map = folium.Map(location = [38.58, -99.09], zoom_start=6, tiles = "Stamen Terrain")
fg = folium.FeatureGroup(name="My Map")
fgv = folium.FeatureGroup(name="Volcanoes")
for lt, ln, el in zip(lat, lon, elev):
                        fg.add_child(folium.Marker(location=[lt, ln], popup=str(el)+ " m",
icon=folium.Icon(color=color_producer(el))))
fgp = folium.FeatureGroup(name="Population")
fg.add_child(folium.GeoJson(data=open("world.json", 'r', encoding='utf-8-sig').read(),
style_function=lambda x:{'fillColor':'yellow' if x["properties"]["POP2005"]<10000000
else 'orange' if 100000000 <= x["properties"]["POP2005"]<20000000 else 'red'}))
map.add_child(fg)
map.add_child(fgv)
```

```
map.add_child(fgp)

map.add_child(folium.LayerControl())

map.save("Map1.html")
```

OUTPUT: