



Shree Rahul Education Society's (Regd.)

SHREE L. R. TIWARI COLLEGE OF ENGINEERING

Kanakia Park, Near Commissioner's Bungalow, Mira Road (East), Thane 401107, Maharashtra

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

Tel. No.: 022-65295732 / 022-65142376 | Email: slrtce@rahuleducation.com | Website: www.slrtce.in

Skill Base Lab Course: Python Programming (CSL405)

Mini Project Report

Geolocation

Subject	Skill Base Lab Course: Python Programming (CSL405)
Course	S. E. (Computer Engineering)
Year	2020 - 2021
Semester	Semester - IV
Subject In charge	K. Jayamalini
Name of the Group Members	1. SURAJ MAURYA (29) 2. ADITYA KINI (24) 3. DEVANAND GUPTA (12)

1. Introduction

Geolocation is one of the features that is often labelled as a HTML5 feature.

Actually, Geolocation is not part of the HTML5 specification, it has its own specification (Geolocation API Specification) and has been created by the W3C rather than the WHATWG but it is worth to discuss about it since it is going to be part of the features that will help developers creating great web applications.

Using geolocation, you can allow your users to:

- To get the current location
- Map a route between two location
- Distance Calculator
- Track a phone number

Social networking services have a lot of interest on geolocation. Facebook, for instance, recently released places that allow mobile users to share their location, tag friends at some places using geolocation.

Geolocation sounds exciting but there are some privacy concerns, the main one being: "Do I have the ability to accept or refuse which website is getting my location?"

The specification clearly states the following:

User agents must not send location information to Web sites without the express permission of the user. User agents must acquire permission through a user interface, unless they have prearranged trust relationships with users, as described below. The user interface must include the URI of the document origin.

Once a user allows the user agent to share the user location with the server, there are still some privacy concerns regarding how long the location data is stored? Is it shared with other sites/partners? Can the location data be updated or deleted by the end user?

The specification recommends the following:

Recipients must clearly and conspicuously disclose the fact that they are collecting location data, the purpose for the collection, how long the data is retained, how the data is secured, how the data is shared if it is shared, how users may access, update and delete the data, and any other choices that users have with respect to the data.

2. Implementation

```
from geopy import distance
from geopy.geocoders import Nominatim
from opencage.geocoder import OpenCageGeocode
import phonenumbers
import pyttsx3
from phonenumbers import geocoder, carrier
import folium

# from myNumber import number
from phonenumbers import geocoder

Key = '204d5c6afbcf4ae88df9e85903da9473' ##Api key ,So what is an api? opencage

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)

def speak(audio):
    engine.say(audio)
    engine.runAndWait()

outer_count = 100
while outer_count!=200:
    print("\n\n      PLEASE SELECT ONE OF THE SERVICE YOU WANT TO USE : \n\n 1>
Location: \n 2> Map A Route Between Two Locations: \n 3> Distance Calculator : \n 4> Track A
Phone Number : \n 5> Exit :  ")
    speak("PLEASE SELECT ONE OF THE SERVICE YOU WANT TO USE 1 Location Details
or 2 Map A Route Between Two Locations Or 3 Distance Calculator Or 4 Track A Phone Number
")

    n = int(input("\n >>> "))

    # 1> Get Your Current Location:

    if n==1:
        # print('1')
        print("      1> Current Location Co-ordinates : \n      2> Destination Location Co-ordinates : \n
")
        speak("      1 Get Your Current Location Coordinates      2 Get Destination Location
Coordinates ")
        x= int(input(" \n>>> "))
        if x==1 :
            import requests
            res = requests.get('https://ipinfo.io/')
            data = res.json() ## ipinfo returns the data in a specific format, so we are converting it with
the help of .json for custom look
            # print(res.text)

            country =str(data['country'])
```

```

region = str(data['region'])
city = str( data['city'] )
location = data['loc'].split(',')
latitude = str( location[0] ) ## conversion to string for speech recognition
longitude = str( location[1] )

print("Country  :", country)
speak(country+ ' is your country')
print("Region   :", region)
speak(region+ ' is your region')
print("City     :", city)
speak(city+ ' is your city')

print("Latitude :",latitude)
print("Longitude :",longitude)
speak(latitude+' is the latitude')
speak(longitude+' is the longitude')

print("      \n 1> MAIN MENU\n 2> EXIT      ")
speak("      1> for MAIN MENU      \n2> for EXIT      ")
y = int(input(" >>> "))
if y == 2:
    outer_count=200
    print('      THANK YOU FOR USING OUR SERVICES!\n      COMEBACK
SOON.. 😊😊')
    speak('      THANK YOU FOR USING OUR SERVICES!\n      COMEBACK SOON.. ')
else:
    continue

if x==2 :

from geopy.geocoders import Nominatim

Geolocator = Nominatim(user_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.72 Safari/537.36")

adress = input("Enter the address : ")
loc = Geolocator.geocode(adress) # object
print(loc)
print("Destination : ")
latitude = str(loc.latitude)
longitude = str(loc.longitude)

print("      Latitude : "+latitude)
print("      Longitude : "+longitude)
speak(latitude+' is latitude'+longitude+' is longitude of your destination')

```

```

print("    \n 1> MAIN MENU\n 2> EXIT    ")
speak("    1> for MAIN MENU    \n2> for EXIT    ")
y = int(input(" >>> "))
if y == 2:
    outer_count=200
    print('    THANK YOU FOR USING OUR SERVICES!\n    COMEBACK
SOON.. 😊😊')
    speak('    THANK YOU FOR USING OUR SERVICES!\n    COMEBACK SOON.. ')
else:
    continue

```

```

if n==2:

```

```

# initialize Nominatim API
geolocator = Nominatim(user_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.72 Safari/537.36")

```

```

# place input
Input_place1 = str(input(" From : "))
Input_place2 = str(input(" To : "))

```

```

# Get location of the input strings
place1 = geolocator.geocode(Input_place1)
place2 = geolocator.geocode(Input_place2)

```

```

print(place1)
print(place2)

```

```

# Get latitude and longitude
Loc1_lat, Loc1_lon = (place1.latitude), (place1.longitude)
Loc2_lat, Loc2_lon = (place2.latitude), (place2.longitude)

```

```

location1 = (Loc1_lat, Loc1_lon)
location2 = (Loc2_lat, Loc2_lon)

```

```

print(location1[0],location1[1])

```

```

import openrouteservice

```

```

client=openrouteservice.Client(key='5b3ce3597851110001cf624823b437fa956f41608563b0a2210
8e3ae') # Specify your personal API key

```

```

coordinates = [[location1[1], location1[0]], [location2[1], location2[0]]] # lon , lat

```

```

print("    \n** Please Select The Mode **\n    1> Driving :\n    2> Walking : ")
speak(" Please Select The Mode ")
speak("    1 Driving    2 Walking ")

```

```

r = int(input(" >>> "))
mode = 'driving-car'

if r==1 :
    route = client.directions(coordinates=coordinates,profile='driving-car', format='geojson')
if r==2 :
    route = client.directions(coordinates=coordinates,profile='foot-walking', format='geojson')

map_directions = folium.Map(location=[float(location1[0]), float(location1[1])], zoom_start=9)
folium.GeoJson(route, name='route').add_to(map_directions)

folium.LayerControl().add_to(map_directions)

map_directions.save("map_directions.html")

print("    * Your HTML file has been generated.  * ")
speak("    * Your HTML file has been generated.  * ")

print("    \n 1> MAIN MENU\n 2> EXIT    ")
speak("    1> for MAIN MENU    \n2> for EXIT    ")
y = int(input(" >>> "))
if y == 2:
    outer_count=200
    print('        THANK YOU FOR USING OUR SERVICES!\n        COMEBACK SOON.. 😊😊')
    speak('    THANK YOU FOR USING OUR SERVICES!\n    COMEBACK SOON.. ')
else:
    continue

if n==3:
    # initialize Nominatim API
    geolocator = Nominatim(user_agent="geopiExercises")

    # place input
    Input_place1 = str(input(" From : "))
    Input_place2 = str(input(" To : "))

    # Get location of the input strings
    place1 = geolocator.geocode(Input_place1)
    place2 = geolocator.geocode(Input_place2)

    print(place1)
    print(place2)

    # Get latitude and longitude
    Loc1_lat, Loc1_lon = (place1.latitude), (place1.longitude)
    Loc2_lat, Loc2_lon = (place2.latitude), (place2.longitude)

    location1 = (Loc1_lat, Loc1_lon)

```

```

location2 = (Loc2_lat, Loc2_lon)

print(distance.distance(location1, location2).km, " Kms")
dist = str(distance.distance(location1, location2).km)
speak("Distance from "+Input_place1 + " to " +
      Input_place2 + " is "+dist + " kilometers")


print("      \n 1> MAIN MENU\n 2> EXIT      ")
speak("      1> MAIN MENU      \n2>EXIT      ")
y = int(input(" >>> "))
if y == 2:
    outer_count=200
    print('      THANK YOU FOR USING OUR SERVICES!\n      COMEBACK
SOON.. 😊😊')
    speak('  THANK YOU FOR USING OUR SERVICES!\n      COMEBACK SOON.. ')
else:
    continue


if n==4:
    print("Please enter the number along with country code :")
    speak('Please enter the number along with country code')
    number = input("\n >>> ")


    some_Number = phonenumbers.parse(number) #prasing String to phone number #split the
phononumber into two parts country code && the actual number


    country= str(geocoder.country_name_for_number(some_Number,'en') )
    your_Location = geocoder.description_for_number(some_Number, 'en')


    print("Country : "+country)
    speak("Country is "+country)


    print(your_Location)
    speak(your_Location +'is your region')


    geocoder = OpenCageGeocode(Key)
    query = str(your_Location)
    results = geocoder.geocode(query) # reurns multiple values


    print(some_Number)
    Carrier = carrier.name_for_number(some_Number, 'en') # get service provider
    print(carrier.name_for_number(some_Number, 'en'))
    speak('Your carrier name is'+Carrier)

```

```

# print(results)

lat = results[0]['geometry']['lat']
lng = results[0]['geometry']['lng']

latitude = str(lat)
longitude = str(lng)

print("Latidue : "+latitude + " Longitudde : "+longitude)

speak(latitude+' is the latitude')
speak(longitude+' is the longitude')


myMap = folium.Map(location=[lat, lng], zoom_start=9)

folium.Marker([lat, lng], popup=your_Location).add_to(myMap)

## save map in html file

myMap.save("phone_NumberLocation.html")
print("    * Your HTML file has been generated.  * ")
speak("    * Your HTML file has been generated.  * ")

print("    \n 1> MAIN MENU\n 2> EXIT    ")
speak("    1> for MAIN MENU    \n2> for EXIT    ")
y = int(input("  >>> "))
if y == 2:
    outer_count=200
    print('        THANK YOU FOR USING OUR SERVICES!\n        COMEBACK
SOON.. 😊😊')
    speak('    THANK YOU FOR USING OUR SERVICES!\n    COMEBACK SOON.. ')
else:
    continue

if n==5 :
    outer_count=200
    print('        THANK YOU FOR USING OUR SERVICES!\n        COMEBACK SOON..
😊😊')
    speak('    THANK YOU FOR USING OUR SERVICES!\n    COMEBACK SOON.. ')

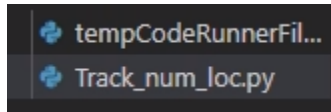
```


3. Result

1. Starting the Program

On opening the desired folder, we can see a files, “*track_num_location*”. The ““*track_num_location*” is the python file which is runnable and which starts the program.

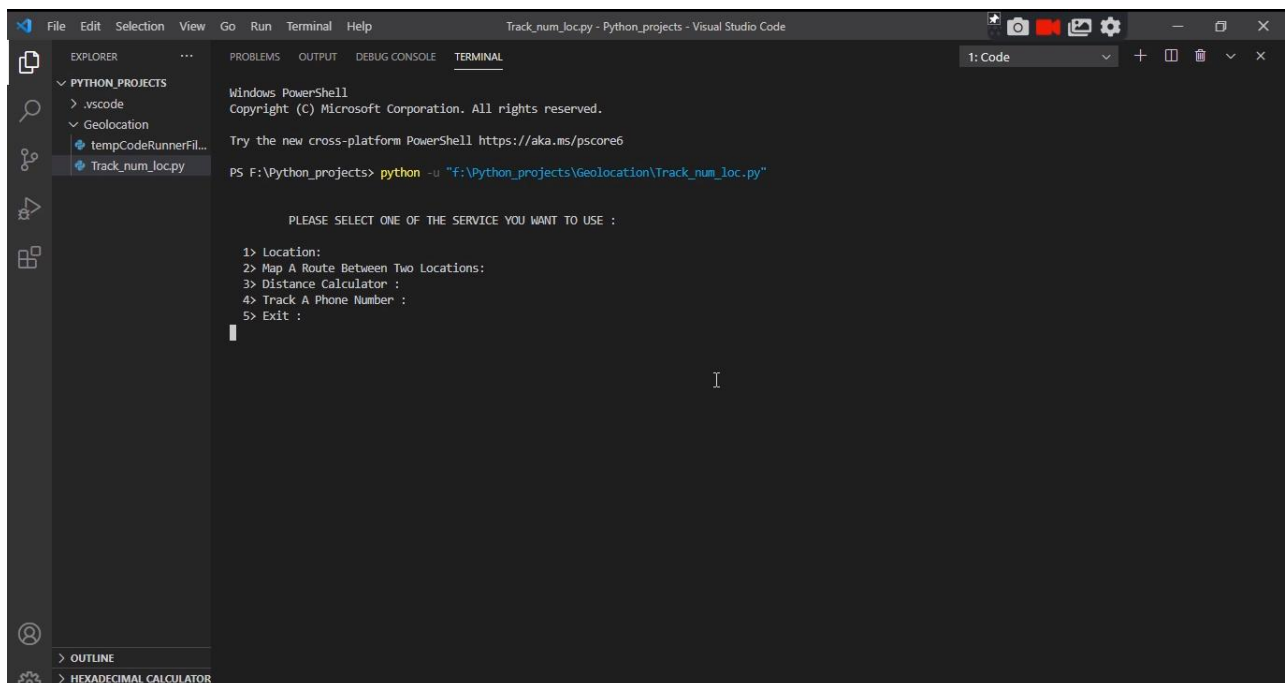
So to start the program, we must double click on the ““*track_num_location*” file.



2. Start Page

When we run the program an interface in terminal will appear which will ask to choose between 5 options and it will also readout the options for the users. The following are the displayed option,

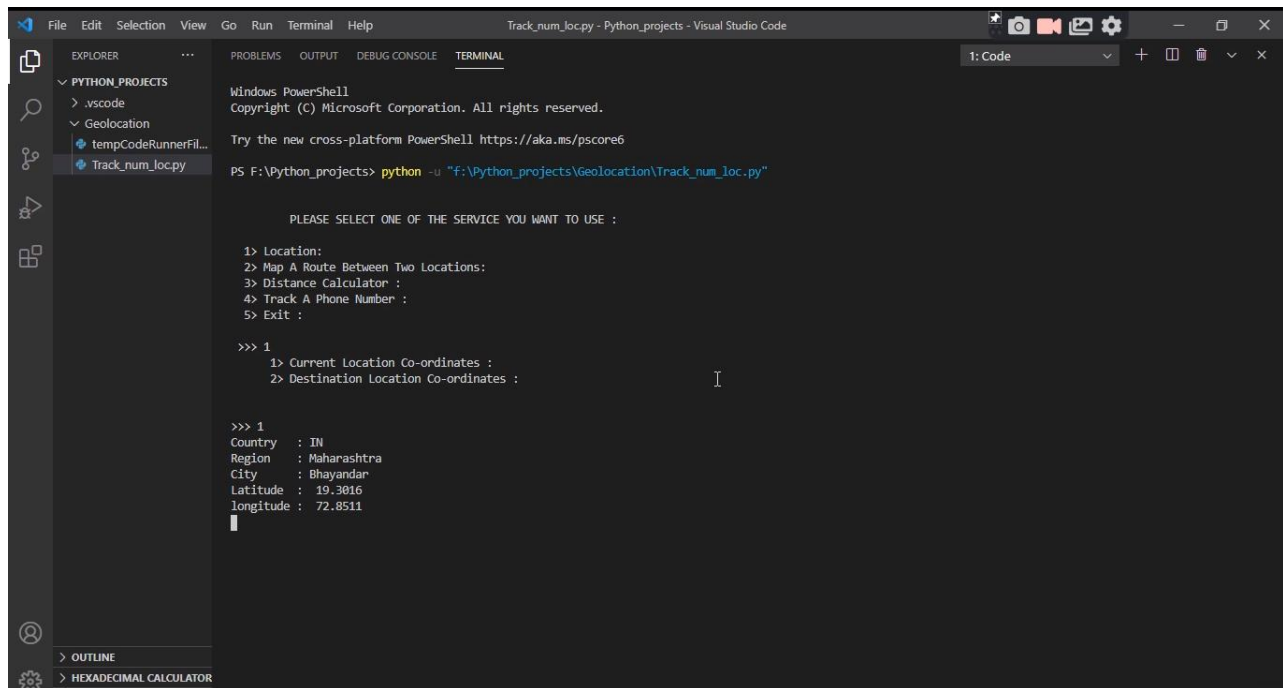
- To get the current location
- Map a route between two location
- Distance Calculator
- Track a phone number
- Exit



1) Location

Once we select the location option it will ask whether to get the current location co-ordinates or destination location co-ordinates. If the user select current location co-ordinates it will print the current country,region,city and co-ordinates of the current location. If the user select destination location co-ordinates then the system will ask for a specific destination once the user enter the destination the system will direct that location and will display the address and longitude and latitude of the correct location .

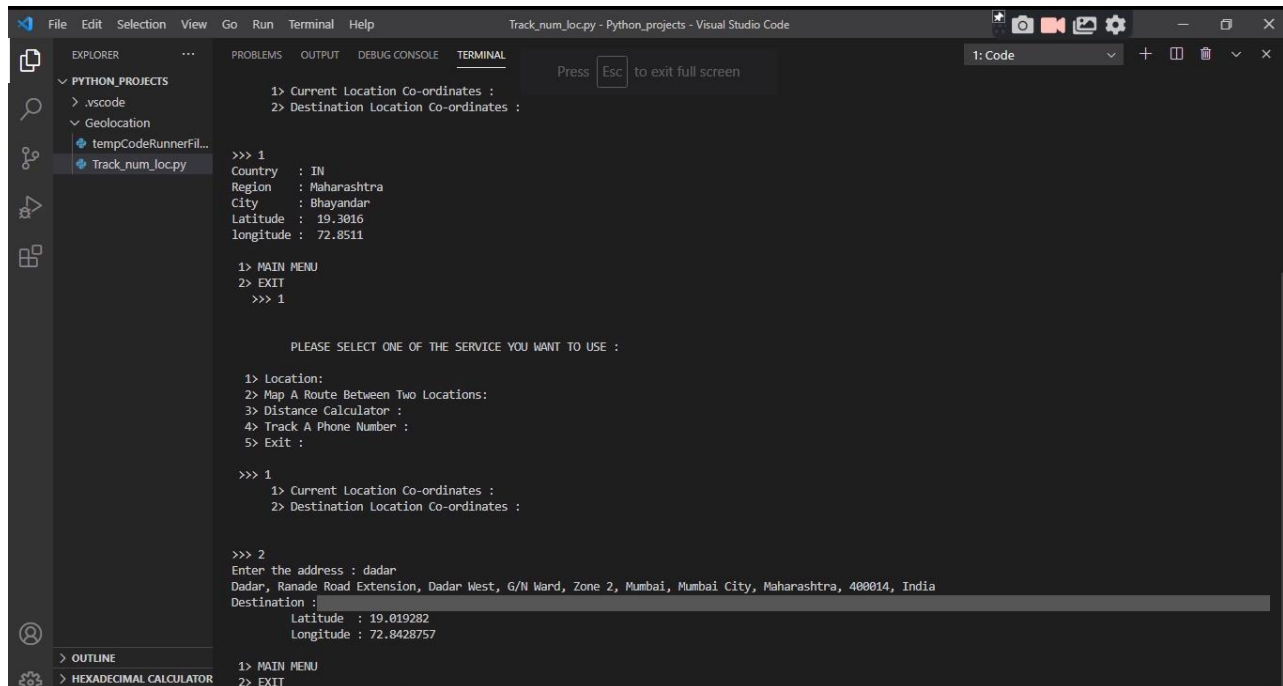
If user select current location



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the output of a Python script named `Track_num_loc.py`. The script prompts the user to select a service, and the user has selected option 1, 'Location:'. The script then prompts the user to select between 'Current Location Co-ordinates' and 'Destination Location Co-ordinates'. The user has selected 'Current Location Co-ordinates', and the script has printed the following information:

```
Country : IN
Region : Maharashtra
City : Bhayandar
Latitude : 19.3816
Longitude : 72.8511
```

If the user select destination location co-ordinates



```
File Edit Selection View Go Run Terminal Help Track_num_locpy - Python_projects - Visual Studio Code
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PYTHON_PROJECTS
> .vscode
> Geolocation
tempCodeRunnerFil...
Track_num_locpy
1> Current Location Co-ordinates :
2> Destination Location Co-ordinates :

>>> 1
Country : IN
Region : Maharashtra
City : Bhayandar
Latitude : 19.3016
Longitude : 72.8511

1> MAIN MENU
2> EXIT
>>> 1

PLEASE SELECT ONE OF THE SERVICE YOU WANT TO USE :

1> Location:
2> Map A Route Between Two Locations:
3> Distance Calculator :
4> Track A Phone Number :
5> Exit :

>>> 1
1> Current Location Co-ordinates :
2> Destination Location Co-ordinates :

>>> 2
Enter the address : dadar
Dadar, Ranade Road Extension, Dadar West, G/N Ward, Zone 2, Mumbai, Mumbai City, Maharashtra, 400014, India
Destination :
Latitude : 19.019282
Longitude : 72.8428757

1> MAIN MENU
2> EXIT
```

2) Map a route between two locations

If the user select the option 2 i.e to map a route between two location ,then system will ask from where to where i.e initial point to final point . once the user enter the initial and final point system will ask whether user want to drive or walk .

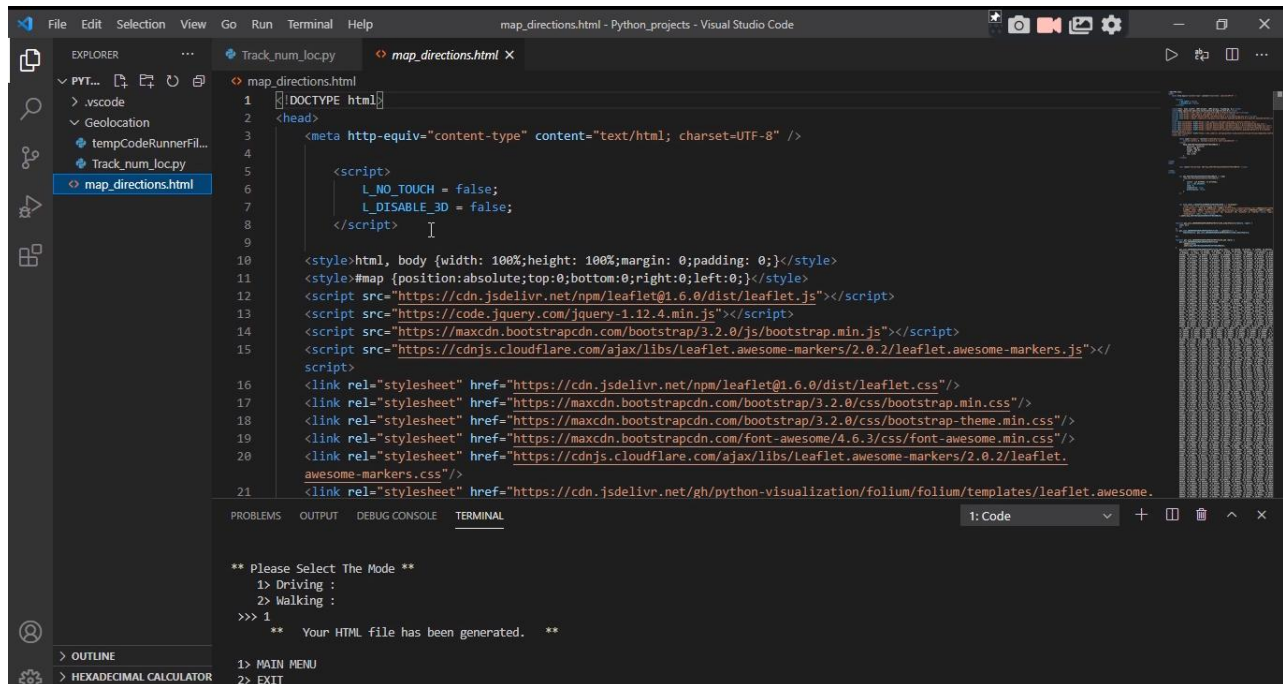
```
PLEASE SELECT ONE OF THE SERVICE YOU WANT TO USE :

1> Location:
2> Map A Route Between Two Locations:
3> Distance Calculator :
4> Track A Phone Number :
5> Exit :

>>> 2
From : Mumbai
To : Dehli
Mumbai, Mumbai Suburban, Maharashtra, India
Delhi, Kotwali Tehsil, Central Delhi, Delhi, 110006, India
19.0759899 72.8773928

** Please Select The Mode **
1> Driving :
2> Walking :
```

If the user select either the drive or walk option it will create an html file name as map_directions.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
5
6     <script>
7       L_NO_TOUCH = false;
8       L_DISABLE_3D = false;
9     </script>
10
11     <style>html, body {width: 100%;height: 100%;margin: 0;padding: 0;}</style>
12     <style>#map {position: absolute;top:0;bottom:0;right:0;left:0;}</style>
13     <script src="https://cdn.jsdelivr.net/npm/leaflet@1.6.0/dist/leaflet.js"></script>
14     <script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
15     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
16     <script src="https://cdnjs.cloudflare.com/ajax/libs/Leaflet.awesome-markers/2.0.2/leaflet.awesome-markers.js"></script>
17     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/leaflet@1.6.0/dist/leaflet.css">
18     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
19     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
20     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.6.3/css/font-awesome.min.css">
21     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/Leaflet.awesome-markers/2.0.2/leaflet.awesome-markers.css">
22     <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/python-visualization/folium/folium/templates/leaflet.awesome-
```

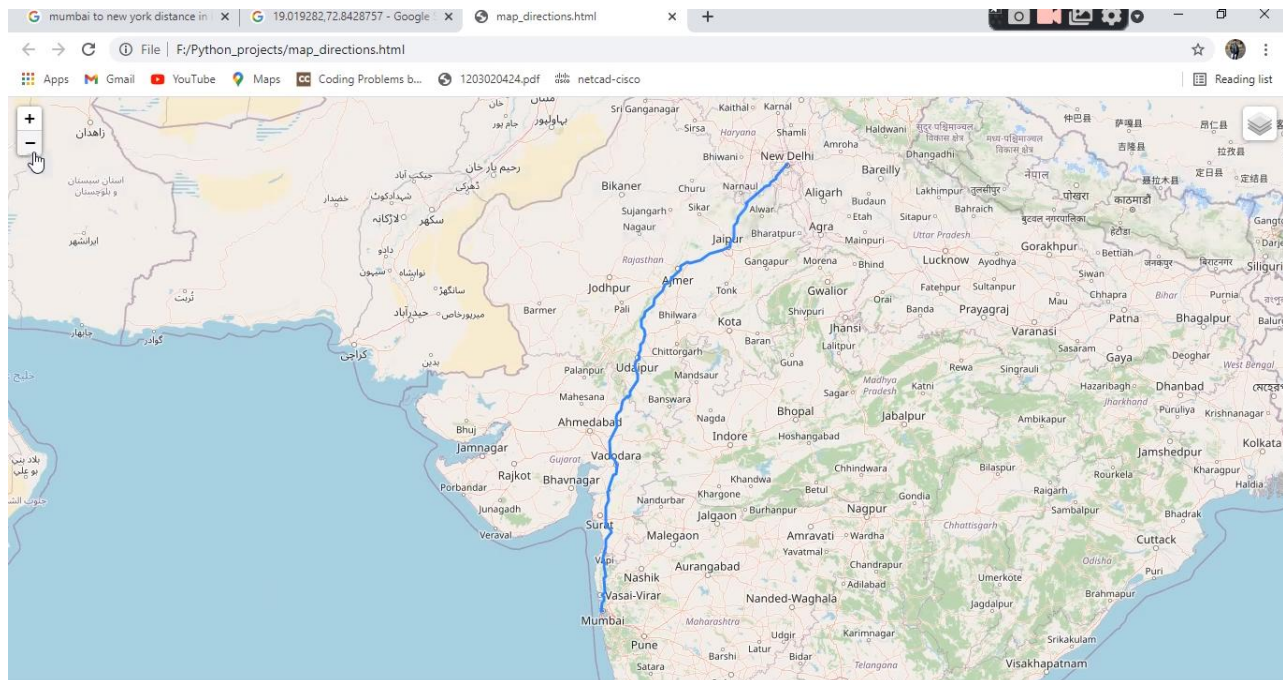
**** Please Select The Mode ****

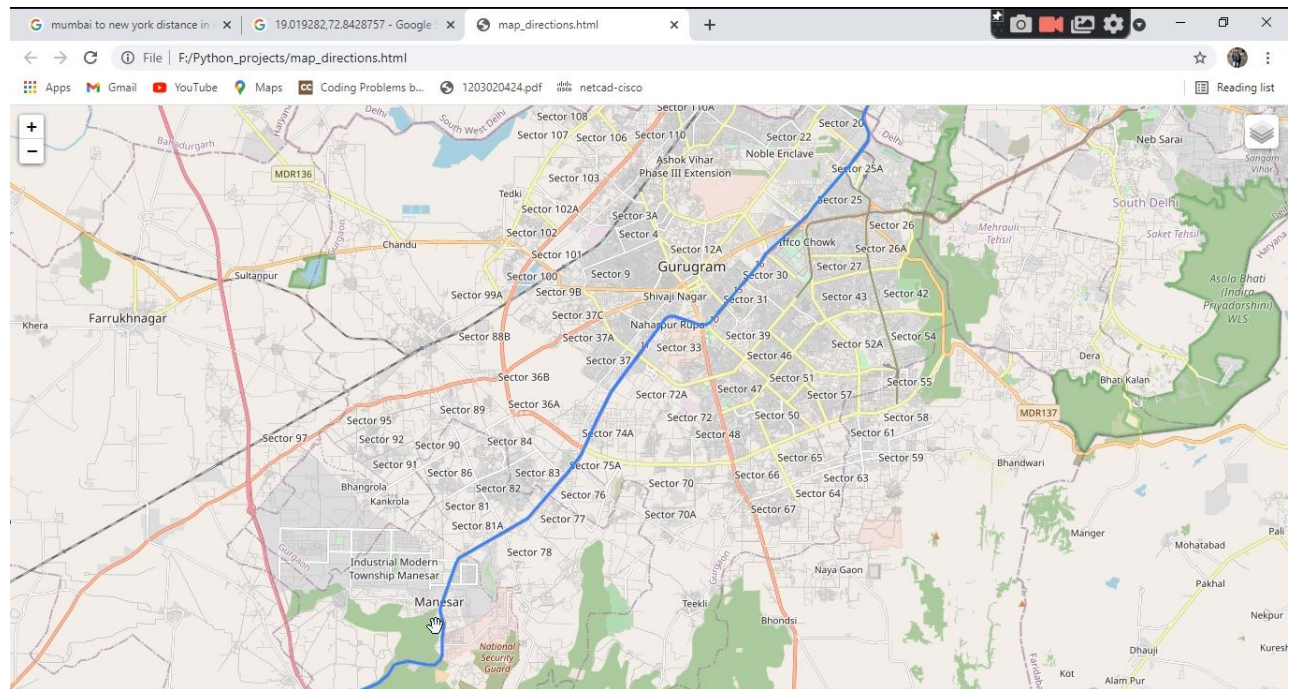
1> Driving :
2> Walking :
>>> 1

**** Your HTML file has been generated. ****

1> MAIN MENU
2> EXIT

Once the html file is created and run it will direct toward the map which will show as the route from the initial point to final point .





Distance Calculator

If the user select the option 3 i.e Distance calculator then the system will ask for the initial point and the final point . once the user provide the initial point and final point then it will calculate the distance in KM and display in the output.

```
PLEASE SELECT ONE OF THE SERVICE YOU WANT TO USE :
```

```
1> Location:
2> Map A Route Between Two Locations:
3> Distance Calculator :
4> Track A Phone Number :
5> Exit :
```

```
>>> 3
From : mumbai
To : New York
Mumbai, Mumbai Suburban, Maharashtra, India
New York, United States
12556.78469618681 Kms
```

I

Track A Phone Number

If the user select option 4 i.e Track a Phone number, the system will ask you the phone number with country code. once the user enter the number with code it will print country, state ,pin code and the sim name with latitude and longitude and it will create an HTML file.

```
PLEASE SELECT ONE OF THE SERVICE YOU WANT TO USE :
```

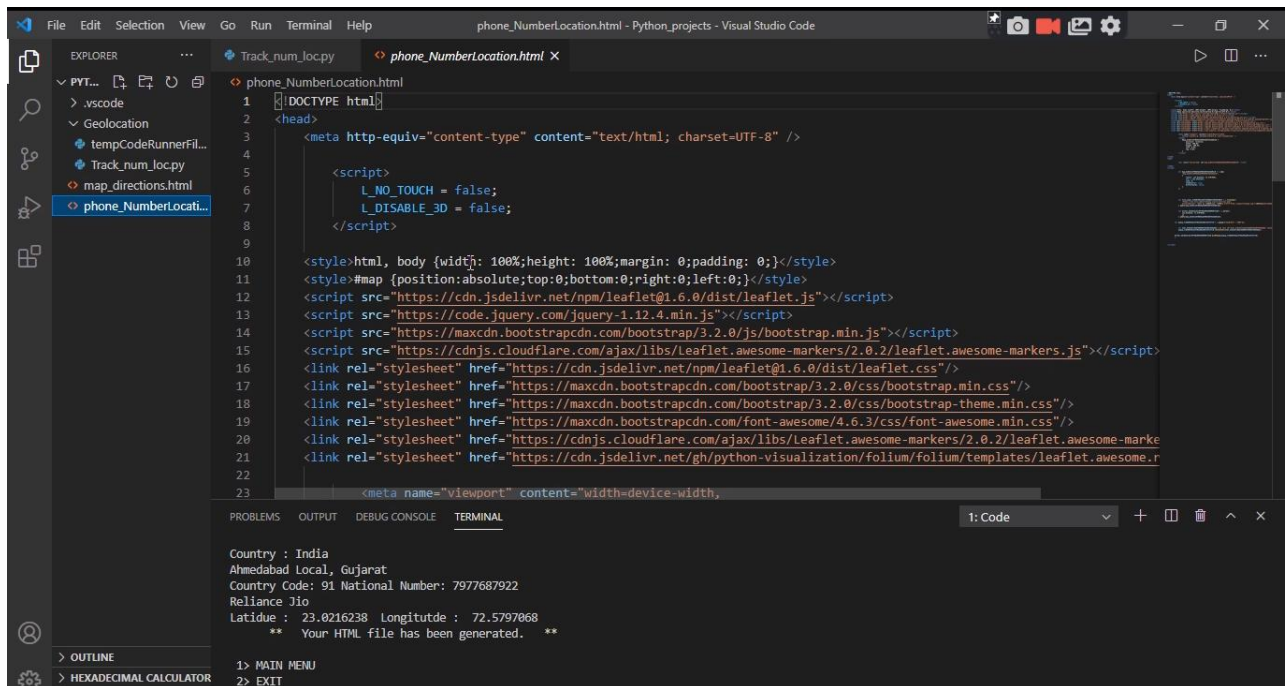
```
1> Location:
2> Map A Route Between Two Locations:
3> Distance Calculator :
4> Track A Phone Number :
5> Exit :
```

```
>>> 4
Please enter the number along with country code :
```

```
>>> +917977687922
Country : India
Ahmedabad Local, Gujarat
Country Code: 91 National Number: 7977687922
Reliance Jio
Latitude : 23.0216238 Longitude : 72.5797068
** Your HTML file has been generated. **
```

The generated HTML file “*phone_numberlocation*” provides the location of the sim card

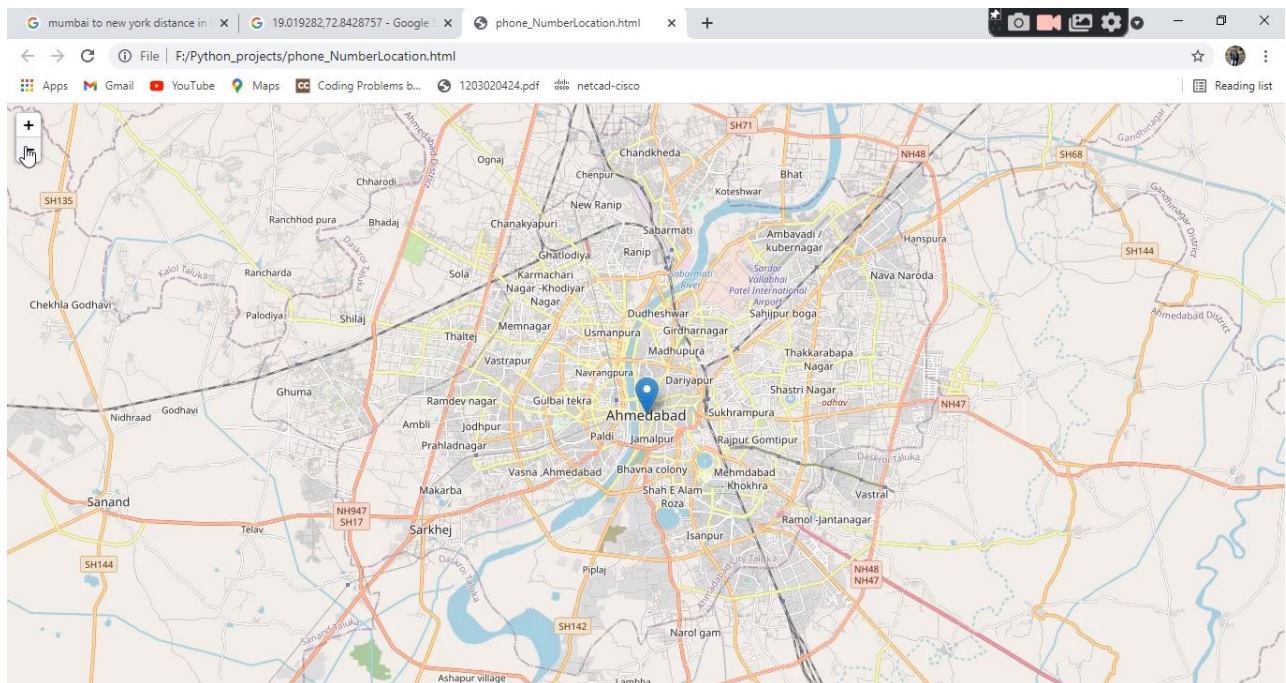
HTML file Generated



The screenshot shows the Visual Studio Code editor with the file 'phone_NumberLocation.html' open. The code is an HTML document that includes a meta tag for content type, a script for Leaflet map functionality, and a style tag for the map. The map is centered on Ahmedabad, India, with a blue pin indicating the location. The terminal output shows the following information:

```
Country : India
Ahmedabad Local, Gujarat
Country Code: 91 National Number: 7977687922
Reliance Jio
Latitude : 23.0216238 Longitude : 72.5797068
** Your HTML file has been generated. **
```

Number Location



4. Applications

It can be used to find the missing person using his

mobile number to track him down.

It can be used to direct a person to a specific location like a google map.

It can be used to find the Sim name of the provided number with its current nationality and longitude and latitude.