Introduction:

The endeavor to navigate through a sprawling college campus as a freshman can often be fraught with challenges. Recognizing this common struggle, I have undertaken the development of a sophisticated navigation project utilizing Python. This documentation serves to elucidate the intricacies of the project, delineating its components, functionality, and utility.

The solution devised entails the creation of a comprehensive navigation system leveraging the capabilities of Python. Central to this system is the integration of Folium, a powerful library for generating interactive maps, and IPython widgets, facilitating intuitive user interaction. By amalgamating these tools, users can seamlessly navigate the campus terrain with precision and ease.

Libraries Used:

- folium :- For creating interactive maps
- folium.plugins :- For using additional plugins in folium
- ipywidgets :- For creating interactive widgets in Jupyter Notebook
- json :- For handling JSON files
- os :- For interacting with the operating system

Implementaion:

- **Class Navigator :-**
  This class encapsulates functionalities related to navigation and map visualization.

  - Attributes:

    geoResources :- Dictionary containing paths to geographical resources.

    BlockLocation :- Tuple containing latitude and longitude coordinates representing the default location.

    Position :- String representing the current position.

    Destination :- String representing the destination.

  - Methods:

    init(self): Constructor method initializing attributes and populating geoResources with paths to geographical resources.

    changeDestination(self, newDestination): Method to change the destination.

    changeStartPoint(self, newStartPoint): Method to change the starting point.

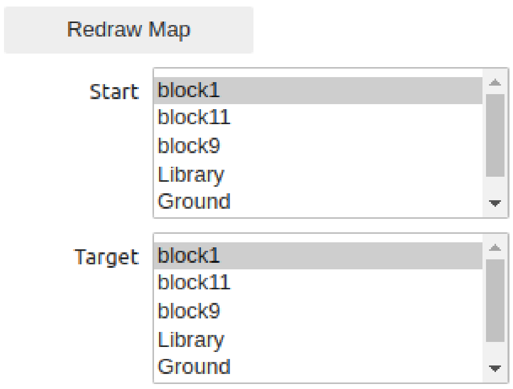
    drawPathWay(self, maps): Method to draw pathways on the map.

    drawBuilding(self, maps): Method to draw buildings on the map.

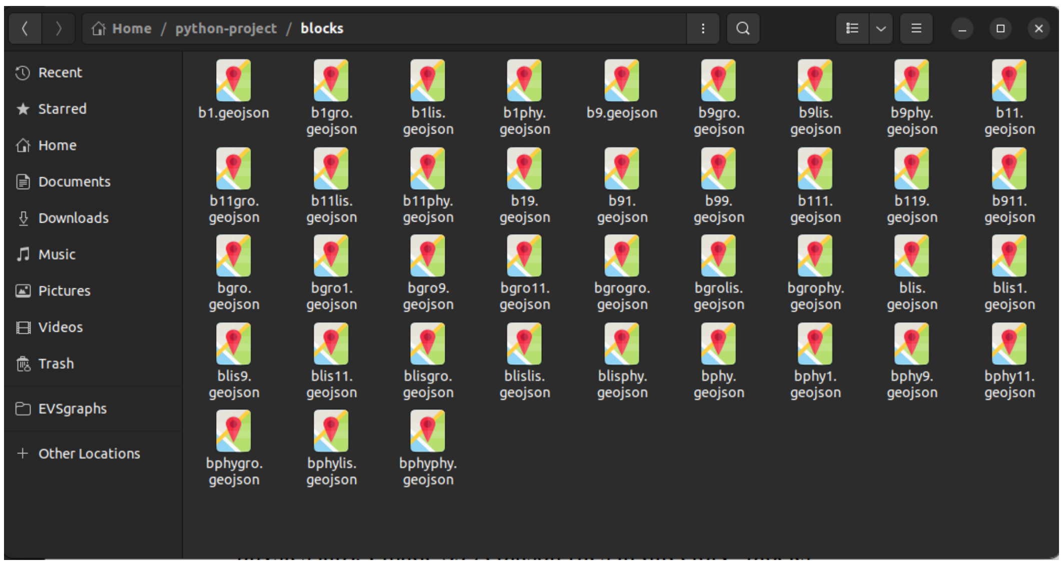
    redrawMap(self): Method to redraw the map with updated data.

- **Functions:**
  displayWay(whereTo): Function to display the selected destination.
  changePosition(whereFrom): Function to change the starting position.
  selectHouse(way): Function to handle the selection of the destination.
  selectPosition(position): Function to handle the selection of the starting position.
  on_button_clicked(b): Function to handle button clicks and redraw the map.

- **Interactive Widgets:**
  selectHouse_widget: Widget for selecting the destination.
  selectPosition_widget: Widget for selecting the starting position.
  button: Button widget for redrawing the map.
  output: Output widget to display the map.



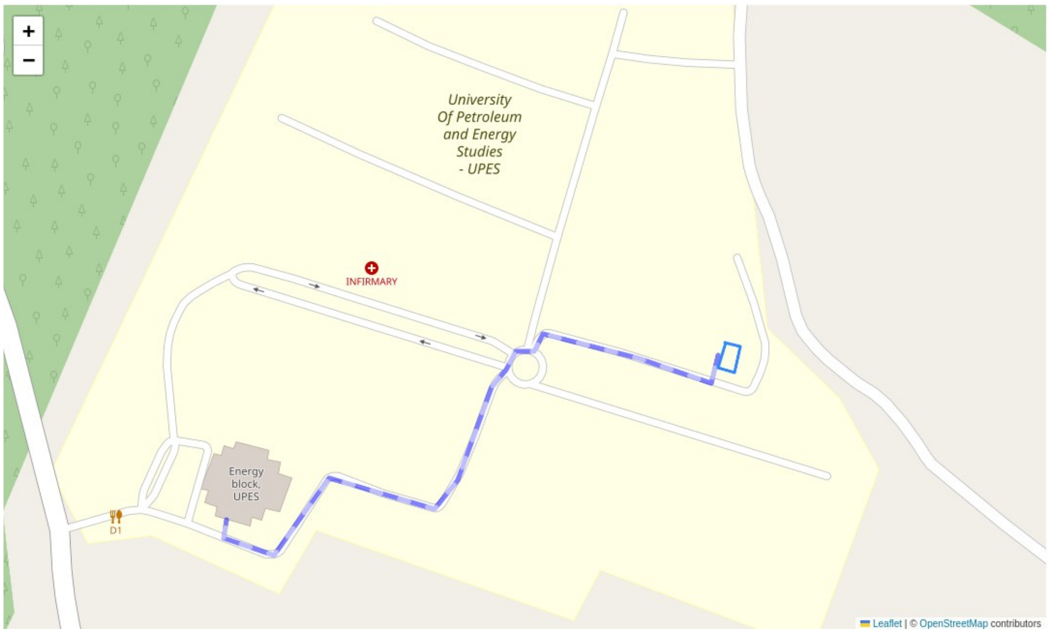
- **Resources:**
  To make this project, I had to note down the coordinates of the boundaries of each block and the coordinates of the path between any two blocks. This task was made simpler using geojson.com . Since I had decided to add 6 blocks (block 1, block 9, block 11, library, ground and physics lab), I made 39 geojson files in directory 'blocks'



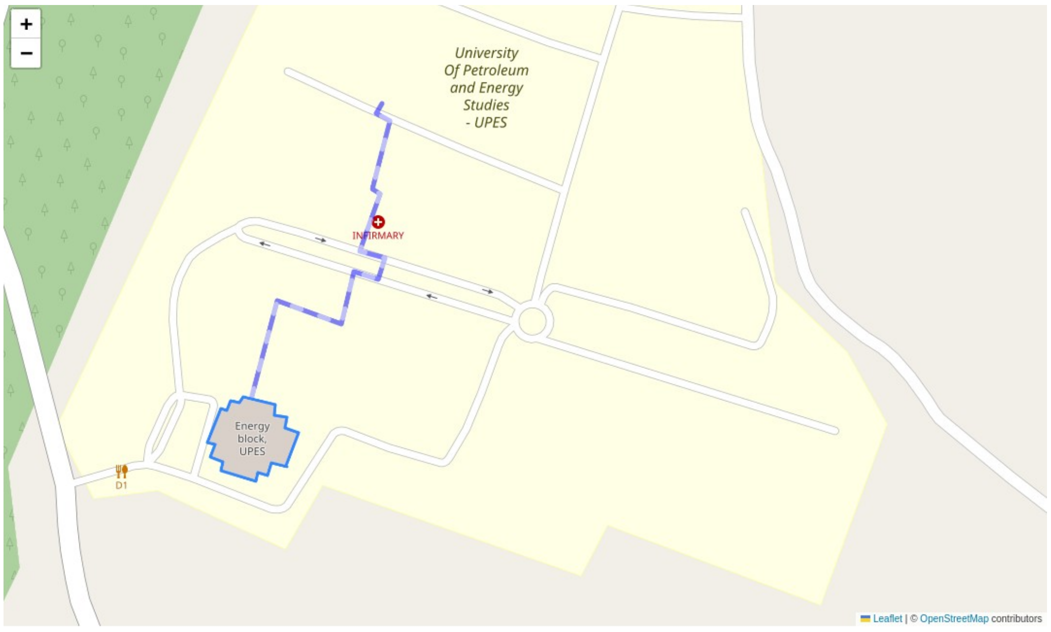
➢ **Execution :**

The usage and execution of the map is quite simple . All we have to do is to select a starting position from ‘Start’ drop down list and final position from ‘Target’ drop down list . After that just click the ‘Redraw Map’ button to render and display the map

Example 1 :- Selecting ‘Block 1’ as starting position and ‘Physics Lab’ as final position, we get the output -


+
−
University Of Petroleum and Energy Studies - UPES
INFIRMARY
Energy block, UPES
D1
Leaflet | © OpenStreetMap contributors

Example 2 :- Selecting ‘Ground’ as starting position and ‘Block 1’ as final position, we get the output -


+
−
University Of Petroleum and Energy Studies - UPES
INFIRMARY
Energy block, UPES
D1
Leaflet | © OpenStreetMap contributors

Source Code -

```
import folium
import folium.plugins
import ipywidgets
from IPython.display import display
import json
import os

coords = (30.416540, 77.968474)

class navigator:

    def __init__(self):
        self.geoResources = {}
        self.blockLocation = coords
        self.position = 'block1'
        self.destination = 'block1'

        for root, dirs, files in os.walk('blocks'):
            for file in files:
                self.geoResources[file.split('.')[0]] = os.path.join(root, file)

    def changeDestination(self, newDestination):
        self.destination = newDestination

    def changeStartPoint(self, newStartPoint):
        self.position = newStartPoint

    def drawPathWay(self, maps):
        def switchPosition(coordinate):
            coordinate[0], coordinate[1] = coordinate[1], coordinate[0]
            return coordinate

        searchString = self.position + self.destination.split('b')[1]


        with open(self.geoResources[searchString]) as f:
            testWay = json.load(f)

        finalPath = []
        for feature in testWay['features']:
            path = feature['geometry']['coordinates']
            finalPath.extend(map(switchPosition, path))

        folium.plugins.AntPath(finalPath).add_to(maps)

    def drawBuilding(self, maps):
        houseOutline = self.geoResources[self.destination]
        folium.GeoJson(houseOutline, name="geojson").add_to(maps)

    def redrawMap(self):
        maps = folium.Map(location=coords, zoom_start=17)
        self.drawPathWay(maps)
        self.drawBuilding(maps)
        display(maps)


myNavigator = navigator()

def displayWay(whereTo):
    myNavigator.changeDestination(whereTo)

def changePosition(whereFrom):
    myNavigator.changeStartPoint(whereFrom)

selectHouse_widget = ipywidgets.Select(
    options=['block1', 'block11', 'block9', 'Library', 'Ground', 'Physics Lab'],
    value='block1',
    description='Target',
    disabled=False)

def selectHouse(way):

    if way == 'block1':
        displayWay('b1')

    if way == 'block11':
        displayWay('b11')

    if way == 'block9':
        displayWay('b9')

    if way == 'Library':
        displayWay('blis')

    if way == 'Ground':
        displayWay('bgro')

    if way == 'Physics Lab':
        displayWay('bphy')

selectPosition_widget = ipywidgets.Select(
    options=['block1', 'block11', 'block9', 'Library', 'Ground', 'Physics Lab'],
    value='block1',
    description='Start',
    disabled=False)

def selectPosition(position):

    if position == 'block1':
        changePosition('b1')

    if position == 'block11':
        changePosition('b11')

    if position == 'block9':
        changePosition('b9')

    if position == 'Library':
        changePosition('blis')

    if position == 'Ground':
        changePosition('bgro')

    if position == 'Physics Lab':
        changePosition('bphy')

button = ipywidgets.Button(description="Redraw Map")
output = ipywidgets.Output()

display(button, output)

def on_button_clicked(b):
  with output:
    output.clear_output()
    myNavigator.redrawMap()

button.on_click(on_button_clicked)

ipywidgets.interact(selectPosition, position=selectPosition_widget)
ipywidgets.interact(selectHouse, way=selectHouse_widget)
```