Assignment Report:

Web Scraping & Data Collection for PC Repair/IT Support Companies in London

# Step 1: Check robots.txt

To determine if scraping is allowed, I first checked the robots.txt file of the website 'https://wiki.openstreetmap.org/robots.txt'. The content of the robots.txt file shows that while some parts of the website are disallowed for bots, there is no restriction on API interaction, which is allowed by OpenStreetMap. This confirms that we can use the Overpass API to retrieve the data we need.

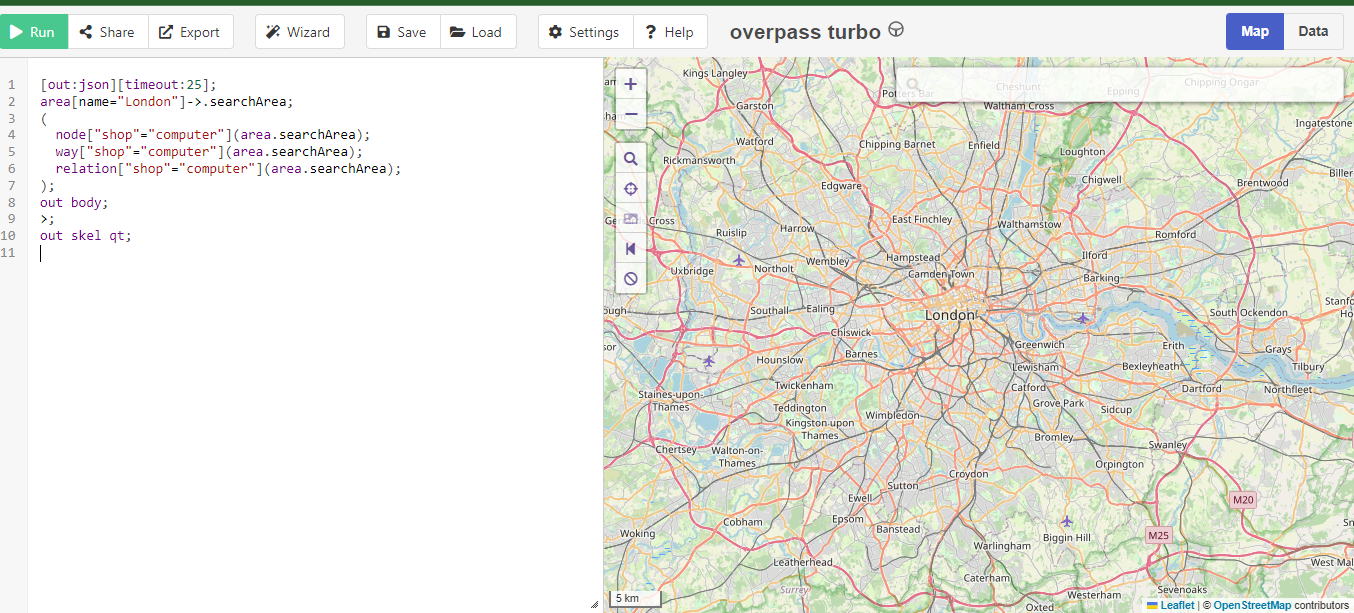


# Step 2: Construct an Overpass API Query

In this step, I used the Overpass API to retrieve data on PC Repair/IT Support companies in London. The query is designed to search for nodes, ways, and relations that are tagged with 'shop=computer' in the London area. Here is the query that I used:

[out:json][timeout:25];  
area[name="London"]->.searchArea;  
(  
 node["shop"="computer"](area.searchArea);  
 way["shop"="computer"](area.searchArea);  
 relation["shop"="computer"](area.searchArea);  
);  
out body;  
>;  
out skel qt;

This query retrieves data about computer repair shops and related businesses in the London area.



# Step 3: Send the Query to the Overpass API

Using the Python `requests` library, I sent the constructed query to the Overpass API and checked the response. If the request was successful, the API would return data in JSON format, which contains information about relevant businesses.

import requests  
response = requests.get("http://overpass-api.de/api/interpreter", params={'data': overpass\_query})  
  
if response.status\_code == 200:  
 data = response.json()  
else:  
 print("Error! Request failed.")

# Step 4: Extract Relevant Information

In this step, I parsed the JSON response from the API to extract relevant details such as the company name, website, and email address. If a field is missing in the data, I default it to 'null'.

companies = []  
for element in data['elements']:  
 if 'tags' in element:  
 name = element['tags'].get('name', 'null')  
 website = element['tags'].get('website', 'null')  
 email = element['tags'].get('contact:email', 'null')  
 companies.append([name, website, email])

# Step 5: Save the Data to a CSV File

Finally, I saved the extracted data into a CSV file using the Pandas library. The DataFrame containing the extracted company information was written into a CSV file named 'results\_p20-0480.csv'.

import pandas as pd  
df = pd.DataFrame(companies, columns=['Name', 'Website', 'Email'])  
df.to\_csv('results\_p20-0480.csv', index=False)

After running the script, a confirmation message is printed indicating the number of records stored in the CSV file.