INTRODUCTION

OBJECTIVE OF THE TASK

The objective of this report is to document the process of web scraping, data processing, and visualization undertaken as part of the **internship recruitment** task assigned by **Point Zero Solutions**. The report will detail the methods employed for scraping data from the web, the techniques used for cleaning and processing the collected data, and the visualizations created using Python and Jupyter Notebook. Additionally, the report will provide insights into the solutions implemented, and key findings derived from the analysis.

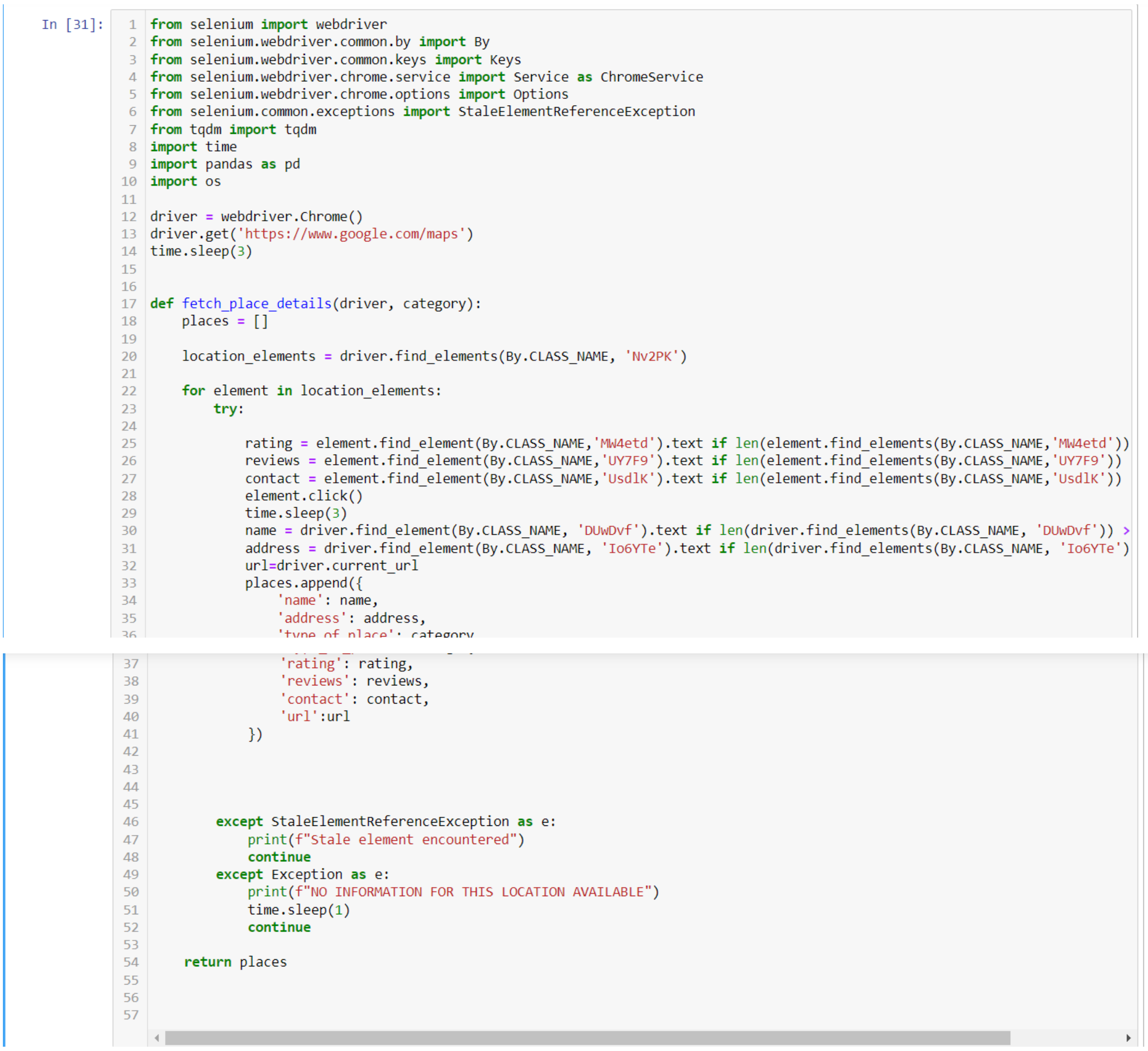
WEB SCRAPING

As the first step, I was tasked with writing a Python script to scrape data for the coordinates of various locations such as warehouses, malls, and parking lots. I was given the option to scrape data from any website that provides relevant information. I chose to scrape data from Google Maps.

Using the Selenium library in Python, I fetched the following important data:

- Name of the place
- Address
- Type of place
- Rating
- Number of reviews
- Contact number (if available)
- URL

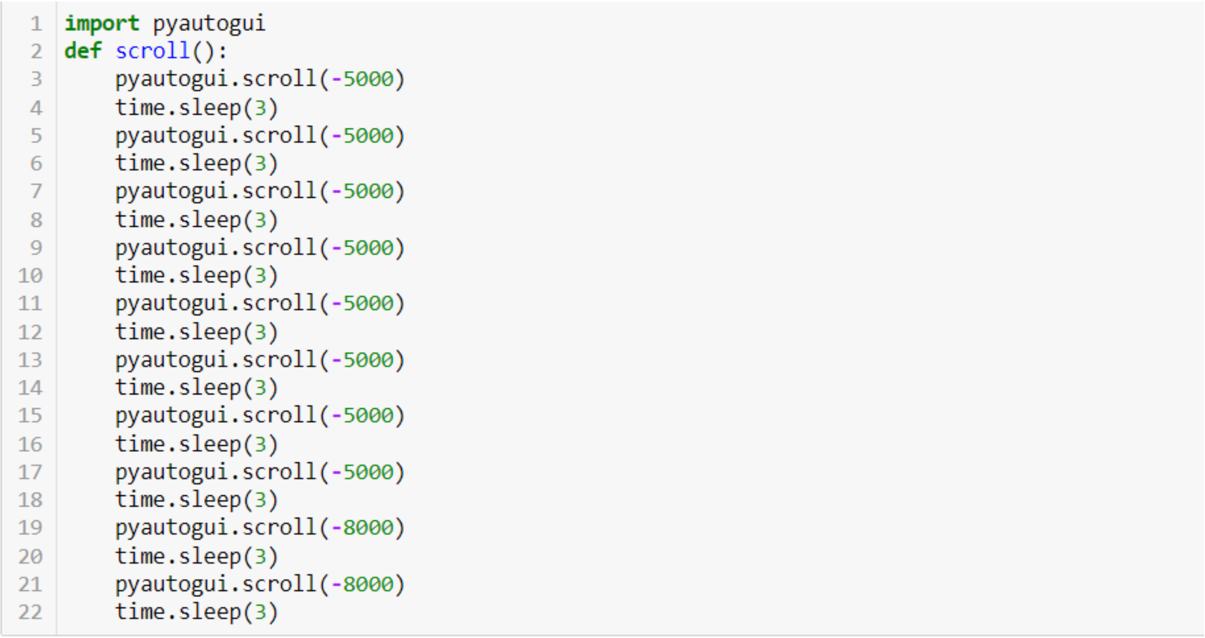
Here is a screenshot of the code I used to fetch data in Python (Jupyter Notebook):



In [31]:

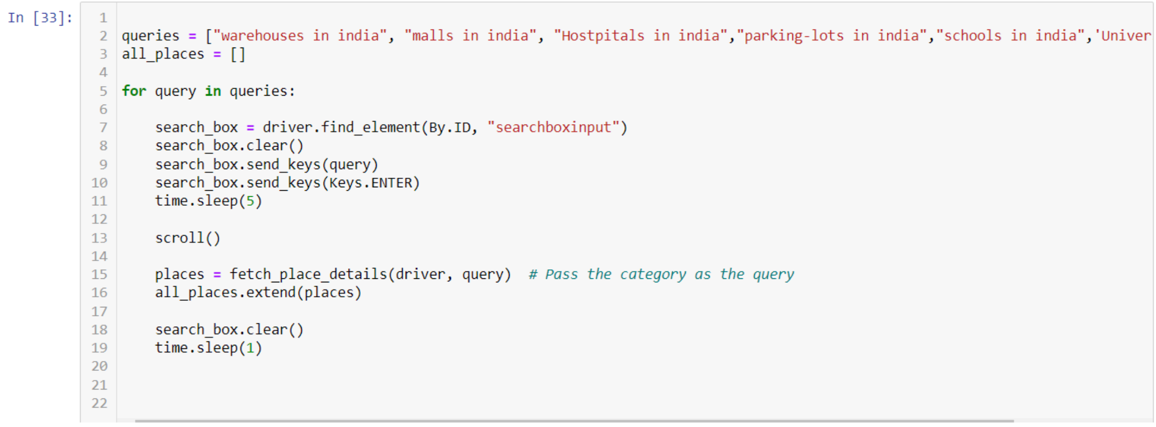
```python
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.service import Service as ChromeService
from selenium.webdriver.chrome.options import Options
from selenium.common.exceptions import StaleElementReferenceException
from tqdm import tqdm
import time
import pandas as pd
import os

driver = webdriver.Chrome()
driver.get('https://www.google.com/maps')
time.sleep(3)


def fetch_place_details(driver, category):
    places = []

    location_elements = driver.find_elements(By.CLASS_NAME, 'Nv2PK')

    for element in location_elements:
        try:

            rating = element.find_element(By.CLASS_NAME,'MW4etd').text if len(element.find_elements(By.CLASS_NAME,'MW4etd'))
            reviews = element.find_element(By.CLASS_NAME,'UY7F9').text if len(element.find_elements(By.CLASS_NAME,'UY7F9'))
            contact = element.find_element(By.CLASS_NAME,'UsdlK').text if len(element.find_elements(By.CLASS_NAME,'UsdlK'))
            element.click()
            time.sleep(3)
            name = driver.find_element(By.CLASS_NAME, 'DUwDvf').text if len(driver.find_elements(By.CLASS_NAME, 'DUwDvf')) >
            address = driver.find_element(By.CLASS_NAME, 'Io6YTe').text if len(driver.find_elements(By.CLASS_NAME, 'Io6YTe')
            url=driver.current_url
            places.append({
                'name': name,
                'address': address,
                'type_of_place': category,
                'rating': rating,
                'reviews': reviews,
                'contact': contact,
                'url':url
            })



        except StaleElementReferenceException as e:
            print(f"Stale element encountered")
            continue
        except Exception as e:
            print(f"NO INFORMATION FOR THIS LOCATION AVAILABLE")
            time.sleep(1)
            continue

    return places


```

To get more data, here is code I wrote to automate scroll in google maps interface:



```python
import pyautogui
def scroll():
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-5000)
    time.sleep(3)
    pyautogui.scroll(-8000)
    time.sleep(3)
    pyautogui.scroll(-8000)
    time.sleep(3)
```

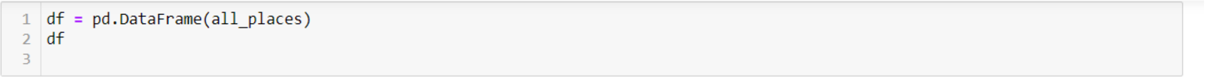
Here is code that I wrote to get all places data and store it in a list :



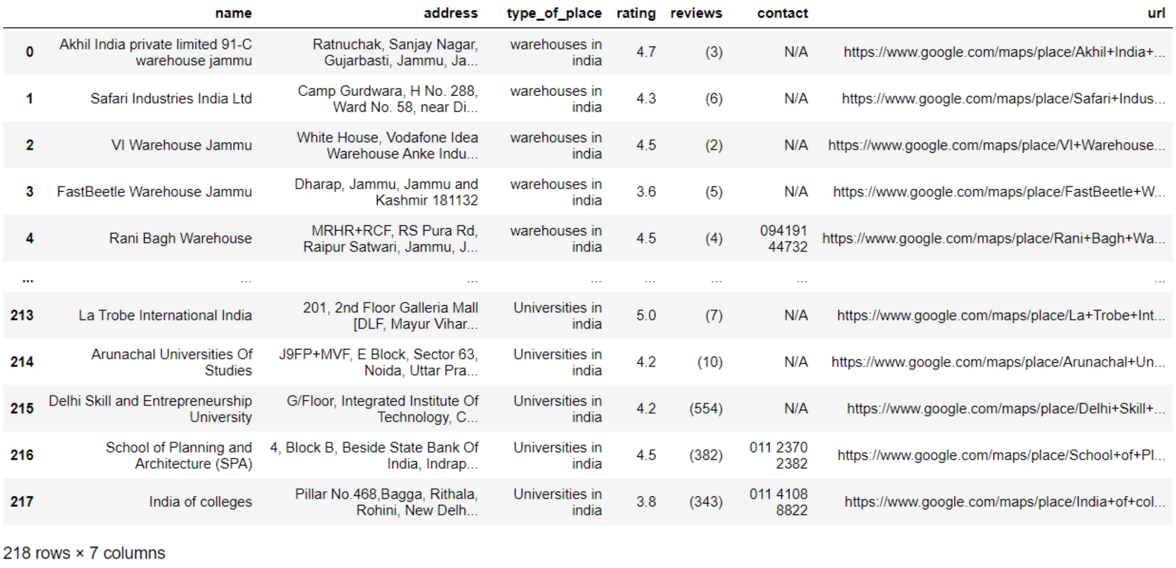
In [33]:

```

queries = ["warehouses in india", "malls in india", "Hostpitals in india","parking-lots in india","schools in india",'Univer
all_places = []

for query in queries:

    search_box = driver.find_element(By.ID, "searchboxinput")
    search_box.clear()
    search_box.send_keys(query)
    search_box.send_keys(Keys.ENTER)
    time.sleep(5)

    scroll()

    places = fetch_place_details(driver, query)  # Pass the category as the query
    all_places.extend(places)

    search_box.clear()
    time.sleep(1)



```

I saved the scraped data into a csv file with name 'map_data.csv'. This is how data looked after scraping:

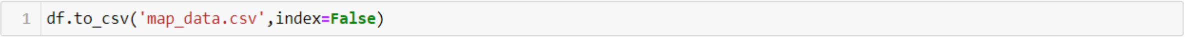


```
df = pd.DataFrame(all_places)
df

```



	name	address	type_of_place	rating	reviews	contact	url
0	Akhil India private limited 91-C warehouse jammu	Ratnuchak, Sanjay Nagar, Gujarbasti, Jammu, Ja...	warehouses in india	4.7	(3)	N/A	https://www.google.com/maps/place/Akhil+India+...
1	Safari Industries India Ltd	Camp Gurdwara, H No. 288, Ward No. 58, near Di...	warehouses in india	4.3	(6)	N/A	https://www.google.com/maps/place/Safari+Indus...
2	VI Warehouse Jammu	White House, Vodafone Idea Warehouse Anke Indu...	warehouses in india	4.5	(2)	N/A	https://www.google.com/maps/place/VI+Warehouse...
3	FastBeetle Warehouse Jammu	Dharap, Jammu, Jammu and Kashmir 181132	warehouses in india	3.6	(5)	N/A	https://www.google.com/maps/place/FastBeetle+W...
4	Rani Bagh Warehouse	MRHR+RCF, RS Pura Rd, Raipur Satwari, Jammu, J...	warehouses in india	4.5	(4)	094191 44732	https://www.google.com/maps/place/Rani+Bagh+Wa...
...	...	...	...	...	...	...	...
213	La Trobe International India	201, 2nd Floor Galleria Mall [DLF, Mayur Vihar...	Universities in india	5.0	(7)	N/A	https://www.google.com/maps/place/La+Trobe+Int...
214	Arunachal Universities Of Studies	J9FP+MVF, E Block, Sector 63, Noida, Uttar Pra...	Universities in india	4.2	(10)	N/A	https://www.google.com/maps/place/Arunachal+Un...
215	Delhi Skill and Entrepreneurship University	G/Floor, Integrated Institute Of Technology, C...	Universities in india	4.2	(554)	N/A	https://www.google.com/maps/place/Delhi+Skill+...
216	School of Planning and Architecture (SPA)	4, Block B, Beside State Bank Of India, Indrap...	Universities in india	4.5	(382)	011 2370 2382	https://www.google.com/maps/place/School+of+Pl...
217	India of colleges	Pillar No.468,Bagga, Rithala, Rohini, New Delh...	Universities in india	3.8	(343)	011 4108 8822	https://www.google.com/maps/place/India+of+col...

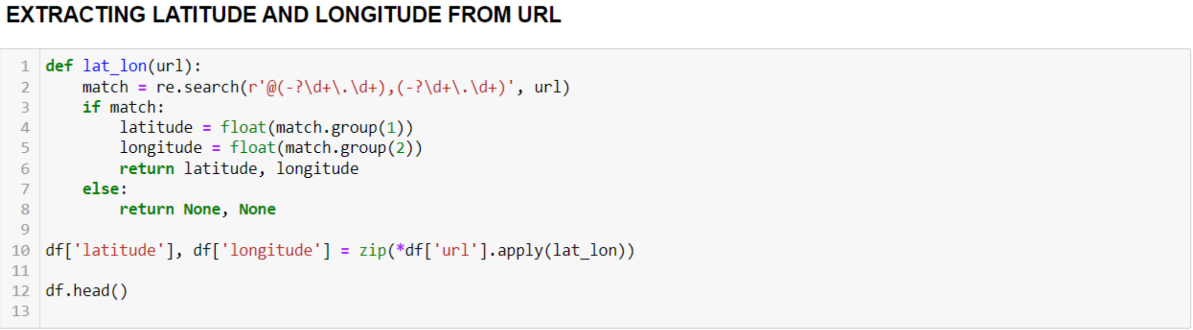
218 rows × 7 columns



```
df.to_csv('map_data.csv',index=False)
```

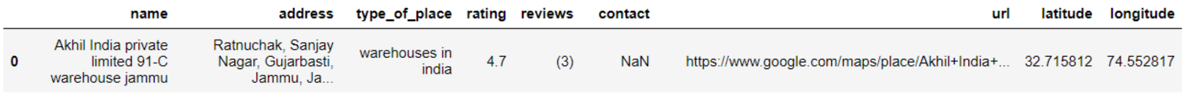
DATA CLEANING & DATA PREPROCESSING

After saving the scraped data in a CSV format, I used Pandas to import the data into a new Jupyter Notebook for further processing. The first step was to extract the latitude and longitude coordinates from the URL column values in the dataset.Here is the code:



EXTRACTING LATITUDE AND LONGITUDE FROM URL

```
def lat_lon(url):
    match = re.search(r'@(-?\d+\.\d+),(-?\d+\.\d+)', url)
    if match:
        latitude = float(match.group(1))
        longitude = float(match.group(2))
        return latitude, longitude
    else:
        return None, None

df['latitude'], df['longitude'] = zip(*df['url'].apply(lat_lon))

df.head()

```



	name	address	type_of_place	rating	reviews	contact	url	latitude	longitude
0	Akhil India private limited 91-C warehouse jammu	Ratnuchak, Sanjay Nagar, Gujarbasti, Jammu, Ja...	warehouses in india	4.7	(3)	NaN	https://www.google.com/maps/place/Akhil+India+...	32.715812	74.552817

Next, I cleaned the data by handling the null values. Specifically, I had three columns with null values: **rating** (5 missing), **reviews** (5 missing), and **contact** (66 missing). Here is how I addressed these null values:

- For the **rating** column, I replaced the null values using the **statistics.mean** function to calculate the average rating from the available data.
- For the **reviews** column, I replaced the null values with **0**, assuming no reviews were left for those entries.
- For the **contact** column, which had a significant number of null values (66), I decided to delete this column as it was not critical for the analysis.

After cleaning the data, I performed additional data preprocessing. Specifically, I noticed that the **ratings** column had values in the format "(2)", "(32)", "(12)", which needed to be converted to float values for further analysis. I stripped the parentheses and converted these values to a standard numerical format like 2, 32, and 12.

DATA ANALYTICS AND VISUALIZATIONS

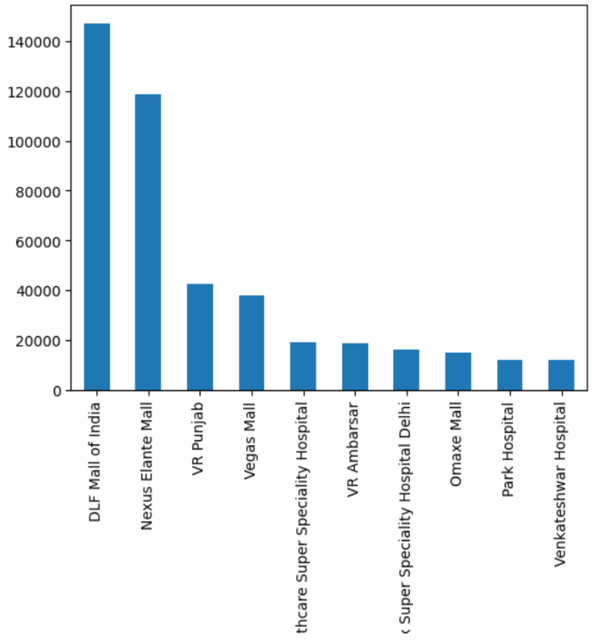
Here are the insights that I got from the data:

1.Average Ratings:

- Average rating for warehouses is: 4.01
- Average rating for malls is: 4.16
- Average rating for hospitals is: 4.19
- Average rating for parking-lots is: 3.95
- Average rating for schools is: 4.22
- Average rating for universities is: 4.27

2. Places with most number of reviews:

```
name                                           type_of_place
DLF Mall of India                              malls               147255.0
Nexus Elante Mall                              malls               118809.0
VR Punjab                                      malls                42750.0
Vegas Mall                                     malls                37797.0
Aakash Healthcare Super Speciality Hospital    hospitals            19065.0
VR Ambarsar                                    malls                18793.0
BLK-Max Super Speciality Hospital Delhi        hospitals            16407.0
Omaxe Mall                                     malls                14898.0
Park Hospital                                  hospitals            12126.0
Venkateshwar Hospital                          hospitals            12124.0
```



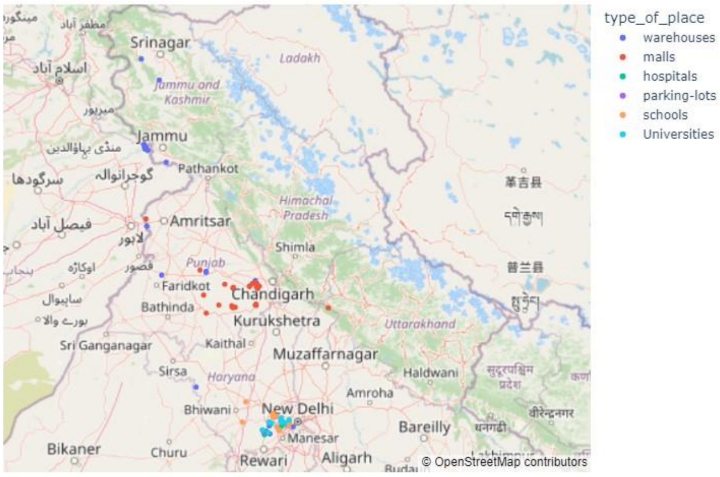
3. Places with highest ratings:

name	type_of_place	rating
Benevolence logistics	warehouses	5.0
Best Cancer Hospital in India	hospitals	5.0
Best Spine Surgery Hospital in India	hospitals	5.0
Central warehousing corporation	warehouses	5.0
CollegeSakha - Top Colleges, Universities & Institutes in india	Universities	5.0
La Trobe International India	Universities	5.0
My Care India | Medical Tourism | Health Tourism in India	hospitals	5.0
Pearl omax parking lot	parking-lots	5.0
RED Ambulances - RED.Health Gurgaon	hospitals	5.0
ServeXplus India Central Warehouse	warehouses	5.0

4. Total Count of places in the dataset:

hospitals	42
Universities	41
schools	39
malls	38
warehouses	33
parking-lots	25

5. A map showing the locations of the places with different markers for warehouses, malls, parking lots etc. :



CHALLENGES

The major challenge I faced during this project was during the web scraping phase. Here are the key difficulties and how I overcame them:

1. **Scrolling with Selenium WebDriver:**
   - **Issue:** Initially, I struggled with implementing the scrolling functionality using Selenium WebDriver. The usual methods to scroll the page were not working as expected.
   - **Solution:** After several attempts, I decided to use the **pyautogui** library to create an automated scroller. This library allowed me to simulate keyboard and mouse actions, which helped in scrolling the webpage effectively.

2. **Fetching Latitude and Longitude:**
   - **Issue:** Google Maps does not provide latitude and longitude coordinates directly in the user interface, making it impossible to fetch them from the HTML code.
   - **Solution:** After some investigation, I noticed that the coordinates were embedded in the URL of each location. I decided to extract the latitude and longitude values from these URLs.

By addressing these challenges with innovative solutions, I was able to successfully scrape and process the required data for further analysis and visualization.

Check my whole project here:

https://github.com/Rizwal/Google_Map_Scraper-Analysis