**Name: Echefu Charles**

**Web Scrapping Workflow**

**Website Review**

Before I started writing any python code to scrape the website provided, I visited the website in order to have a feel of the interface as well as confirm availability of the information required in the result set.

After doing this, I made an attempt to get the html script of the website using a python library called BeautifulSoup. However, based on the result, I figured that the website was a dynamic website and therefore can not be scrapped with BeautifulSoup. I eventually used the python library called Selenium to get the html of the website.

**Hyperlink Fetching and Information Scraping**

I observed that each webpage contained a list of 20 organizations. Therefore, I decided to first get the hyperlinks for all organizations in each webpage and save the links in a text file. I did this so I could keep track of the URLs that have been scrapped so that I know where to reinitiate the scrape if the selenium driver should time out. I imported the required libraries for the web scraping (selenium, pandas and time). Having obtained my list of hyperlinks, I iterated through list in order to load the URL for each link and pick the required information from each page and save them in a python list.

**Saving Information**

Finally, I used the pandas library to convert the list of information I had scrapped to a pandas dataframe which I then saved as a csv file.

**Challenges and Steps Taken**

|  |  |  |
| --- | --- | --- |
| S/N | Challenge | Step Taken |
| 1 | The website is dynamic. Hence, I had to load each URL one after the other in order to scrape information specific to the URL. | I used the selenium library in place of the BeautifulSoup library |
| 2 | Bypassing reCAPTCHA on the website | I used a for loop to iterate through the links obtained and increased sleep time of the driver per url load. |
| 3 | Selenium Driver Timing out | I added a counter to the script so that I kept track of URLs that had been scrapped and reinitiate the driver appropriately. |
| 4 | Inconsistent HTML Structure across URLs | I tried as much as possible to create methods that were suitable for the dynamic structures. |

Some other limitations include:

* Internet speed
* Duration for completion

**Tools Used**

The tools used for the assessment include:

* Python
* Selenium Library
* Pandas Library
* Chrome Browser/Driver

**Note:**

Due to the time constraint of the work, and due to the nature of the website, I was unable to scrape all the available URLs. The csv file sent contains the data I was able to scrape. However, the python script for my work is available for your perusal should the need arise.