SCRAPPING FOR INFORMATION ON HOMES FOR SALE IN VIRGINIA BEACH, VA (UNITED STATES OF AMERICA) FROM THE WEBSITE 'HOMES.COM'

Name of Participants:

Mark Asamoah – 428905
Cynara Nyahoda – 435030

Description of Topic:

The main purpose of this project as stated in the topic above, is to gather information on properties that are listed for sale on homes.com. In this project, we limited our scrapping to the sub domain https://www.homes.com/virginia-beach-va/homes-for-sale to scrape for information only for properties in Virginia Beach (VA).

Homes.com is a popular website that allows individuals or real estate agencies in the United States list their properties, mostly houses and lands for sale, rent or other mortgages.

Description of Scrapper Mechanisms:

- **Beautiful Soup:** The beautiful soup scrapper uses the **requests.get** library to receive information from the defined url and then passes it on to the BS scrapper in the function **BeautifulSoup** as the **'html'** and uses **'lxml'** as a parser. The **soup.find_all** function is then used to find the elements which contains the listings on the website, in our case a **'tag_name'** called **'article'** with a class **"f1vrb3k6 radius-5 box-shadow-cards relative overflow-hidden c1or1u3b".** Then the **article.find** (from our tag name) function is applied on the tag to find elements we want to scrape for using their various **'divs', 'class', 'id',** etc**.**
  The scrapped results are saved as described in Description of Output below.

- **Scrapy:** The scrapy spider with name **'homespider'** is created under a **class**, in our case **HomespiderSpider**. We defined the urls (allowed and starting url) to scrap under this class. Using the self and response as a parse, the homespider scrap for information on the **"start_url"** using **'response.XPATH'** method. In the fisrt case, it scraps for elements with XPATH, **"//article"** which contains all the listings on the website and saves it as **"all_houses"**.
  For each element found in this XPATH (//article), we find the sub elements, inheriting the XPATH from "//article". In our case, it was defined as **"each_house.XPATH"**. All information gathered from this sub elements are stored as described in Description of Output below.

- **Selenium:** The selenium scraper uses the chrome web driver to automatically control the browser and enters in to the website to scrap for the required information. In our case, the **WebDriver** was defined as **"driver"**. So the **"driver.get"** command is used to enter the website in the browser. Since our aim is to get the information from listings in Virginia beach, the scrapper locates the search bar on the items using the XPATH method and enter the text **"Virginia Beach"**

into it and hits enter using the **"send_keys(key.RETURN)"**. Using the **WebDriverWait** and **ExpectedCondition,** the scrapers waits for presence of the element with "CLASS_NAME"; **"c12kybvb"** then stores the element as **"class_item"**. From the class_item using "**find_element.XPATH"** we scrap for sub elements with the **tag name "article"** and stores the elements as **"ads"**. From the items in the ads the WebDriver locates the up arrow in the article and clicks on it to reveal all items to be scrapped. Now with **"ads.find_element_by_XPATH"** the information required is being scrapped and described in the Description of Output below.

Description of Output:

The information gathered by our scrappers are:

- **Property Type:** This describes the type of property type whether is a ***house, land or a condominium.***
- **Price:** This is the price of the property listed for sale on the website.
- **Number of Beds:** This indicates the number of bed rooms in the house if the property listed is a house or a condominium.
- **Number of Baths:** This indicates the number of bath rooms in the house if the property listed is a house or a condominium.
- **Size in Sqft:** This indicates the land size of the property.
- **Address:** This is the information that shows the exactly location in Virginia Beach where the property can be found.
- **Link:** This information gives the web address of the property to view other specific information.

Elementary Data Analysis:

The data collected by our scrappers can be used by individuals or companies to determine vital information like:

- The average price of a house/condominium/land in this state
- Average sizes of the properties in this area
- Average number of bathrooms and bedrooms found in the properties.
- Comparison of price against property attributes for a more informed decision

These statistical analysis can be performed on the data using programs like R or by using excel. Since the data was saved in a csv file, we used Excel functions to calculate some of the mentioned averages. Using a random sample of 230 listed properties in Virginia Beach, we calculated an average price of one property to be $398,549.58 which will likely consist of 3 bedrooms and 2.5 bathrooms on average, with an average size of 1925sqft. The lowest priced property from a sample of 1582 properties is a 2.431sqft house with 3 bedrooms and 3 bathrooms that is listed at $2310. One can also find from analysis that the most expensive property is priced at $11,800,000 and has 7 bedrooms and 8 bathrooms, with a size of 10,835sqft. From this simple snapshot of the data, we can instantly determine that the bigger the size of the house, the more expensive it will be to purchase.

Participants Work:

We both contributed immensely towards this work. In our first discussion towards this project, we agreed to share the work among ourselves and later work on the last parts together. We both wrote one full program each but collaborated with each other and helped to figure out a solution to a particular error one runs into. The last one was written during our scheduled days for meeting and we gave ourselves time to figure out a solution individually and later met again to discuss. I, Mark, prepared this document whilst Cynara wrote the final comments on the codes. We later read through each other's work to make sure it was as required.