Logo

Description automatically generated

**College of Professional Studies**

**Northeastern University San Jose**

**MPS Analytics**

**Course: ALY6140 – Analytics Systems Technology**

**Assignment:**

Final Project Draft

**Submitted on:**

March 27th, 2024

**Submitted to:**  **Submitted by:**

Professor: BEHZAD AHMADI ARCHIT BARUA

BHAGYASHRI KADAM

NIKSHITA RANGANATHAN

**Introduction**

In the vast and ever-expanding digital universe, the internet is a treasure trove of data, sprawling across myriad websites, each harboring valuable insights waiting to be unearthed. Our group project for the ALY 6140 class ventures into the intricate realm of web scraping, an indispensable technique in data science that enables the systematic extraction and analysis of web data. This collaborative endeavor goes beyond mere data retrieval; it's an exploration into distilling actionable insights from the complex structures of HTML and JavaScript-laden pages.

The project initiates with an introductory foray into web scraping, leveraging the BeautifulSoup library as our primary tool. BeautifulSoup stands out for its proficiency in parsing HTML and XML documents, providing us with the capability to traverse the parse tree with ease and extract pertinent information, setting the stage for our data extraction journey.

Advancing further, our collective focus shifts towards Wikipedia, particularly the "The World's Billionaires" list. This segment of our project is dedicated to employing web scraping techniques to collate and scrutinize data concerning the global distribution of wealth among billionaires. The insights derived from this analysis not only serve an academic purpose but also illuminate the trends and dynamics in global economic power distribution.

The project then transitions to the realm of dynamic web scraping through Selenium, a tool that emulates human interaction with web browsers. This phase centers on Amazon, one of the behemoths of the e-commerce world, illustrating the application of web scraping on a platform teeming with dynamic content. Selenium's ability to mimic human actions—such as clicking, navigating, and form submission—enables us to access data that transcends the limitations of static HTML, offering a deeper dive into the intricacies of dynamic web content.

Together, BeautifulSoup and Selenium underscore the multifaceted nature of web scraping, from handling straightforward static pages to navigating the complexities of dynamic, script-intensive websites. Through this project, we endeavor to showcase the robust capabilities and diverse applications of web scraping techniques, providing a panoramic view of their potential to transform data into insightful and actionable knowledge.

**Approach**

Our project employs the Python programming language, utilizing libraries like BeautifulSoup for parsing HTML and XML, and Selenium for automating web browser interactions. These tools allow us to collect data from various web sources systematically.

We start by scraping a practice website designed for learning scraping techniques. Using the requests library, we access the content of the webpage and then employ BeautifulSoup to parse this content. This enables us to navigate through the HTML structure of the page, extract data, and print it in a readable format.

The next part of our project focuses on Wikipedia's "The World's Billionaires" page. We target this specific page to extract structured data about billionaires, such as their net worth and rankings. We navigate through the HTML tables using BeautifulSoup's functions to locate and process the relevant information, eventually saving the data into a structured CSV file using the pandas library.

Lastly, we explore dynamic web scraping with Selenium on Amazon's website. This part of the project demonstrates how to handle JavaScript-heavy websites where content is loaded dynamically. Selenium's WebDriver allows us to interact with the webpage, performing searches and capturing the resultant data. We then use BeautifulSoup to parse the data obtained by Selenium, extract product details like names, prices, and ratings, and save this data into a CSV file, much like we did with the Wikipedia data.

The techniques demonstrated in this project highlight the powerful combination of BeautifulSoup for static content and Selenium for dynamic content, providing a versatile approach to web scraping.

**Conclusion**

Throughout this project, our team has successfully navigated the intricacies of web scraping, uncovering the potential of both BeautifulSoup and Selenium for gathering data from the web. We've showcased the flexibility of Python for such tasks, blending technical acuity with analytical prowess to extract meaningful insights from the data-rich landscape of the internet.

From static web pages like Wikipedia to dynamic platforms like Amazon, our methods have proven effective in harvesting data that is not only comprehensive but also ready for further analysis. Our exploratory journey through the billionaires' data has given us a snapshot of global financial standings, while scraping Amazon has provided a window into the competitive world of e-commerce.

The proficiency in employing these tools not only enhances our skill set but also opens up avenues for potential research and business intelligence applications. As we conclude this project, we reflect on the importance of ethical considerations, reminding ourselves to scrape responsibly and respect data privacy. Our findings underscore the transformative power of data when leveraged with the right tools, promising exciting prospects for future endeavors in data science.