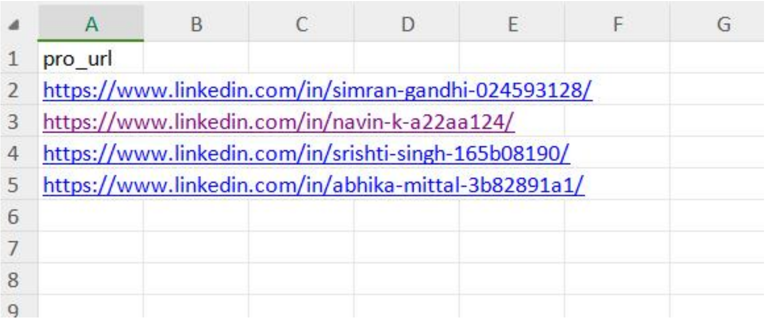
Task Report

Task: scraping the linked in profle
Prerequisites: python ide, selenium, beautifulsoup, excel sheet containing links of profiles

Employee.xlsx



	A	B	C	D	E	F	G
1	pro_url						
2	https://www.linkedin.com/in/simran-gandhi-024593128/						
3	https://www.linkedin.com/in/navin-k-a22aa124/						
4	https://www.linkedin.com/in/srishti-singh-165b08190/						
5	https://www.linkedin.com/in/abhika-mittal-3b82891a1/						
6							
7							
8							
9							

About:

It is a python code, that automates the process of extracting information from LinkedIn profiles using web scraping techniques and the Selenium WebDriver. The project allows users to provide a LinkedIn profile URL and then gathers valuable data from the profile, including the individual's name, current company, location, connection count, about section, and work experience details.

Selenium: Selenium is a web automation framework that allows developers to control web browsers programmatically. It's particularly useful for tasks that involve interactions with dynamic web content, such as clicking buttons, filling forms, and navigating through pages. Selenium is commonly employed for tasks like automated testing, web scraping, and data extraction from websites that heavily rely on JavaScript.

Beautiful soup: Beautiful Soup is a Python library used for web scraping. It provides tools to parse HTML and XML documents, allowing developers to extract specific elements, attributes, and text content from web pages. Beautiful Soup is often used to navigate and manipulate HTML and XML structures, making it an essential tool for extracting structured data from websites and generating useful insights.

Steps:

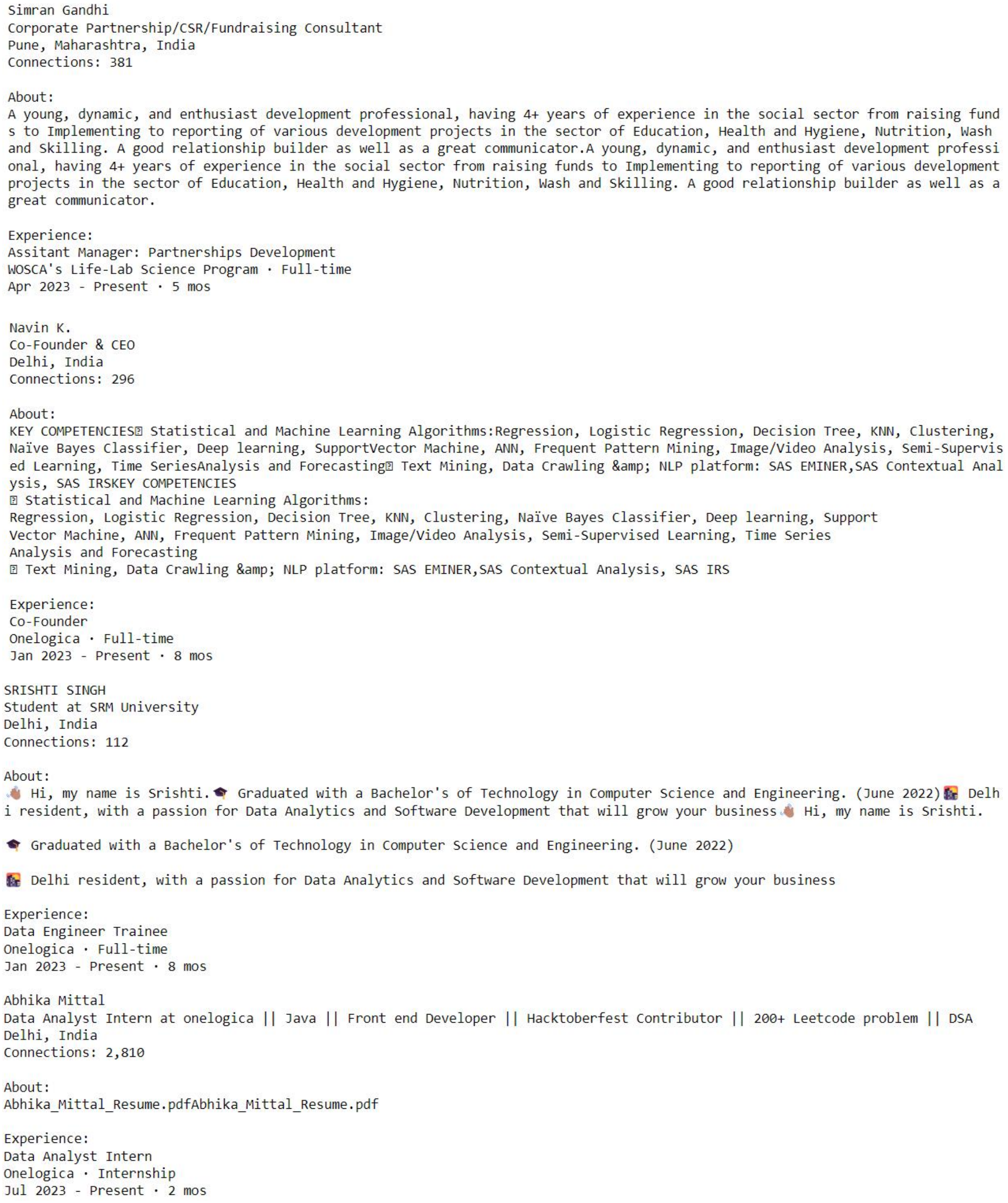
1) firstly install selenium and beautiful soup
- Go to command prompt>run as administrator
- Type: “pip install selenium”, “pip install beautifulsoup4”

2) now open your python IDE and type the following code:

```
import requests,time,random
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By

#creating an object for webdriver
driver=webdriver.Chrome()
driver.get("https://linkedin.com/uas/login")
time.sleep(5)

# entering username
username = driver.find_element(By.ID, "username")

# Enter Your Email Address
username.send_keys("your_email")

# entering password
pword = driver.find_element(By.ID, "password")

# Enter Your Password
```

```
pword.send_keys("your_password")

# Clicking on the log in button
driver.find_element(By.XPATH, "//button[@type='submit']").click()
time.sleep(20)

for url in urls:
    #opening the profile using driver
    driver.get(url)
    # scrolling down to the bottom of the page
    SCROLL_PAUSE_TIME=5
    last_height=driver.execute_script("return document.body.scrollHeight")

    for i in range(3):
        driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
        time.sleep(SCROLL_PAUSE_TIME)
        new_height=driver.execute_script("return document.body.scrollHeight")
        if new_height==last_height:
            break
        last_height=new_height

    #using beautiful soup to extract the html tags
    src=driver.page_source
    soup=BeautifulSoup(src,'lxml')

    #extracting the html tags to find the name and place of work
    name_div=soup.find('div',{'class':'pv-text-details__left-panel'})
    #extracting html tags to find location
    intro=soup.find('div',{'class':'pv-text-details__left-panel mt2'})

    name_loc = name_div.find("h1")
    # Extracting the Name
    name = name_loc.get_text().strip()
    # strip() is used to remove any extra blank spaces

    # this gives us the HTML of the tag in which the Company Name is present
    works_at_loc = name_div.find("div", {'class': 'text-body-medium'})
    # Extracting the Company Name
    works_at = works_at_loc.get_text().strip()
    print("\n"+name)
    print(works_at)

    # this gives us the HTML of the tag in which the location is present
    location_loc = intro.find_all("span", {'class': 'text-body-small'})

    # Ectracting the Location
    location = location_loc[0].get_text().strip()
    print(location)

    #extracting the html tags to find the number of connections
    conn_div = soup.find('li', {'class': 'text-body-small'})
    conn_loc = conn_div.find("span", {'class': 't-bold'})
    connection = conn_loc.get_text().strip()
    print("Connections: "+connection)

    #printing the about of the profile
    print("\nAbout:")
    abt_sec=soup.find('div',{'class':'display-flex full-width'})
    about=abt_sec.get_text().strip()
    print(about)

    print("\nExperience:")
    #extracting the html tags to find the experience of the profile
    exp_sec=soup.find('div',{'class':'pvs-list__outer-container'}).find('ul')
    job_loc=exp_sec.find('span',{'class':'visually-hidden'})

    #printing the job title, company name and duration from the experience section
    job_title=job_loc.get_text().strip()
    print(job_title)
    comp_loc=exp_sec.find('span',{'class':'t-14 t-normal'}).find('span')
    comp_title=comp_loc.get_text().strip()
    print(comp_title)
    time_loc=exp_sec.find('span',{'class':'t-14 t-normal t-black--light'}).find('span')
    time_title=time_loc.get_text().strip()
    print(time_title)

driver.quit()
```

Output:



```
Simran Gandhi
Corporate Partnership/CSR/Fundraising Consultant
Pune, Maharashtra, India
Connections: 381

About:
A young, dynamic, and enthusiast development professional, having 4+ years of experience in the social sector from raising fund
s to Implementing to reporting of various development projects in the sector of Education, Health and Hygiene, Nutrition, Wash
and Skilling. A good relationship builder as well as a great communicator.A young, dynamic, and enthusiast development professi
onal, having 4+ years of experience in the social sector from raising funds to Implementing to reporting of various development
projects in the sector of Education, Health and Hygiene, Nutrition, Wash and Skilling. A good relationship builder as well as a
great communicator.

Experience:
Assitant Manager: Partnerships Development
WOSCA's Life-Lab Science Program · Full-time
Apr 2023 - Present · 5 mos


Navin K.
Co-Founder & CEO
Delhi, India
Connections: 296

About:
KEY COMPETENCIES⍰ Statistical and Machine Learning Algorithms:Regression, Logistic Regression, Decision Tree, KNN, Clustering,
Naïve Bayes Classifier, Deep learning, SupportVector Machine, ANN, Frequent Pattern Mining, Image/Video Analysis, Semi-Supervis
ed Learning, Time SeriesAnalysis and Forecasting⍰ Text Mining, Data Crawling &amp; NLP platform: SAS EMINER,SAS Contextual Anal
ysis, SAS IRSKEY COMPETENCIES
⍰ Statistical and Machine Learning Algorithms:
Regression, Logistic Regression, Decision Tree, KNN, Clustering, Naïve Bayes Classifier, Deep learning, Support
Vector Machine, ANN, Frequent Pattern Mining, Image/Video Analysis, Semi-Supervised Learning, Time Series
Analysis and Forecasting
⍰ Text Mining, Data Crawling &amp; NLP platform: SAS EMINER,SAS Contextual Analysis, SAS IRS

Experience:
Co-Founder
Onelogica · Full-time
Jan 2023 - Present · 8 mos

SRISHTI SINGH
Student at SRM University
Delhi, India
Connections: 112

About:
👋🏽 Hi, my name is Srishti.🎓 Graduated with a Bachelor's of Technology in Computer Science and Engineering. (June 2022)🌇 Delh
i resident, with a passion for Data Analytics and Software Development that will grow your business👋🏽 Hi, my name is Srishti.

🎓 Graduated with a Bachelor's of Technology in Computer Science and Engineering. (June 2022)

🌇 Delhi resident, with a passion for Data Analytics and Software Development that will grow your business

Experience:
Data Engineer Trainee
Onelogica · Full-time
Jan 2023 - Present · 8 mos

Abhika Mittal
Data Analyst Intern at onelogica || Java || Front end Developer || Hacktoberfest Contributor || 200+ Leetcode problem || DSA
Delhi, India
Connections: 2,810

About:
Abhika_Mittal_Resume.pdfAbhika_Mittal_Resume.pdf

Experience:
Data Analyst Intern
Onelogica · Internship
Jul 2023 - Present · 2 mos
```

Breakdown of code:

- Importing Libraries: You import the necessary libraries including requests, time, random, BeautifulSoup, and webdriver from selenium.

- Initializing Web Driver: You create a webdriver.Chrome() instance, which is an automated browser controlled by your Python code. You open LinkedIn's login page.

- Login: You locate the email and password input fields by their IDs and provide your email and password. Then you locate and click the login button.
- DataFrame: creating a dataframe of the excel sheet whose rows(containing profile link) needs to be traversed one at a time

- Navigating to Profile: You provide the profile URL you want to scrape and open it using the WebDriver.

- Scrolling Down: You scroll down the profile page to load more content. This is done by repeatedly executing the command that scrolls to the bottom of the page. You wait for a few seconds after each scroll.

- Parsing HTML with BeautifulSoup: You extract the HTML source code of the loaded page using driver.page_source and create a BeautifulSoup object (soup) to parse it.

- Extracting Name and Company:

You locate the <div> element with the class pv-text-details__left-panel which contains the name and company information.
You find the <h1> tag within this <div> to extract the name.
You locate the <div> with class text-body-medium to extract the company name.

- Extracting Location:

You locate the <div> with class pv-text-details__left-panel again to find the location information.
You find the <span> elements with class text-body-small to extract the location.

- Extracting Connection Count:

You find the <li> element with class text-body-small which contains the number of connections.
You locate the <span> with class t-bold to extract the connection count.

- Extracting About Section:

You find the <div> with class display-flex full-width which contains the About section.
You extract the text content from this section.

- Extracting Experience Section:

You locate the <div> with class pvs-list__outer-container and find the <ul> element within it. This contains the experience section.
You locate the job title, company name, and duration by finding specific <span> elements within the experience section.

- Printing Results:

You print the extracted name, company, location, connection count, about section, job title, company name, and duration.