Perfect! 🚀 Let’s outline and build the **"LinkedIn Job Scraper"** project with Selenium, BeautifulSoup, and pandas.

**Note**: LinkedIn has strict anti-scraping measures (dynamic content, bot detection, ToS restrictions). This project should only be used **for learning/testing with your own account**. For production, prefer the **LinkedIn Jobs API alternatives** (if available).

**Project: LinkedIn Job Scraper**

**Objective**

Automate scraping job listings (title, company, post date) from LinkedIn with Selenium and BeautifulSoup, then analyze them.

**Tools**

* **Selenium** → automate LinkedIn navigation & login
* **BeautifulSoup** → parse job HTML
* **pandas** → clean & store data
* **matplotlib** → job frequency visualization

**Step-by-Step Mini Guide**

**1. Setup Selenium + Login**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.common.keys import Keys

import time

# Setup

driver = webdriver.Chrome()

driver.get("https://www.linkedin.com/login")

# Login (replace with your own test creds)

username = driver.find\_element(By.ID, "username")

password = driver.find\_element(By.ID, "password")

username.send\_keys("your\_email")

password.send\_keys("your\_password")

password.send\_keys(Keys.RETURN)

time.sleep(5) # wait for login

**2. Search Jobs with Keywords & Location**

# Navigate to Jobs page

driver.get("https://www.linkedin.com/jobs/")

# Enter search

time.sleep(3)

search\_box = driver.find\_element(By.CSS\_SELECTOR, "input[aria-label='Search jobs']")

search\_box.send\_keys("Data Scientist")

search\_box.send\_keys(Keys.RETURN)

time.sleep(5)

**3. Scrape Job Titles, Company, Post Date**

from bs4 import BeautifulSoup

import pandas as pd

job\_data = []

soup = BeautifulSoup(driver.page\_source, "html.parser")

jobs = soup.find\_all("div", {"class": "base-card"})

for job in jobs:

title = job.find("h3", {"class": "base-search-card\_\_title"})

company = job.find("h4", {"class": "base-search-card\_\_subtitle"})

date = job.find("time")

job\_data.append({

"title": title.text.strip() if title else None,

"company": company.text.strip() if company else None,

"date": date.text.strip() if date else None

})

df = pd.DataFrame(job\_data)

**4. Save to CSV**

df.to\_csv("linkedin\_jobs.csv", index=False)

print("Jobs saved to linkedin\_jobs.csv")

**5. Remove Duplicates**

df.drop\_duplicates(subset=["title", "company"], inplace=True)

**6. Visualize Job Frequency by Company**

import matplotlib.pyplot as plt

company\_counts = df["company"].value\_counts().head(10)

company\_counts.plot(kind="bar", figsize=(8,5))

plt.title("Top Hiring Companies")

plt.xlabel("Company")

plt.ylabel("Job Postings")

plt.xticks(rotation=45)

plt.show()

**Deliverables**

1. **Bot Script** → linkedin\_scraper.py
2. **CSV Data** → linkedin\_jobs.csv
3. **Visualization** → bar chart of job frequency by company

**Run Project**

python linkedin\_scraper.py

You’ll get:

* linkedin\_jobs.csv with scraped jobs
* A **bar chart** of top hiring companies