LinkedIn Job Scraper

❖ Introduction

This project automates the process of job hunting on LinkedIn using Python, Selenium, and BeautifulSoup. It searches for jobs based on keywords and location, scrapes job titles, companies, and posting dates, and stores them in a structured CSV file. The script also removes duplicates and generates a visual graph showing which companies are hiring the most. It's a complete end-to-end tool for data-driven job search and market analysis.

❖ Abstract

The LinkedIn Job Scraper is a Python-based automation tool designed to streamline the job search process by extracting job listings directly from LinkedIn. Utilizing Selenium for browser automation and BeautifulSoup for data parsing, the system searches for jobs based on user-defined keywords and location, collects essential job details such as title, company, location, and date posted, and exports the results into a CSV file. To enhance data quality, it includes a duplicate removal mechanism. Additionally, the tool provides a graphical visualization of job frequency by company, offering insights into hiring trends. This project demonstrates practical applications of web scraping, data cleaning, and visualization — making it highly relevant for job seekers, data analysts, and HR tech solutions.

❖ Tools Used

Tool / Library	Purpose
Python	Core programming language for scripting and automation
Selenium	Automates browser actions like logging in, searching, and page loading
BeautifulSoup	Parses and extracts data from HTML content
pandas	Handles data storage, cleaning, and CSV export
matplotlib	Visualizes job frequency by company
Chrome Driver	Interface between Selenium and Chrome browser
python-dotenv	Loads environment variables (e.g., LinkedIn credentials) securely

❖ Steps Involved in Building the Project

- **Set up Environment**
  Install Python, Selenium, BeautifulSoup, pandas, and ChromeDriver.
- **Login to LinkedIn**
  Use Selenium to automate login (with or without cookies).
- **Search for Jobs**
  Automate job keyword and location input.
- **Scrape Data**
  Extract job title, company, location, and post date using BeautifulSoup.
- **Save to CSV**
  Store data using pandas in a CSV file.
- **Remove Duplicates**
  Clean the dataset with drop_duplicates().
- **Visualize Results**
  Plot job counts by company using matplotlib.

❖ Conclusion

The LinkedIn Job Scraper successfully automates job listing extraction, streamlines the data collection process, and provides valuable insights through visualization. By combining Selenium, BeautifulSoup, and pandas, the project demonstrates how automation and data analysis can simplify job hunting and support career or market research effectively.