Project Title: LinkedIn Job Scraper

Introduction

The LinkedIn Job Scraper is a Python automation project designed to extract job listings from LinkedIn using web scraping tools. It helps users automatically gather job data based on specific keywords and locations, making job search faster and more efficient.

Problem Statement

Manual job searching on LinkedIn can be time-consuming, especially when filtering through hundreds of postings. This tool solves that by automatically scraping relevant job listings, helping users quickly find opportunities that match their interests.

Abstract

This project uses Selenium to automate LinkedIn login and job search, BeautifulSoup to parse HTML, and pandas to organize data. The scraper collects job titles, companies, post dates, and more, then stores everything in a CSV file. It also removes duplicates and supports visualization like job count by company.

Tools & Technologies Used

- Python
- Selenium
- BeautifulSoup
- pandas
- CSV and time modules

Internship Project Report - LinkedIn Job Scraper

Steps Involved in Building the Project

1. Set up Selenium to automate LinkedIn login securely.
2. Defined search keywords and target locations.
3. Navigated job listings and scraped details using BeautifulSoup.
4. Cleaned and stored data in CSV using pandas.
5. Removed duplicates and added date/time stamps.
6. (Optional) Visualized job frequency by company using matplotlib.

Sample Output Description

The output is a CSV file containing job titles, companies, job locations, and posting dates. Each run of the scraper can collect multiple job entries, which helps create a structured job database. You can optionally create a bar chart to visualize which companies posted the most jobs.

Conclusion

This project allowed me to explore web automation, HTML parsing, and data handling in Python. It also helped me understand how real-time job data can be collected and organized efficiently. The LinkedIn Job Scraper can be further enhanced with filters and alerts in the future.

What I Learned

I improved my skills in Selenium-based automation, web scraping using BeautifulSoup, and data storage with pandas. I also learned how to handle page loads, extract specific HTML elements, and export clean CSV files. Overall, this project strengthened my confidence in real-world Python applications.