LinkedIn Job Scraper Agent

Overview

The **LinkedIn Job Scraper Agent** is a Python script designed to automate the extraction of job postings from LinkedIn. Users can input a specific job title (e.g., "Software Engineer") and location (e.g., "India") to gather details such as job titles, company names, locations, and application links. The data is then saved into a CSV file for convenient access and analysis.

This tool is perfect for job seekers, researchers, or analysts looking to efficiently collect job market data.

Features

- Extracts key job details: job titles, company names, locations, and application links.
- Prevents duplicate entries by tracking unique job links.
- Scrapes multiple pages of job listings (default is 5 pages).
- Operates in headless mode (no visible browser window) for streamlined execution.
- Saves scraped data to a CSV file named linkedin_sde_jobs.csv.

Requirements

To use the LinkedIn Job Scraper Agent on your local system, you'll need:

- **Python 3.x**: Download from python.org.

- **Google Chrome Browser**: Required for web automation. Install from google.com/chrome.
- **ChromeDriver**: Managed automatically by the script using the webdriver-manager library—no manual setup needed.

Dependencies

The script requires the following Python libraries:

- **selenium**: For web automation and interacting with LinkedIn's job search page.
- **webdriver-manager**: Automatically handles ChromeDriver compatibility.
- **pandas**: Processes data and exports it to a CSV file.

Install these using Python's package manager, **pip.**

Installation

Follow these steps to set up the LinkedIn Job Scraper Agent:

1. **Clone the GitHub Repository**
   Download the project files with:

```
git clone
https://github.com/srujan-zenshastra/week1-project-4-linkedin-job-scraper-agent.git cd week1-project-4-linkedin-job-scraper-agent
```

2. **Install Python Dependencies**

   In a terminal or command prompt, navigate to the project directory and run:

```
pip install selenium webdriver-manager pandas
```

3. **Verify Chrome Installation**

   Ensure Google Chrome is installed. Open Chrome to confirm, or install it if missing.

Usage

After setup, you can run and customize the script as follows:

1. **Run with Default Settings**
   Pre-set to scrape "Software Engineer" jobs in "India" across 5 pages. From the project directory, run:

   ```
   python linkedin_scraper.py
   ```

2. **Customize the Search**

   To change the job title or location, edit the script in a text editor (e.g., VS Code). Locate and modify this line:

```
scrape_linkedin_jobs("Software%20Engineer", "India", num_pages=5)
```

Example: For "Data Scientist" jobs in "United States" across 3 pages:

```
scrape_linkedin_jobs("Data%20Scientist", "United%20States", num_pages=3)
```

```
python linkedin_scraper.py
```

1. **Output**
   The scraped data is saved to linkedin_sde_jobs.csv in the project directory, with columns:
   - **Title:** Job title
   - **Company:** Company name
   - **Location:** Job location
   - **Link:** Job posting URL

How It Works

The script:

1. **Sets Up the Browser:** Launches a headless Chrome browser with Selenium and webdriver-manager.

2. **Navigates to LinkedIn:** Builds a search URL with the job title and location, then opens it.
3. **Scrolls and Loads Jobs:** Scrolls the page to load more listings, monitoring for new content.
4. **Extracts Data:** Finds job postings via their HTML class (base-card) and collects details.
5. **Saves Results:** Stores unique jobs in a list, converts to a pandas DataFrame, and exports to CSV.

Configuration Options

Customize these in the script:

- **Job Title:** First argument in scrape_linkedin_jobs() (e.g., "Product%20Manager").
- **Location:** Second argument (e.g., "Canada").
- **Number of Pages**: num_pages (default 5). Adjust as needed.
- **Wait Times:** Modify time.sleep() values (e.g., time.sleep(5) to time.sleep(10)) for slower/faster internet.