Advanced LinkedIn Profile Scraper using Browser Automation, Caching, and LLM Agents

Project Overview

This project focuses on the development of an advanced LinkedIn browser automation scraper designed to extract detailed information from at least 200 LinkedIn profiles. The scraper integrates browser automation tools with intelligent agents driven by large language models (LLMs) to optimize extraction, enhance accuracy, and ensure ethical, stealthy scraping.

Key Objectives

- Scrape 200+ LinkedIn profile entries with high reliability.
- Use browser automation tools (e.g., Selenium, Playwright) to simulate human-like behavior and bypass bot detection.
- Implement caching mechanisms to avoid redundant data fetching and reduce server load.
- Integrate LLM-driven agents for intelligent decision-making such as:
- - Filtering relevant profiles.
- - Dynamically generating scraping strategies.
- - Handling unexpected page layouts or CAPTCHA interruptions.

Technologies Used

- Python 3.x
- Selenium / Playwright for browser automation
- SQLite / Redis / LocalStorage for caching
- OpenAI GPT or similar LLM for agent-based strategy
- Pandas for data handling and export
- CSV / JSON for data output

LLM Agent Capabilities

- Classify job titles and skills using NLP
- Detect and skip duplicate or irrelevant profiles
- React to DOM changes or LinkedIn updates
- Summarize scraped data for fast analysis

Output

Structured CSV file containing name, job title, location, company, profile link, and relevant skills for each individual.

Ethical Considerations

- Follows ethical scraping principles by respecting LinkedIn's robots.txt where applicable
- Designed for academic and placement support use only
- Does not store or distribute private or login-protected data

Project Status

- Core scraper: ✅ Complete
- Caching mechanism: ✅ Implemented
- LLM integration: 🛠️ In progress
- UI or dashboard: 🔜 Planned (optional)