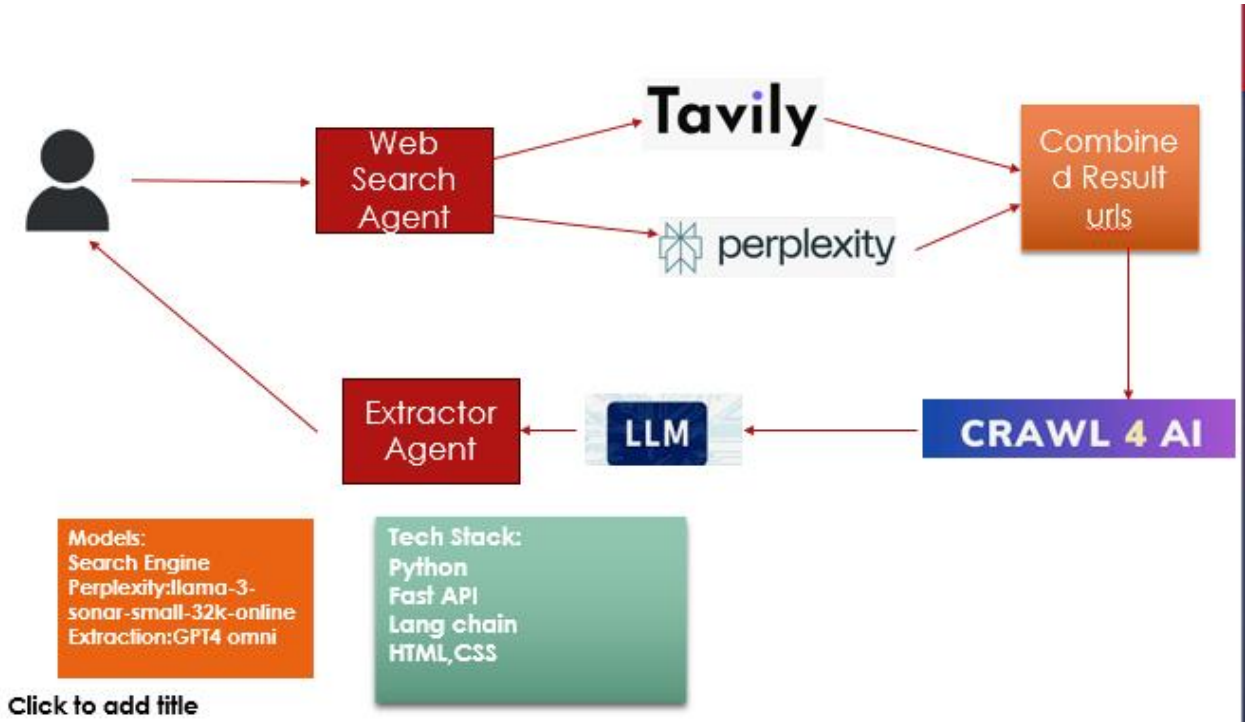
Company Search Subsidiaries Associated Domains Agents

Objective:

The goal of this assignment is to evaluate your ability to design and implement an AI agent capable of **browsing the internet to identify and list subsidiaries and associated domains for a given company name**. This task will assess your skills in data collection, web scraping, natural language processing, and AI-driven information retrieval, all within a constrained timeframe.

Approach:

Architecture



1. **User Input:**
    - The user will provide the company name through the user interface.
    - This triggers the **Search Company API**.
2. **Web Search Process:**

- The initial API call will be directed to the **Web Search Agent**, where **Tavily Search** and **Perplexity Search** APIs will be executed.
- These tools search the web and return URLs along with summarized content related to the company. This process leverages the **Lang chain orchestration framework**, utilizing custom prompt recipes.

3. **URL Filtering and Ranking:**
    - The retrieved URLs are filtered and ranked based on an AI scoring algorithm to identify the most relevant sources.
    - After the Web Search Agent gathers the data, unique URLs are filtered for further processing by the **Web Crawler Agent**.
4. **Web Crawling with Crawl4Ai:**
    - Instead of traditional methods like Beautiful Soup, the **Crawl4Ai** tool is utilized.
    - **Advantages of Crawl4Ai**:
        1. Free and open-source.
        2. LLM-friendly output formats and concurrent crawling.
        3. Comprehensive media extraction (images, audio, video).
        4. Metadata extraction for additional context.
        5. Custom hooks for authentication, headers, and page modifications.
        6. User agent customization for HTTP requests.
        7. Screenshot capability during crawling.
        8. Execution of custom JavaScript before crawling.
        9. Advanced chunking and extraction strategies (topic-based, regex, sentence chunking, cosine clustering, LLM extraction).
        10. CSS selector support for targeted content extraction.
    - **LLM Extraction Strategy and Schema** within Crawl4Ai is employed to focus on extracting metadata related to company subsidiaries and associated domains.
5. **Content Extraction and Output:**
    - After web scraping, the **Extraction Agent** is executed, running an LLM chain flow with specific prompt instructions and context.
    - The final output is a JSON file containing the subsidiaries and associated domains of the given company.

Optimization Opportunities

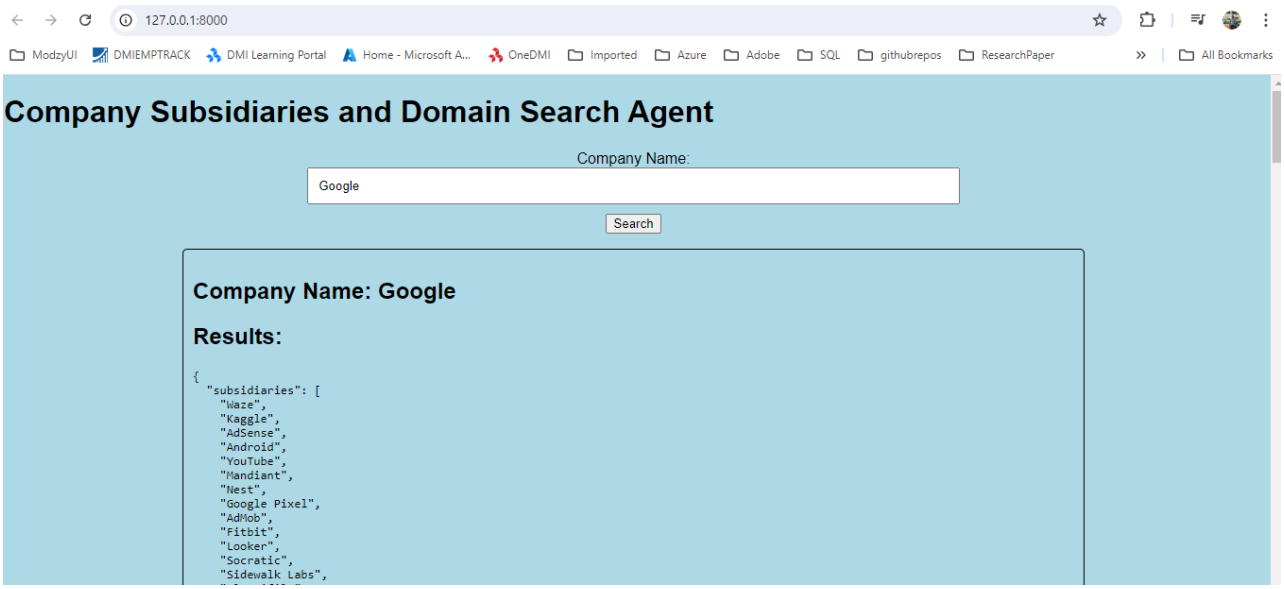
1. **Expanded Search Sources:**

- Integrate **Google Serper API** and **Bing Search** to retrieve additional company-specific URLs.

2. **URL Validation:**
    - Implement eval mechanisms to validate the relevance of fetched URLs in identifying company subsidiaries and domains.
3. **Web Crawler Enhancements:**
    - Filter out URLs where content cannot be extracted due to legal or non-public domain access.
    - Develop strategies to address these issues, including custom search mechanisms.
4. **Entity Recognition:**
    - Utilize **NER extraction** during web scraping to identify key entities within the content.
5. **Evaluation Mechanisms:**
    - Incorporate tools like **Uptrain, Ragas,** or **Guardrails** to evaluate context relevancy, faithfulness, answer completeness and security aspects. This helps assess the AI agent's performance in extracting subsidiaries and domains.
6. **Noise Reduction:**
    - Enhance response quality by implementing noise reduction mechanisms for domain extraction, ensuring outputs follow the pattern of valid domains (e.g., google.com, adsense.co) instead of irrelevant words.

7. **Companies with no public data:**
    - Strategy needs to be created for companies that doesn't have public access data. May be manually search for data. Introducing the database or storage to fetch the data, create some documents, then apply crawler to read data and find the desired information.

API/UI Interface Response



How to Use:

- Project has Readme.md file that will walk through the solution.
- Code is also committed in GitHub with public read only access:
  Reference url:
  https://github.com/vikassalaria2412/SearchCompanyDomainAgent.git