Rufus Web Scraping Client

Explanation

In this application, I designed a solution that leverages Langchain agents to intelligently scrape relevant content from the web based on user queries and a target website URL. The process starts with the user providing both the query and URL, which are passed to GPT-3.5 Turbo via a Langchain agent. The agent utilizes the `get_sitemap` tool, built with BeautifulSoup, to generate the sitemap of the target website. GPT-3.5, guided by the agent, creates a step-by-step plan to identify which sections of the website are most relevant to the user's query. Using the sitemap and the plan, the Langchain agent selects the most relevant links. These links are then passed to a web scraping script that collects the content and exports it into a JSON format. The entire application is exposed via a FastAPI, providing a simple API endpoint, and can be easily installed via pip for seamless deployment.

Challenges

One challenge I encountered was that, initially, the LLM struggled to select the most relevant links based solely on the user query. To address this, I prompted the LLM to first come up with a rational plan outlining how it would approach answering the query. This plan would then guide its decision-making process in choosing the most relevant links from the sitemap. Another challenge was ensuring that the LLM provided structured and usable output. Early attempts resulted in inconsistent or unstructured data, which made downstream processing difficult. Through iterative experimentation, I developed an example format that consistently produced well-structured output, which significantly improved the flow of the scraping process and the quality of the extracted content.