Web Content Q&A Tool Documentation

Overview

This Web Content Q&A Tool allows users to enter URLs of websites, scrape their content, and ask questions based on that content. The tool will only answer based on the data scraped from the provided URLs, ensuring that answers are contextually relevant and grounded strictly in the scraped website content.

Features

1. **Enter multiple URLs**: Users can input one or more website URLs from which the tool will scrape data.
2. **Ask questions**: After scraping the content, users can ask questions related to the content of the websites.
3. **Get answers**: The tool generates answers based only on the scraped data, using AI to analyze and retrieve the most relevant information.

Tech Stack

Backend (Flask)

- **Flask**: Lightweight Python web framework to build the API.
- **LangChain**: Used to process and store the scraped data efficiently.
- **Google Gemini (via LangChain)**: For generating AI-powered answers.
- **Chroma**: Vector database for storing and retrieving content.
- **crawl4ai**: A web scraping library for advanced web crawling and content filtering.
- **Asyncio**: To handle asynchronous scraping tasks efficiently.

Frontend (React + Tailwind CSS)

- **React**: A modern JavaScript framework to build a dynamic single-page application (SPA).
- **Tailwind CSS**: A utility-first CSS framework to style the application and create a responsive UI.
- **Lucide React Icons**: To use icons for UI components like send, link, loader, etc.
- **Card and Alert components**: For better UI structuring and error handling.

Conclusion

This Web Content Q&A Tool uses modern technologies to create an interactive and efficient way of extracting, storing, and querying website content. The tool's design is minimal and user-friendly, ensuring that users can easily enter URLs, scrape data, and ask questions based on the scraped content.