Web Research Agent

**1. Project Overview and Agent Architecture**

The Web Research Agent is an AI-powered research assistant that automates the end-to-end process of conducting research online. By combining natural language understanding, search API integration, content extraction, and summarization using large language models, the agent simulates human-level research tasks in real time. This system is suitable for students, professionals, and researchers who need quick, accurate summaries from across the web.

* 1. **Project Structure**

A flowchart of a web scraping

AI-generated content may be incorrect.

**2. Agent-Based Workflow**

The Web Research Agent works as a team of specialized assistants (agents). Each one has a specific job. They pass the information step by step to complete the task of researching and summarizing web content based on what the user wants.

**2.1 Query Analyzer (query\_analyzer.py)**

This agent is like when a user types a question or topic, it figures out what kind of information the user is looking for.

How it works:

• It sends the user's input to the Groq language model.

• The model identifies:

o Query type

o Intent – what exactly the user wants to learn.

o Top keywords – the most important words from the query, used for search.

**2.2 Web Search Agent (web\_search.py)**

This agent uses those keywords to search Google using SerpAPI, a tool that gives structured search results.

How it works:

• Searches the web using SerpAPI and the extracted keywords.

• Pulls around the most relevant results.

• Returns:

o The title of each article.

o The link (URL).

o A short description (snippet).

**2.3 Web Scraper (web\_scraper.py)**

This agent visits each of the article links and downloads their full webpage content.

How it works:

• Uses the requests library to fetch web pages.

• Has error-handling in case of Bad URLs and Connection failures

**2.4 Content Extractor (content\_extractor.py)**

This agent “cleans” the web pages by extracting only the useful parts and ignores the junk.

How it works:

• Uses BeautifulSoup to parse the HTML.

• Applies logic to:

o Remove unwanted content like ads, footers, navigation bars.

o Keeps the main body text.

Thus, providing clean, readable text so the summary can be more accurate.

**2.5 Content Analyzer (content\_analyzer.py)**

This agent summarizes each article by understanding its content and what the user is asking.

How it works:

• Builds a prompt for the Groq LLM that includes:

o The original user query

o The cleaned content from the article

• Asks Groq to:

o Summarize only what's relevant to the user query

o Highlight the main points and facts

**2.6 Summarizer (summarizer.py)**

If multiple summaries are available, this agent combines them into one final report.

How it works:

• Collects summaries from all the articles.

• Analyzes and merges them into one clean, easy-to-read final summary.

• Removes repetition and ensures flow.

**3. Integration with External Tools**

Each agent utilizes specific tools and APIs to perform its function. The three main integrations are with Groq API, SerpAPI, and BeautifulSoup.

**4. Streamlit Integration and User Interface**

The interface is built using Streamlit, offering an intuitive, interactive frontend that ties together all backend agents into a single pipeline. Users simply enter their query, and the system displays summarized research results and links to sources

**5. Agent Responsibilities Summary**

The Web Research Agent is designed to perform a variety of autonomous research tasks. It searches the web for information related to the user's query, extracts relevant data from multiple websites, and identifies and summarizes the latest news articles on specific topics. Additionally, it can crawl web pages to gather detailed information and compile all findings into an organized, easy-to-understand format tailored to the user's research needs.

**6. Deployed in Streamlit Community**

The entire application is deployed on the Streamlit Community Cloud, allowing anyone to access it via a web browser without installing anything locally.

\*\*\*\*\*\*\*\*\*