# **Assessment Report: Web Research Agent**

Poornima MC - https://github.com/POORNIMA-MC/web-research-agent

## **1. Objective**

The objective of this project is to develop a Web Research Agent that can automatically:

* Accept a user query,
* Search the web for relevant information using SerpAPI,
* Scrape web content from the top search results,
* Clean and summarize the extracted content using transformer-based models,
* And display the summarized information neatly on a web interface built using Flask.

The goal is to save users time by providing condensed, accurate, and relevant summaries from multiple sources without the need to manually browse through various articles.

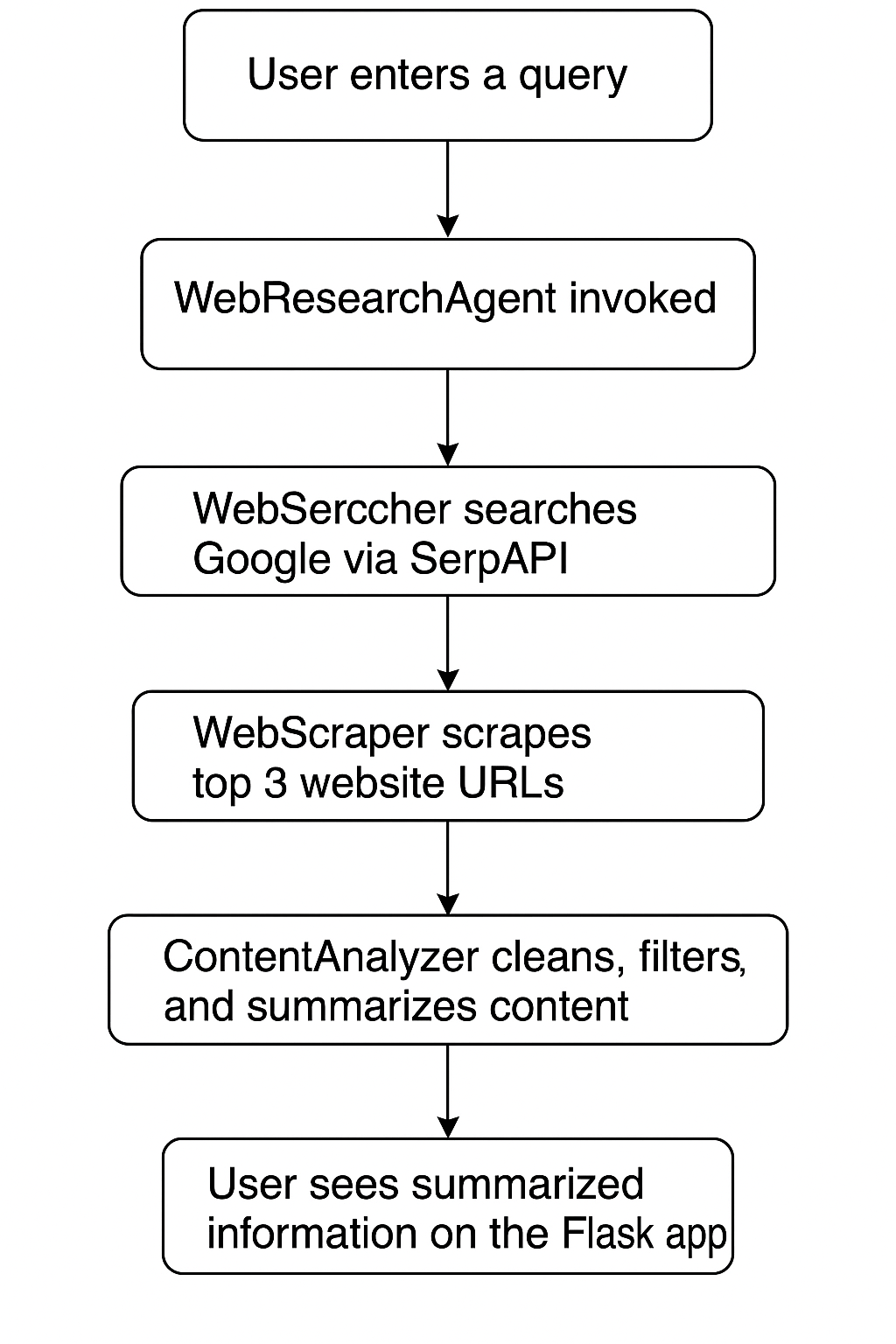
## **2. System Architecture and Flowchart**

### **2.1 Architecture Overview**

The system is composed of the following major components:

* **Flask Web Interface** for user interaction,
* **WebSearcher** to perform search operations using SerpAPI,
* **WebScraper** to extract website content,
* **ContentAnalyzer** to clean and summarize text.
* **WebResearchAgent** as the orchestrator of all components.

### **2.2 Flowchart**



**3. Module-wise Explanation**

### **3.1 Flask Web Application**

* Accepts user queries through an HTML form.
* Handles both GET and POST methods.
* Displays both the user's query and the resultant summarized content.

### **3.2 WebSearcher (Using SerpAPI)**

* Takes a user query as input.
* Performs a Google search using SerpAPI.
* Extracts and returns top 3 organic result URLs.

### **3.3 WebScraper (Using Selenium and BeautifulSoup)**

* Loads each URL in a headless Chrome browser.
* Extracts all <p> (paragraph) tags from the HTML.
* Joins all paragraph texts into a single raw text body.

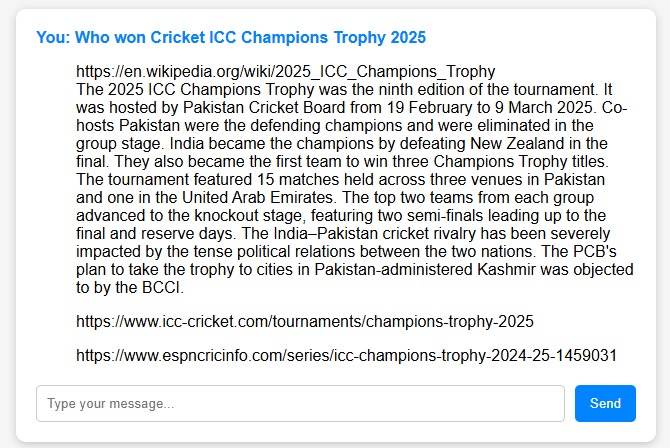
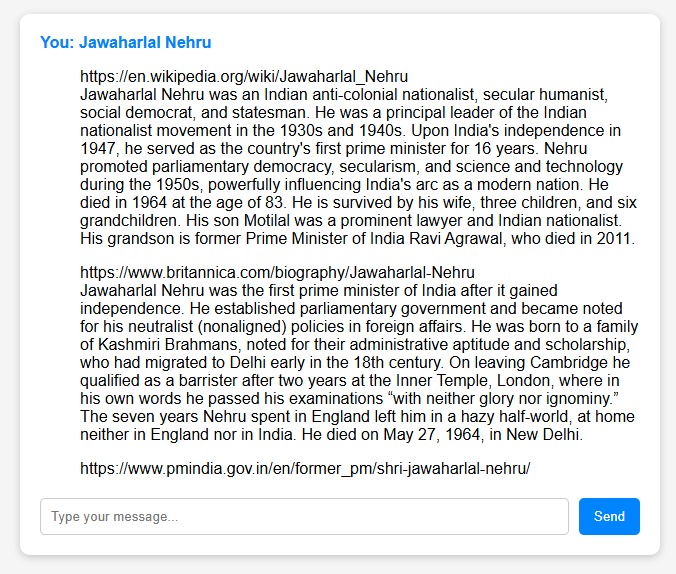
### **3.4 ContentAnalyzer (Cleaning and Summarization)**

* Cleans the scraped content by removing unwanted boilerplate texts (e.g., "contact us", "terms of service").
* Filters relevant sentences based on keyword matching with the query.
* Uses Hugging Face’s BART-large-cnn model to summarize large extracted content.
* If the extracted text is already short and relevant, it is returned directly without summarization.

### **3.5 WebResearchAgent (Main Controller)**

* Acts as the central coordinator.
* Manages the flow:  
  + Search → Scrape → Analyze → Return summarized results.

## **4. Results**



## **5. Conclusion**

The Web Research Agent effectively automates web research by integrating:

* Google search automation,
* Website scraping,
* Content cleaning and summarization,
* And web-based display of results.

It achieves the following benefits:

* Saves time for users by summarizing multiple articles.
* Improves efficiency in gathering information.
* Provides clean and relevant outputs without unnecessary text.

The system is modular, scalable, and can be extended in the future by:

* Adding more sophisticated summarization models,
* Introducing document ranking based on content quality,
* Or even providing source citations with each summary.