Gemini RAG Web Research Agent

Project Documentation

1. How the Agent Works

The system uses a layered approach to research information online and provide comprehensive answers:

User Interface

- Built with Streamlit (app.py)
- Provides simple controls for entering questions and setting up API access
- Shows search results and final answers in an easy-to-read format

Query Understanding

- Analyzes what you're really asking using Gemini 2.5 Pro
- Identifies:
    - What you want to know (comparisons, news, etc.)
    - What type of information you need (official sources, reports, etc.)
    - Relevant time period
    - Key search terms (organized into logical groups)

Search & Collection

- Uses Google Programmable Search to find relevant websites
- Special handling for homepages:
    - Detects if a link goes to a site's main page
    - If so, explores internal links on that site
- Smart link selection:
    - Analyzes internal links for relevance to your question
    - Only follows links that meet a minimum relevance score (0.3)

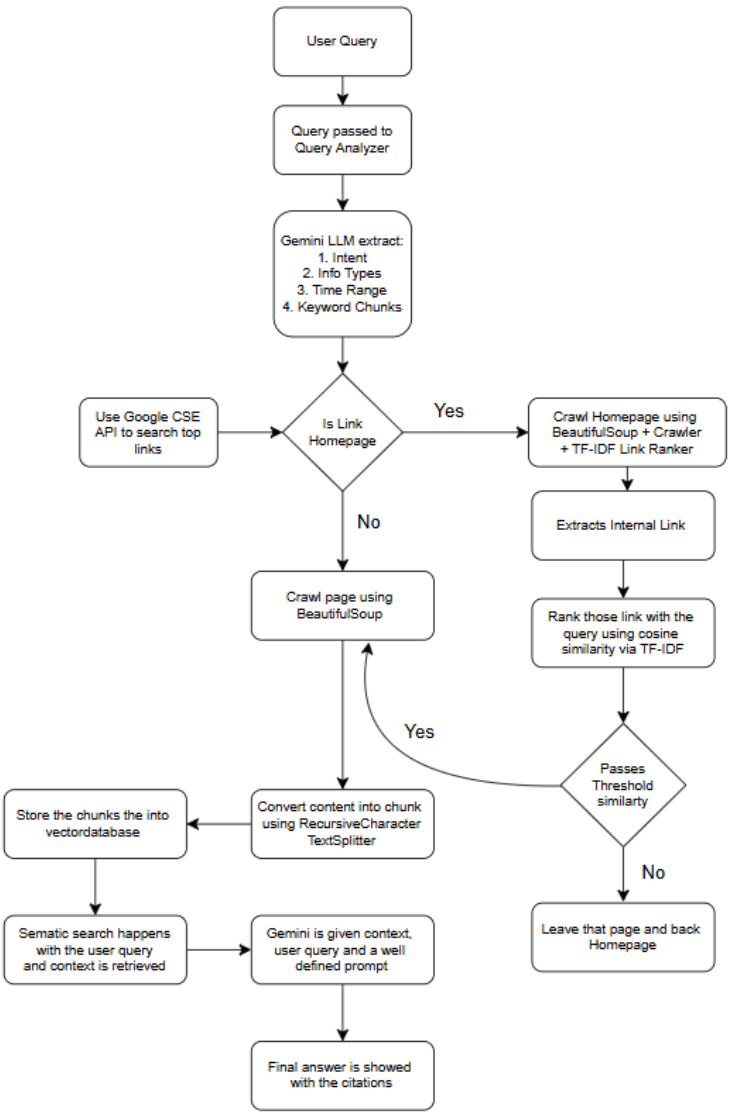
Processing & Storage

- Extracts main content from web pages using BeautifulSoup
- Breaks text into manageable chunks that overlap slightly
- Creates embeddings (numerical representations) of each text chunk
- Stores everything in ChromaDB with source information

Answer Generation

- Retrieves the most relevant text chunks based on your question
- Scores each piece of information for quality and relevance
- Ensures information comes from diverse sources
- Combines everything into a well-structured answer
- Includes citations and relevance ratings

2. Architecture



3. How I Make Gemini Work Smarter

I've designed specific instructions for Gemini 2.5 at each step:

For Query Analysis

- Structured format to extract key information about your question
- Groups related keywords to make searches more effective

For Relevance Scoring

- Rates each piece of information on a 1-5 scale
- Ensures I use diverse, high-quality sources

For Final Answer Creation

- Provides all relevant information pieces with their sources
- Clear formatting guidelines for readable results
- Rules for consistent citations, tables, and structure

3. Tools & Connections

External Tools

- Google Custom Search Engine API: Finding relevant websites
- BeautifulSoup: Extracting content from web pages
- LangChain: Processing text and connecting to the database
- Gemini 2.5 Pro: Understanding questions and creating answers
- ChromaDB: Storing and retrieving information efficiently

System Connections

- API keys stored securely in .env file
- Local database storage in chroma_store/ folder
- Gemini configured at startup

4. How I Handle Problems

Search Issues

- Gracefully handles Google API failures
- Logs problem pages and continues with available results

Content Extraction Problems

- Catches website connection errors
- Records problematic URLs
- Continues with successfully retrieved content

AI Processing Failures

- Manages Gemini errors during content generation
- Uses simpler response formats as backup
- Includes original query information when needed

Ranking Difficulties

- Falls back to unranked results if similarity scoring fails

Conclusion

This system is designed to be reliable and transparent. All information in the final answers can be traced back to their original sources. I avoid unnecessary processing by only exploring web pages that are truly relevant to your question.