<AI-Powered Web Research Assistant>

For Up-to-Date Information Retrieval

ÇANKAYA UNIVERSITY

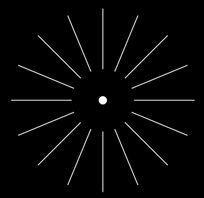
CENG-407 FINAL PROJECT

Team Members:

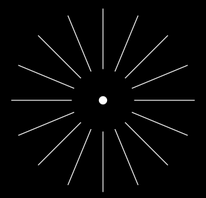
- Ahmet Buğra ARSLAN
- Ege ALTUĞ
- Eren TÜRKMEN
- İbrahim Efe ÇELEMEN
- Yaşar Kağan ŞAHBAZ

Supervisor: Faris Serdar TAŞEL

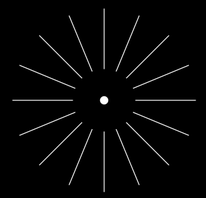
The Problem: Limitations of Current Solutions



Standard LLMs function as "black boxes." They generate answers without providing citations or references, making verification impossible



Hallucinations:Without external data, AI models often invent false facts to fill the gaps in their knowledge.



Manual Search Effort: Traditional search engines provide data but require time-consuming manual verification by the user.

Our Solution: Evidence-Based AI Assistant

Transparent Citations ,
Every claim is directly linked to its source URL, allowing users to verify information instantly.

Real-Time Data Access, Integrates Search APIs & Web Crawlers to fetch up-to-the-minute information, overcoming the limits of pre-trained models.

Evidence-First Reasoning,The AI generates answers based only on retrieved facts, minimizing hallucinations and ensuring accuracy.

Why Our Assistant?

- **Transparency:** The system is designed to cite every retrieved chunk explicitly.
- **Conflict Resolution:** We plan to implement specific strategies to filter conflicting facts .
- **Privacy:** The architecture supports Ollama for local execution to ensure privacy .

Target Audience & Use Cases

Who is this system for?

University Students: Require up-to-date facts for assignments without manual searching.

Academic Researchers: Need access to latest papers and verified citations.

Professionals: Need timely industry updates and market trends.

USE CASES

Current Events Research: Summarizing breaking news from multiple sources.

Technology Updates: Tracking new software releases and tech trends.

General Information Retrieval: Quick, reliable answers with direct references.

System Architecture

Modular Design: Independent layers for UI, API, Orchestrator, and Retrieval.

Core Orchestrator: Manages flow control, parallel fetching, and error handling.

Local Execution: Uses Ollama for local LLM inference to ensure privacy.

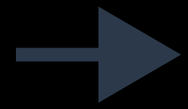
Robustness: Includes fallback mechanisms for API failures.

From Query to Answer: Data Flow

Parallel Retrieval: Fetches data from Search APIs and Web Crawlers concurrently.

Data Processing: Cleans HTML, removes boilerplate, and chunks text.

Conflict Resolution: Resolves contradictions using consensus and recency strategies .



Synthesis: LLM generates the final answer based only on retrieved evidence.

Tools & Technologies

Frontend (UI)

Backend (API):

AI & LLM:

Tools & Technologies

Frontend

React.js: Selected for rapid UI development and responsive design components .

Backend

Python + FastAPI: Chosen for high-performance asynchronous request handling

AI Inference

Decision pending based on Phase-2 testing:
Option A (Local): Ollama (Running LLaMA/Mistral) for maximum privacy and zero cost .
Option B (Cloud): OpenAI / Gemini APIs for higher reasoning capability and speed.

Data Retrieval

Scraping: BeautifulSoup & Scrapy.

Search: Google/Bing Search APIs.

Data Structure & Entities

Evidence-First Approach: Every response is built on "Chunks" of evidence.

Key Entities:

- Source: Domain, URL, Publication Date .
- Chunk: Cleaned text segment from a source .
- Claim: Fact extracted from chunks .
- Final Response: Synthesized answer + Citations .

Reliability & Challenges

Rate Limits: Managed via backoff strategies and caching to prevent blocking.
Dirty Data: Advanced parsing removes ads, scripts, and HTML boilerplate.
Hallucinations: Mitigated by the "Evidence-First" principle; the model explicitly states uncertainty if data is insufficient.
Conflict Handling: Prioritizes peer-reviewed and recent sources over informal ones .

Conclusion & Project Goals

Core Objective: We designed a system to act as an intelligent guide that connects users to the most accurate information on the web.

The "Where": The project aims to solve the "lost in search results" problem by pointing users directly to valid **URLs.**

The "What": By integrating LLM reasoning, we plan to explain the found information clearly, saving users from manual reading.