Deep Research AI Agentic System

Submission for Kairon Qualifying Assignment
Author: Ashutosh Sharma
Date: 24 April 2025

---

1. Project Objective

The goal of this project was to design an **AI agent-based Deep Research System** that:

- Crawls online sources for information (using Tavily API),
- Organizes and summarizes findings (using LangChain + Google API),
- Drafts a final, detailed answer for the user.

---

2. Architecture Overview

The system follows a **Dual-Agent Flow**:

Agent	Role
Research Agent	Uses Tavily API to fetch search results and summarizes key points.
Drafting Agent	Takes the summarized research and crafts a complete, user-friendly answer.

The agents are orchestrated using **LangGraph** (state graph) to model the multi-step flow.

---

3. Technologies Used

- **LangGraph:** For managing agent workflows.
- **LangChain:** For chaining together different LLM calls.

- **Tavily API:** For web crawling and real-time information retrieval.
- **Google Generative AI API:** For text summarization and drafting.
- **Python** as the core programming language.

---

4. System Workflow

1. **User Input:**
   User enters a research question.
2. **Search Phase:**
   `TavilyTool` searches the internet for the latest information.
3. **Summarization Phase:**
   `research_chain` uses Google LLM to summarize search results.
4. **Drafting Phase:**
   `draft_chain` refines the summary into a detailed answer.
5. **Output:**
   The final drafted answer is printed for the user.

---

5. Special Features

- Modular design: Easily extendable to add more agents.
- Minimal dependencies: Lightweight and fast.
- Real-time search: Always updated with latest web results.
- Clean CLI Interface: User can interact smoothly.

---

6. Folder Structure

```
deep-research-ai/
├── main.py
```

```
├── requirements.txt
├── README.md
└── explanation.pdf
```

---

7. How to Run

pip install -r requirements.txt
python main.py

- Set environment variables for API keys before running.
- Enter a research query when prompted.

---

8. Future Improvements

- Add multi-modal capabilities (images, graphs).
- Implement memory for multi-turn research.
- Deploy as a simple web app interface.

---

9. Conclusion

This project demonstrates the ability to combine **web search**, **AI summarization**, and **workflow management** to create a powerful, modular deep research assistant.

It is fully functional and ready for further scaling and deployment.

Thank You