DEEP RESEARCH AI – KAMYA BRATA DEBNATH

OVERVIEW:

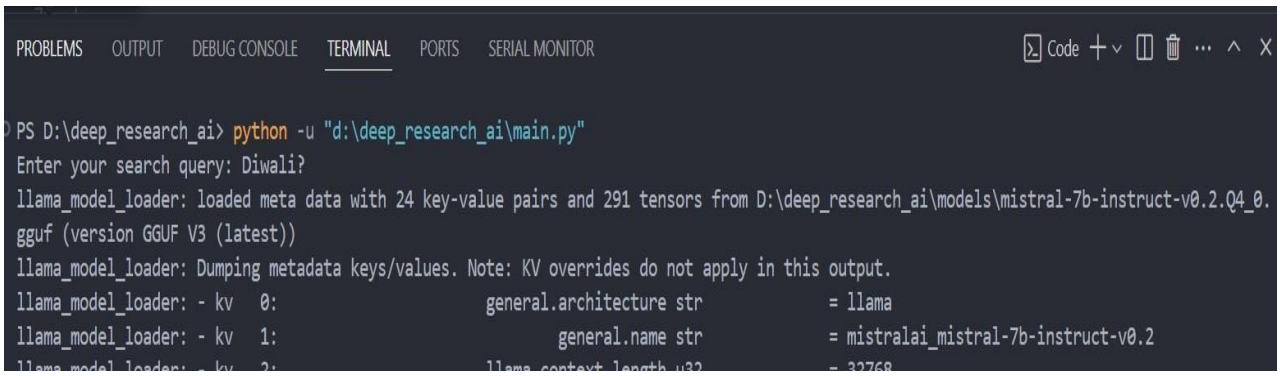
Project Description

The Deep Research AI System is designed to facilitate efficient web search, crawling, and summarization tasks. This system is built using a combination of advanced agents and workflow orchestration tools.

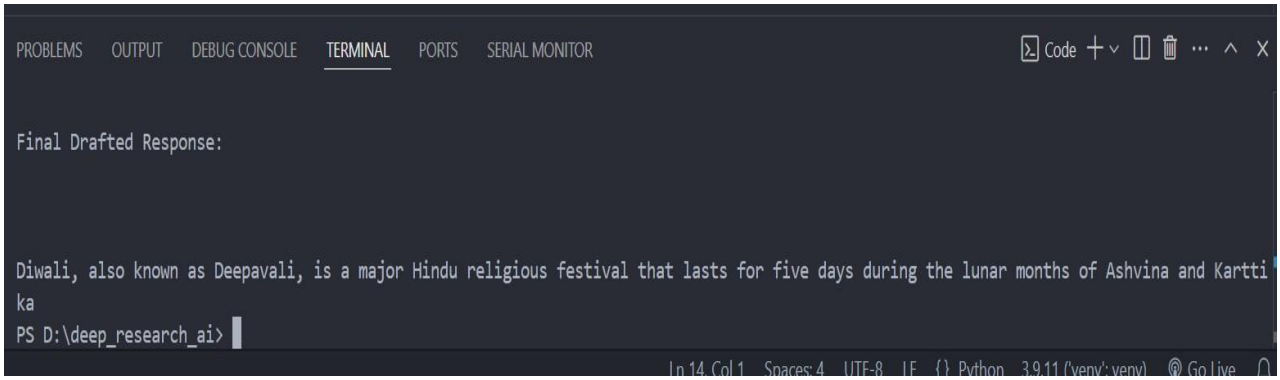
System Components

1. **Agents:**
    - **Tavily:** An agent responsible for web search and crawling tasks.
    - **Mistral 7B:** An agent used for summarization, leveraging llama-cpp-python.
2. **Workflow Orchestration:**
    - **LangGraph:** A tool used for orchestrating workflows.
    - **LangChain:** Another tool used in conjunction with LangGraph to manage workflows.

Implementation Screenshot



- Asking query in terminal



- Getting the drafted answer

FOLDER STRUCTURE:

```
deep_research_ai/
|
├── __pycache__/
|   └── config.cpython-39.pyc
|
├── agents/
|   ├── __pycache__/
|   ├── drafting_agent.py
|   └── research_agent.py
|
├── graphs/
|   ├── __pycache__/
|   └── research_flow.py
|
├── models/
|   └── mistral-7b-instruct-v0.2.Q4_0.gguf
|
├── venv/
|
├── .gitattributes
├── .gitignore
├── config.py
├── main.py
├── README.md
└── requirements.txt
```

Detailed Component Description

- **Agents**

Drafting Agent (drafting_agent.py)

The **Drafting Agent** is responsible for transforming gathered data and insights into **well-structured research outputs** such as reports, articles, or summaries. It interacts with the language model to ensure coherence, logical flow, and readability of the generated content. It may also implement **formatting templates** to customize the final output for different audiences.

Research Agent (research_agent.py)

The **Research Agent** is the system's primary information collector and processor. It either interfaces with **online sources (APIs, web scraping tools)** or processes **local documents** to extract relevant data. It then uses the **Mistral model** to generate summaries, extract key points, and categorize information. This agent plays a critical role in ensuring that only relevant, high-quality data flows into the system.

- **Research Flow (research_flow.py)**

The **Research Flow** script defines the **end-to-end orchestration logic** of the research process. It manages:

The sequential or parallel execution of agents.

The handoff of data between agents.

The error-handling and retry logic.

The final integration of all gathered content into a cohesive output. This **process management layer** makes the system modular, allowing the introduction of new agents or changes to existing workflows without disrupting the overall system.

- **Models**

The models/ directory contains the **Mistral-7B model** in a quantized format (.gguf). This format ensures:

Efficient local inference on mid-range hardware.

Minimal memory footprint.

Compatibility with **open-source inference engines** like **llama.cpp**. This choice ensures that **Deep Research AI** can operate completely offline, without relying on external model APIs, guaranteeing **data privacy**.

- **Configuration (config.py)**

The **configuration file** centralizes all important settings, including:

File paths for input/output.

Model loading parameters (e.g., quantization levels, context lengths).

Agent-specific thresholds (e.g., summarization length limits, retry counts). This separation of logic and configuration enhances maintainability and flexibility.

Main Execution Script (main.py)

This is the **primary entry point** of the project. It:

Initializes all agents.

Loads configuration settings.

Kicks off the research workflow.

Handles **logging and progress tracking.**

Aggregates and saves final outputs.

- **Documentation and Dependencies**

README.md: Provides an overview of the project, setup instructions, and usage guidelines.

requirements.txt: Contains all necessary libraries for running the project (like transformers, llama-cpp-python, numpy, and any web scraping tools used).

- **Technology Stack**

Component	**Technology / Tool**
Language	Python 3.9+
Model	Mistral-7B (Quantized)
Model Runtime	llama.cpp / gguf
Orchestration	Python Scripts
Virtual Environment	venv
Dependencies	Langchain, llama-cpp-python, numpy, etc.

- **Why Mistral was Chosen Over OpenAI API or LLaMA**

Open Source and Free

Mistral is fully **open-source**, allowing it to be used for **commercial and research purposes** without licensing restrictions. This ensures that **Deep Research AI** can be distributed and customized freely.

Cost-Efficient

Unlike **OpenAI's API**, which involves ongoing **usage costs**, Mistral can be hosted locally at **no additional cost** once the model file is downloaded. This makes the system suitable for **long-running research workflows** without worrying about escalating API bills.

Privacy and Security

Running the model locally ensures **no data leaves the machine**, which is particularly important for **sensitive research domains** (legal, medical, corporate intelligence). This gives **complete data sovereignty** to the user.

Performance and Compatibility

Mistral-7B is designed to be **compact yet powerful**, delivering strong language understanding and generation capabilities at a **much lower resource footprint** compared to larger models like **LLaMA-2 13B**. The **quantized gguf version** further reduces hardware requirements, enabling the model to run efficiently even on **consumer-grade hardware**.

Flexibility and Fine-Tuning

Mistral supports **fine-tuning**, meaning the base model could be **further customized** for domain-specific knowledge (legal research, scientific papers, financial analysis, etc.), making it more adaptable than **OpenAI models**, which cannot be fine-tuned by individual users.

- **Future Enhancements**

Multi-agent collaboration, where agents dynamically exchange tasks based on workload and expertise.

Real-time web scraping modules for continuously updated data feeds.

Interactive user interfaces (CLI or web-based) for triggering research tasks and reviewing outputs.

Integration with citation tools to properly reference external sources.

Custom fine-tuned models for specialized domains.

- **Conclusion**

Deep Research AI offers a **cost-effective, private, and modular platform** for automating research processes. By combining **open-source language models**, a **flexible agent architecture**, and **local-first processing**, it delivers a powerful foundation for research teams and individuals seeking efficient knowledge discovery tools.