



AI Agent Assignment Report

i. How the LLM Was Used for Reasoning

In this project, we implemented a Web Research Agent based on the ReAct pattern (Reasoning + Acting). The **reasoning** phase of the agent is performed using **Gemini 1.5 Flash**, a powerful language model from Google's Generative AI suite.

During the reasoning phase:

- The user provides a **topic** (e.g., "AI in Mental Health").
- The agent queries the LLM (Gemini) with a prompt like:

"Generate 5 to 6 insightful and well-structured research questions about the topic: {topic}."

- The LLM generates diverse, meaningful, and relevant research questions that cover different angles of the topic.
- This step ensures the agent has a structured plan before taking action. It simulates how a human might brainstorm sub-questions before beginning research.

Purpose of Reasoning:

- Breaks down a complex topic into manageable sub-questions.
- Directs the agent's attention toward different aspects of the problem.
- Improves the quality and focus of the web search phase by avoiding vague or overly broad queries.

This planning step reflects how humans research: first thinking through what to ask, then acting on those thoughts.

ii. Explanation of the Code and Program Flow

The system is implemented in Python using Google Colab and follows a modular and object-oriented approach. Below is the flow of the program:

1. Initialization

A class WebResearchAgent is defined with the following components:

- topic: User-defined topic.
- model: Gemini 1.5 Flash model for reasoning.
- tavily: Tavily API client for performing web searches.
- questions: List of LLM-generated research questions.
- research_data: Dictionary for storing search results.

2. Reasoning Phase (LLM)

- The method generate_questions() prompts Gemini with a structured instruction to produce 5–6 questions.
- The LLM returns the response as a block of text.
- This is split and cleaned into a list of questions.
- These are printed for verification.

Example:

1. What are the applications of AI in early diagnosis of mental health conditions?
 2. How effective are AI-powered mental health chatbots?
- ...

3. Acting Phase (Web Search)

- The search_web() function takes each generated question.
- Queries Tavily's web search API for up-to-date and relevant articles or data.
- Handles any API issues (like overly long questions) gracefully by truncating and using error handling.

Results are saved in the format:

```
{  
  "question1": [  
    {"title": "Article Title", "content": "Summary of the article..."},
```

```
...  
]  
}
```

4. Report Generation

- The method `generate_report()` compiles the entire output into a markdown-formatted report:
 - Title
 - Introduction
 - One section per question with bulleted answers
 - Conclusion

It follows a clear and readable format ideal for documentation, export, or display in a notebook.

Final Output Sample:

```
# Research Report on AI in Mental Health
```

```
## Introduction
```

```
...
```

```
### What are the challenges of integrating AI in traditional mental health care?
```

```
- **Forbes**: Many clinics face technological and regulatory hurdles...
```

```
- **Wired**: Therapists express concerns about data privacy...
```

```
## Conclusion
```

```
This concludes the structured research conducted by the AI agent.
```

Technologies Used

- **LLM Reasoning:** Gemini 1.5 Flash (models/gemini-1.5-flash-latest)
- **Web Search Tool:** Tavily Python SDK
- **Notebook:** Google Colab
- **Language:** Python 3

github URL: https://github.com/SridharSquare/Agentic_AI_Workshop/tree/main/Day_4/Web%20Research%20Agent%20Using%20the%20ReAct%20Pattern

-