LangGraph Assignment: Multi-Agent Research and Summarization System

---

Problem Statement:

Create a research and summarization agent using LangGraph. The agent should process user queries by determining whether they require reasoning from an LLM, web research, or retrieval from a knowledge base. The agent should leverage multiple specialized sub-agents to generate well-structured responses.

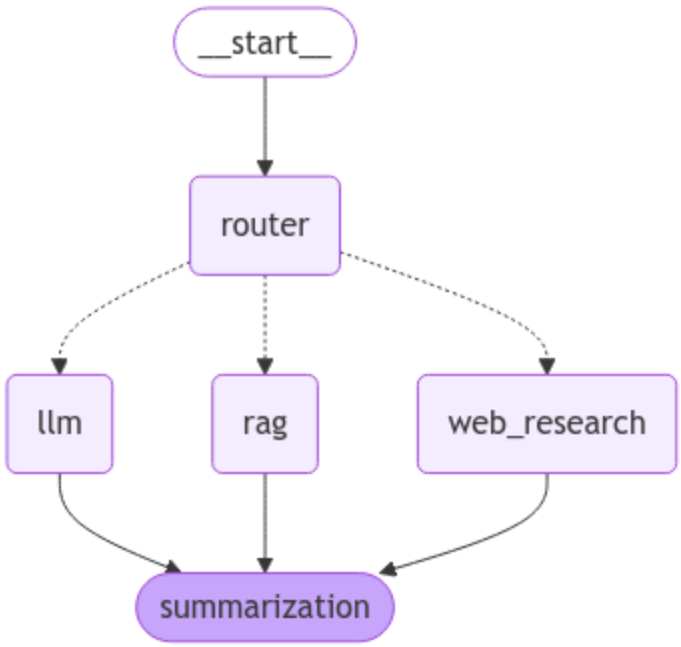
The system should include:

1. **Router Agent** – Determines whether a query should be answered using the LLM, web research, or retrieval-augmented generation (RAG).
2. **Web Research Agent** – If the query requires up-to-date information, this agent performs a web search and extracts relevant details.
3. **RAG Agent** – If the query is related to a predefined dataset (e.g. any dataset that you have)
4. **Summarization Agent** – After gathering information, this agent synthesizes a final structured response.

Requirements:

- **Use an LLM API** Use OpenAI API or any other LLM powerful in Tool Calling.
- **Implement Specialized Functions:**
    - A **Router Agent** to determine how to process each query.
    - A **Web Research Agent** to fetch live information.
    - A **RAG Agent** to retrieve information from a vector database.
    - A **Summarization Agent** to refine and structure the final response.
- **Set Up a LangGraph Workflow:**
    - Implement a **state graph** with multiple nodes representing different agents.
    - Use **conditional edges** to route the query to the appropriate agent based on the type of question. For example: ‘Latest’, ‘current’ in query leads to websearch otherwise RAG.
- **Summarizer:**
    - After the information has been collected, summarize your findings using LLM.
- **Architecture:**

  Router Agent Architecture



Expected Deliverables

1. **Code Submission**
   - A Python script or Jupyter Notebook implementing the agent using LangGraph.
2. **Report**
   - A brief explanation (in markdown) describing how the LLM and different agents are used.
   - Explanation of the **agent workflow and decision-making process**.
   - **Graph structure and flow** of the system.
   - **Examples of test queries** with their results.

Approach:

1. **Define Custom Agents:**
   - Implement a Router Agent, Web Research Agent, RAG Agent, Summarization Agent.
2. **Integrate LLM:**
   - Use LLM for general queries and reasoning tasks.
3. **Create LangGraph Structure:**
   - Set up a state graph with multiple interconnected nodes.

    - Use **conditional routing** to dynamically select the appropriate agent for processing a query. For example: ‘Latest’, ‘current’ in query leads to websearch, otherwise RAG.
4. **Enable Memory for Conversational AI:**
    - Implement a system to remember past interactions for improved responses.
5. **Summarize** the collected information
