Agent Framework for RAG-Based Question Generation and Summarization

The objective of this tech test is to evaluate the candidate's technical skills and approach to developing an agent-based framework for generating questions (MCQs and fill-in-the-blanks) and summarizing key topics from a given PDF file using Large Language Models (LLMs) and RAG using relevant libraries in Python.

Please Do:

- **Include Instructions**: Provide instructions on how to run the code.
- **Keep it Simple**: Ensure the solution is simple and concise.
- **Submit Within Time**: Return the zipped solution within **72 hours** by email to *technicaltest@alefeducation.com* or provide a link to a Google Drive folder with the zipped solution.
- **Ask Questions**: If you have any questions, email *technicaltest@alefeducation.com*.

Please Don't:

- Do not use public source control systems (e.g., GitHub).
- Do not build a user interface; the focus is on the technical solution and not the interface.
- Do not spend more than 2-3 hours on the test; the goal is to understand your approach and way of thinking.
- Perform Unnecessary Tasks: Do not perform exploratory data analysis, model evaluation.
- Cheat: Do not cheat; cheaters will be immediately dismissed and blacklisted by Alef.

Technical Requirements:

1. Design an agent-based framework that integrates with LLMs to generate questions and summaries from an uploaded file.
2. Use libraries like Langchain, LangGraph, Huggingface, FastAPI to facilitate the development.
3. Use openly available LLMs available via APIs from providers like Groq, Google AI, or Huggingface.
4. Use an open-source vector database to store and retrieve relevant information efficiently.
5. Create a Dockerfile with commands to run the application and expose the endpoints.

Endpoints:

1. **/ingest**: Upload and process the PDF file.
2. **/generate/questions**: Generate either MCQ and Fill-in-the-blank type questions based on the PDF.
3. **/generate/summary**: Generate summaries based on a given topic from the uploaded PDF.

Functional Test Conditions:

- Upload a sample PDF file "**A quick Algebra Review**" and ingest into a Vector DB with appropriate metadata.
- Generate MCQs and fill-in-the-blank questions based on the topics in the PDF file
- Generate summaries for any specified topics.

Assessment Criteria:

- Should cover all Technical requirements.
- Should cover the functional test case.
- Demonstrate clean coding and test driven development.
- Should be organized to demonstrate as a python production code file rather than a .ipynb file used for experimentation.