GenAI Engineer Hiring Task Scenario

Your company is working on building an AI-powered assistant that can help healthcare organizations manage and make sense of large collections of medical documents. These documents may come in different formats such as PDFs, Word files, Excel sheets, and even images containing tables, graphs, or clinical charts.

The system should allow a user to upload their documents into the platform and also make use of the documents provided in the Google Drive. User can then interact with an assistant through a chat interface.

Google Drive: Click Here

Part 1 – Q&A / Conversational Requirements

Build a Q&A service that:

- Answers questions **only using information grounded in the provided documents**, which include:

    1. Documents uploaded by user.
    2. Documents available in the provided Google Drive.

- If an answer cannot be found in any document, the system must explicitly state that the information is not available (no hallucinations).

Citations and References:

For every Answer:

- Return citations specifying:

    1. Which document(s) were used.
    2. Which chunk(s) or portions of the answer map to those documents.

- If the reference is from the **Google Drive**, include **a clickable link** to that specific file in the Drive so that the user can directly open it.

Also, the assistant should maintain **memory across turns**, enabling users to carry on natural multi-turn conversations without repeating context.

Note: Completion of this part is mandatory for submission.

Part 2 – Report Generation Requirement

Beyond just Q&A, the assistant should be capable of generating structured medical reports from the uploaded documents only.

When the user requests a report, they may specify what sections they want (for example, *"Introduction," "Clinical Findings," "Patient Tables," "Graphs," and "Summary"*). To achieve this, the assistant should use **LLM function calling** to trigger the appropriate tools for:

- Extracting exact data from the relevant sections of the medical documents.
- Pulling tables, charts, or clinical figures from PDFs and Word files.
- Inserting those elements into the correct section of the report.

The extracted content should be preserved exactly as it appears in the original documents — no rewriting or paraphrasing — except for sections where the user explicitly asks for a summary. In that case, the assistant should generate a concise summary and append it.

The completed report must then be exported as a downloadable PDF.

Part 3 – Backend and Frontend Requirements

The entire backend should be exposed as an API so that other applications can integrate with it. On top of that, create a simple frontend interface where a user can:

- Upload multiple types of documents.
- Interact with the assistant through a conversational chatbot.
- Request a report and download the generated PDF.

Note: Completion of this part is mandatory for submission.

Bonus Challenge

As an additional challenge, you may, containerize the system using Docker and deploy it to a cloud platform with basic monitoring and logging to make the system robust and maintainable at scale.

Final Deliverables

By the end of this task, candidates are expected to submit:

1. A 2-3-minute video demonstrating and explaining the working prototype.
2. GitHub repository with the complete code, including proper comments and clear documentation.

3. An architecture diagram showing system components and workflow.
4. A description of the tech stack used (programming languages, frameworks, libraries, deployment platform, etc.).