Junior Data Scientist / AI Engineer Take-Home Assignment: Client Document Q&A Pilot

Assignment Goal:

This assignment assesses your ability to understand a client's business need, design a feasible technical solution using modern AI techniques, write clean and well-documented code, and communicate your approach and rationale effectively.

The Business Problem:

One of our clients, a medium-sized enterprise, is struggling with information accessibility. Their employees spend significant time searching through various internal documents (product specifications, HR policies, project reports, etc.) to find answers to common questions. This inefficiency impacts productivity.

The client has requested a **proof-of-concept (PoC)** for a simple internal tool. They envision a system where an employee can ask a question in natural language and receive a concise answer derived *only* from a specific, curated set of company documents. They want to evaluate the feasibility and potential value of using modern AI for this task before committing to a larger project.

Your Task: Build the PoC

Your task is to build this PoC Question-Answering (QA) system. You will be provided with a small sample set of the client's documents (3-5 short .txt files).

Your system should:

1. **Process Documents:** Ingest and prepare the provided text documents for querying.
2. **Answer Questions:** Accept a user's question as input.
3. **Generate Grounded Answers:** Return an answer synthesized *solely* from the information present in the provided documents. The system must avoid making assumptions or providing information external to the supplied texts.

Key Considerations (What the client cares about):

- **Accuracy:** Answers must accurately reflect the information in the documents.
- **Relevance:** The system should identify the most relevant parts of the documents to answer the question.

- **Clarity:** Answers should be clear and concise.
- **Justification:** As this is a PoC, the client wants to understand *how* it works and *why* specific technical choices were made.

Provided Materials:

- A zip file containing 3-5 sample .txt documents representing typical client materials.
- (Optional, upon request) If your chosen solution involves external APIs requiring keys/credits (e.g., OpenAI), we may be able to provide limited access. Please state this need clearly in your submission plan if applicable.

Deliverables:

1. **Working PoC:** Submit your complete, runnable code (e.g., Python scripts, Jupyter Notebook). Ensure it is well-commented, explaining the logic and key components.
2. **README.md:** A Markdown file containing:
    - **Problem Interpretation:** Briefly restate the client's problem and the goal of the PoC.
    - **Proposed Solution & Rationale:** Describe your chosen technical approach (e.g., document processing strategy, core AI/ML techniques used for retrieval and answer generation). Crucially, explain *why* you selected this approach and these specific tools/libraries, considering the client's needs (accuracy, relevance, etc.). Discuss any trade-offs.
    - **Setup Instructions:** Clear steps on how to set up the environment (e.g., install dependencies using requirements.txt or environment.yml) and run your PoC. Include instructions on how to handle any necessary API keys or credentials.
    - **How to Use:** Simple instructions on how to interact with your PoC to ask a question.
    - **Limitations & Next Steps:** Briefly discuss any limitations of your PoC and suggest potential next steps or improvements if this were to move beyond the pilot phase (e.g., scalability, handling more document types, improving answer quality).

Evaluation Criteria:

- **Problem Solving:** Does the PoC effectively address the client's core problem of answering questions based *only* on the provided documents?
- **Technical Design & Justification:** Is the chosen technical approach sound, modern, and well-suited to the task? Is the rationale for technical choices clear and well-argued?

- **Code Quality & Implementation:** Is the code clean, well-organized, readable, commented, and functional?
- **Communication:** Is the README clear, comprehensive, and does it effectively communicate the solution, rationale, and usage?
- **Reproducibility:** Can we easily set up and run your PoC following your instructions?

Time Estimate:

We estimate this task should take approximately 4-8 hours of focused work. Please prioritize a clean, functional PoC and clear justification over adding excessive features.

Submission:

Please package your code and README file into a single zip archive or provide a link to a Git repository (e.g., GitHub, GitLab).

We look forward to seeing your approach to solving this client problem! Good luck!