**Interview Assignment: AI-Powered FAQ Assistant**

**Objective**

Build a simple FAQ assistant using a Generative AI (GenAI) model integrated with Python. The assistant should answer user questions based on a provided knowledge base of FAQs using retrieval and LLM-based response generation.

**Assignment Brief**

**Task Description**

You are given a small knowledge base of FAQs (questions and answers). Your task is to:

1. Ingest the FAQ knowledge base.
2. Implement a retrieval method to select the most relevant FAQ(s) based on a user query.
3. Use a GenAI model (OpenAI, Hugging Face, or equivalent) to generate a natural language answer using the retrieved FAQ context.
4. Build a Python-based API or app to demonstrate the assistant’s functionality.

**Details**

* Use Python with appropriate libraries (e.g., FastAPI, Flask, Streamlit).
* The knowledge base file (faqs.csv) contains two columns: *Question* and *Answer*.
* Implement either keyword-based or embedding-based similarity search to find relevant FAQs.
* Design a prompt template to pass context + question to the GenAI API.
* Return the generated response to the user query.

**Deliverables**

* Fully documented Python code in a public/private GitHub repo with clear instructions.
* A working demo (e.g., API endpoint /ask or a Streamlit app).
* A README with setup, usage instructions, and explanation of design.
* Optional: Screenshots or short video demo of example Q&A runs.

**Sample FAQ Data (faqs.csv)**

|  |  |
| --- | --- |
| Question | Answer |
| What are your working hours? | Our working hours are 9 AM to 6 PM, Monday to Friday. |
| How can I reset my password? | You can reset your password using the ‘Forgot Password’ option on the login page. |
| What is the refund policy? | Refunds can be requested within 30 days of purchase with a valid receipt. |
| Do you offer customer support? | Yes, customer support is available 24/7 via phone, email, and chat. |

**Suggested Steps**

1. Load the CSV file and preprocess it.
2. Implement a similarity search (keyword matching or embeddings).
3. Use a GenAI model API for answer generation.
4. Create an interactive demo where user inputs a question and gets a generated response.
5. Package code cleanly with comments and instructions.

**Submission Instructions**

1. Push your complete solution to a GitHub repo.
2. Include all necessary files (requirements.txt, README, code).
3. Share the repo URL during the interview.
4. Be prepared to demo your app and explain your approach.