AI Engineer Challenge – Customer Support AI Assistant

📌 Objective

Your task is to develop an **AI-powered customer support assistant** that enhances responses using a **foundational LLM** and **retrieval-augmented generation (RAG)**.

This challenge will evaluate your ability to:
✅ Implement **RAG** with a **vector database** (e.g. FAISS/ChromaDB).
✅ Integrate an **LLM-based response system**.
✅ Ensure **explainability** of model decisions.
✅ Deploy a simple **API** for real-world usability.

📂 Dataset

We will use a **public customer support dataset**:
📌 Customer Support on Twitter

This dataset contains customer queries and support responses. You can use this for **retrieval-based response generation**.

🔹 Challenge Tasks

1️⃣ Implement RAG-Based AI Assistant *(4 hours)*

✅ Use a **foundation model** (e.g., Mistral-7B, Llama-2, Falcon).
✅ Implement **retrieval-augmented generation (RAG)**:

- Retrieve relevant past queries & responses.
- Use LLM to generate a response to a new support query.

2️⃣ Ensure Explainability & Evaluation (2 hours)

✅ **(Preferred)** Define an evaluation approach to assess response quality, correctness, and relevance.

✅ **(Optional)** Implement basic explainability by highlighting retrieved documents or reasoning behind responses.
✅ Be prepared to discuss explainability and evaluation strategies during the live demo.

3️⃣ API Deployment *(2-4 hours)*

✅ Develop a REST API (`/generate_response`) using **FastAPI/Flask**.
✅ Ensure it accepts **user queries** and returns AI-generated responses.
✅ Store responses in a mock database (JSON/SQLite) to log interactions for future reference. Alternatively, you can simply print the responses or save them in a CSV file - whatever works best for the purpose.

📥 Submission Requirements

You must submit the following:
✅ **GitHub/Code repository** with clean, modular code.
✅ **API endpoint (if deployed)** for live testing.
✅ **PPT (Template shared)/Short README** explaining:

- Approach taken
- RAG implementation details
- Explainability techniques
  ✅ *(Optional Bonus)* 2-minute video demo.

🏆 Evaluation Criteria

Category	Weight (%)	Evaluation Focus
RAG Implementation	**40%**	Retrieval accuracy, LLM enhancement
Explainability	**20%**	Reasoning
API Deployment	**20%**	Usability, efficiency
Code Quality	**10%**	Clean, structured code
Innovation	**10%**	Enhancements (multi-turn, caching, integrations)

📌 Example Queries for Testing

Use the following queries to test your model:

1️⃣ **"I ordered a laptop, but it arrived with a broken screen. What should I do?"**
2️⃣ **"I need help resetting my password."** (Follow-up) **"I didn't receive the reset link."**
3️⃣ **"My cat chewed my phone charger. Is this covered under warranty?"**
4️⃣ **"Why did you suggest contacting support?"** *(Checks explainability!)*

⏳ Time Duration & Deadline

📅 **You have 1 week to complete this challenge.**
⏳ Submission Deadline: **[Insert Date - 7 days from today]**
📩 Share your **GitHub/Code repo + API link**(optional) with us via **email**.

✅ **Final Report/PPT:** Candidates must submit a summary report/ppt **24 hours before the panel presentation**.

We look forward to seeing your innovative solutions! 🚀