Assignment: Build a Retrieval-Augmented Generation (RAG) Application using Open-Source Tools

You are required to build a **Knowledge Assistant** that answers user questions strictly based on the documents provided. The system should demonstrate a complete RAG pipeline, including document ingestion, embedding generation, vector search, prompt augmentation, and response generation, using **open-source frameworks and models**.

Requirements

- Use open-source tools/models for:
    - Text embeddings (e.g., Sentence Transformers, BGE, E5, Instructor)
    - LLM inference (e.g., LLaMA, Mistral, Mixtral, Phi, Gemma)
- Ingest 5–20 documents (PDF, TXT, or Markdown)
- Implement document chunking and embedding
- Store embeddings in a vector database:
    - FAISS, Chroma, Weaviate, Qdrant, Milvus, or similar
- Implement complete RAG flow:
    - User query embedding
    - Vector similarity search
    - Context retrieval
    - Prompt augmentation
    - Response generation
- Implement prompt engineering with clear system instructions to prevent hallucinations
- Include basic guardrails or safety controls, such as:
    - Output filtering
    - Confidence thresholding
    - “Answer only from context” enforcement
- Provide one interaction method (choose any one):
    - REST API
    - Command-line interface (CLI)
    - Simple UI (Streamlit / web UI)
- Responses must include source document references

Expected Outcome

The final solution should demonstrate a working, end-to-end RAG system using open-source tools, with grounded answers, clean design, and the ability to explain architectural and model choices.