AI INTEGRATED CHATBOT

Overall Approach :

The main goal of this project is to create a chatbot that can embed a PDF document and answer questions based on the document's content using Streamlit and various AI tools.

- **Document Loading**: Load the PDF document using **`PyPDFLoader`**.
- **Text Splitting**: Split the document into manageable chunks using **`RecursiveCharacterTextSplitter`**.
- **Embedding:** Embed the text chunks using HuggingFace embeddings.
- **Vector Store**: Store the embedded vectors using **FAISS**.
- **LLM Integration**: Use **`ChatGroq`** for question-answering based on the embedded document.
- **Conversation Context**: Maintain a history of user interactions to provide context-aware responses.

Frameworks/Libraries/Tools Used :

- **Streamlit**: For building an interactive web application.
  - Used to create the UI and handle user interactions.

- **Langchain:** Various modules from Langchain for document processing and LLM integration.
  - **`langchain_groq`:** Used for integrating the ChatGroq model.
  - **`langchain_community.embeddings`:** Used for HuggingFace embeddings.
  - **`langchain.text_splitter`**: Used for splitting documents into chunks.
  - **`langchain.chains`**: Used for creating the retrieval chain.
  - **`langchain_community.vectorstores`**: Used for storing embeddings in FAISS.
  - **`langchain_community.document_loaders`**: Used for loading PDF documents.
- **dotenv**: For loading environment variables from a `.env` file.
  - Used to securely manage API keys.

Problems Faced and Solutions:

- **Environment Variable Management**:
  - Problem: Missing or incorrect API keys.
  - Solution: Implemented checks and informative error messages if API keys are not set correctly.
- **Document Loading Issues**:
  - Problem: PDF file not found or unreadable.
  - Solution: Added checks for file existence and readability, with appropriate error handling.

- **Embedding and Text Splitting**:
  - Problem: No text chunks created due to non-extractable text.
  - Solution: Added validation checks to ensure chunks are created successfully.
- **Response Handling**:
  - Problem: Errors in response generation.
  - Solution: Enhanced error handling and context management to improve response accuracy.

Future Scope :

- **Enhanced Context Management**:
  - Implement advanced methods for managing and retrieving conversation context to improve coherence in responses.
- **Additional Features**:
  - Add capabilities for handling multiple file types (e.g., DOCX, TXT).
  - Implement user authentication to personalise interactions and context management.
  - Integrate additional AI models for more robust and varied responses.
- **User Interface Improvements**:
  - Enhance the UI with more interactive elements and better visual feedback.
  - Provide options for users to upload their documents for embedding and questioning.
- **Scalability**:
  - Optimise the application for deployment on cloud platforms for handling larger user bases.
- **Integration with Other Services**:
  - Integrate with external knowledge bases and APIs to provide more comprehensive answers.