Conversational PDF Assistant with RAG (Retrieval-Augmented Generation)

This is an interactive PDF assistant built with **Streamlit**, designed to help users engage with and extract key information from long PDF documents (including 230+ pages). The app provides a conversational AI interface that offers:

- **Document Summarization**: Automatically generates a concise summary of the uploaded PDF.
- **Suggested Questions**: Displays a list of auto-generated, relevant questions to help users explore the document.
- **Conversational Q&A**: Enables users to ask questions about the document's content with context-aware responses.

Features

1. PDF Upload and Processing

- Upload large PDF files (including documents with 230+ pages).
- PDFs are processed, split, and indexed to make them queryable.
- Generates embeddings using HuggingFace models (`all-MiniLM-L6-v2`).

2. Automatic Document Summary

- A brief summary (5–10 key points) is generated from the content of the uploaded PDF to give users an overview of the document.

3. Auto-Generated Suggested Questions

- The app generates **5-10 starter questions** based on the content of the document, guiding users in their exploration of the material.

4. Conversational Q&A (RAG)

- Users can engage in a **Q&A chat** with the assistant, which retrieves relevant context from the document to answer questions.
- The assistant responds to queries contextually and remembers the chat history for ongoing sessions.

5. User-Friendly Interface

- **Streamlit** is used to build a clean, intuitive interface.
- **Sections** are clearly separated for easy navigation:
  - PDF Upload
  - Document Summary
  - Suggested Questions
  - Chatbot Interaction

Getting Started

1. Clone the Repository

```
git clone https://github.com/yourusername/conversational-pdf-assistant.git
cd conversational-pdf-assistant
```

2. Install Required Libraries

```
pip install -r requirements.txt
```

3. Set Up Environment Variables

Create a `.env` file in the root directory and add the following environment variables:

```
HF_TOKEN=your_huggingface_api_key
GROQ_API_KEY=your_groq_api_key
```

- You can get the HuggingFace API key from: Hugging Face

- You can get the Groq API key from: Groq Cloud

4. Run the Streamlit App

```
streamlit run app.py
```

This will open a new tab in your browser with the interactive app.

UI Overview

Sections:

- **PDF Upload Section**: Upload your PDFs here.
- **Summary Section**: Displays a concise summary of the uploaded document (5-10 key points).
- **Suggested Questions**: Provides auto-generated questions based on the document to get you started.
- **Q&A Section**: Interactively ask questions related to the document and receive context-based responses.

Tech Stack

Component	Library/Tool
LLM (Language Model)	Groq API (Gemma2-9b It)
Embeddings	HuggingFace Transformers
Vector Database	Chroma
Text Splitter	LangChain RecursiveSplitter
UI Framework	Streamlit
Chat History	LangChain ChatHistory

Notes

- The app works efficiently with **large PDFs** (including documents over 230 pages).
- The document is **split into chunks** for efficient querying and context retrieval.
- The app **remembers the chat history** across sessions for an enhanced conversational experience.
- **Works best with text-based PDFs**, not scanned documents (OCR is not included).

Contact

If you have any questions or need additional features, feel free to reach out!

- **Developer**: Sahil Chandel
- **Phone Number**9717891203
- **Email**: sahilchandel.anee@gmail.com
- **GitHub**: Sahil-Chandel GitHub

This app helps you interactively analyze and extract information from PDFs using the latest advancements in AI and document processing. It's perfect for handling large documents in research, business reports, and any scenario where quick information extraction is key.