PDF CHATBOT – Pratyush Kumar

Chatbot: Development Overview

Overall Approach

Corpus Integration: Extracted text from a PDF document to create a knowledge base.

AI-Powered Responses: Utilized Google's Gemini AI model for generating context-aware responses.

Conversation Management: Implemented a system to maintain conversation history and provide coherent follow-up responses.

User Interface: Created an interactive chat interface using Streamlit for easy user interaction.

Frameworks, Libraries, and Tools Used

Streamlit: Used for creating the web-based user interface and managing app state.

PyMuPDF (fitz): Employed for extracting text from the PDF corpus.

Google GenerativeAI: Integrated for powering the AI responses using the Gemini model.

Asyncio: Utilized for handling asynchronous operations in response generation.

Challenges Faced and Solutions

1) **PDF Text Extraction:**

Challenge: Ensuring accurate and formatted text extraction from PDF.

Solution: Used PyMuPDF for its robust PDF parsing capabilities.

2) **Maintaining Context:**

Challenge: Enabling the chatbot to understand and refer to previous conversation points.

Solution: Implemented a conversation history system using Streamlit's session state.

3) **Handling Out-of-Corpus Queries:**

Challenge: Responding appropriately to questions not covered in the knowledge base.

Solution: Added logic to suggest contacting Jessup Cellars directly for such queries.

4) **Asynchronous Operations:**

Challenge: Managing potentially slow AI response generation without blocking the UI.

Solution: Implemented asynchronous functions with asyncio to keep the interface responsive.

5) **Balancing Information Retention and Performance:**

Challenge: Keeping enough conversation history without overloading the system.

Solution: Implemented pruning of chat history, retaining only the last 10-20 exchanges.