1. Overall Approach

In this project, we set out to create a user-friendly chatbot with a sleek and minimalistic interface. Our main goal was to build a chatbot that could effectively respond to user queries based on the content of a PDF file. We aimed to achieve:

- **Intuitive Interaction**: A simple and clean user interface where users can upload a PDF and chat with the bot.
- **Context-Aware Responses**: The chatbot should remember and utilize past conversation history to handle follow-up questions.
- **Optimized Performance**: Fast response times, ideally within 2-3 seconds, to ensure a smooth user experience.
- **Fallback for Out-of-Corpus Questions**: If the bot couldn't answer based on the PDF content, it would guide users to contact the business directly.

We used Streamlit for the UI, PyPDF2 for extracting text from PDFs, and Ollama's Generative AI model for generating responses. Here's how it all came together.

2. Tools and Libraries

Streamlit

- **What It Is**: A powerful tool for building interactive web apps with Python.
- **How We Used It**: Created a clean, minimalistic chat interface where users can upload PDFs and interact with the chatbot.

PyPDF2

- **What It Is**: A Python library for working with PDF files.
- **How We Used It**: Extracted text from uploaded PDF files so the chatbot could use this information to generate accurate responses.

Ollama

- **What It Is**: A Generative AI model provider.
- **How We Used It**: Utilized the `gemma:2b` model to generate intelligent and relevant responses based on the context provided by the PDF and conversation history.

3. Challenges and Solutions

Extracting Text from PDFs

- **Challenge**: PDF files can have complex layouts that make text extraction tricky.
- **Solution**: We used PyPDF2 for basic extraction. For more complicated PDFs, future enhancements might include advanced text extraction techniques or OCR.

Maintaining Conversation Context

- **Challenge**: Keeping track of ongoing conversations and understanding context from previous messages.
- **Solution**: Implemented session-based storage to keep track of conversation history and added context from the PDF to each user query.

Ensuring Fast Response Times

- **Challenge**: Achieving response times within the 2-3 seconds limit.
- **Solution**: Optimized token usage and employed streaming responses to minimize latency.(latency can be reduced even more with gpu utilisation)

Handling Unanswered Questions

- **Challenge**: Managing questions that are not covered by the PDF.
- **Solution**: Added a fallback response that directs users to contact the business if the chatbot cannot provide an answer.

4. Future Enhancements

Improved PDF Processing

- **What We Can Do**: Incorporate advanced PDF processing techniques or OCR to better handle complex PDFs.

Integration with Additional Data Sources

- **What We Can Do**: Connect the chatbot to other data sources or databases for richer and more comprehensive answers.

Enhanced NLP Capabilities

- **What We Can Do**: Use more advanced NLP models to improve the chatbot's ability to understand and respond to nuanced queries.

Personalized User Experience

- **What We Can Do**: Implement user profiles to tailor responses based on individual preferences and past interactions.

Multi-Language Support

- **What We Can Do**: Expand the chatbot's capabilities to support multiple languages, making it accessible to a broader audience.

Analytics and Monitoring

- **What We Can Do**: Add features to track performance and user interactions, helping us continuously improve the chatbot's efficiency and effectiveness.