Chatbot Project Plan

Step 1: Setup Environment

- Install the required Python libraries:
  - **PyPDF2:** To extract text from PDF files.
  - **transformers:** To leverage a pre-trained NLP model for question answering.
  - **streamlit:** To create a simple web-based interface for the chatbot.

    Run this command to install the necessary libraries:

    pip install streamlit PyPDF2 transformers

Step 2: PDF Text Extraction

- Create a function to extract text from the uploaded PDF.
  - Use PyPDF2 to read the PDF file page by page.
  - Concatenate the extracted text from all pages into a single string.

Step 3: Question-Answering Functionality

- Create a function that uses a pre-trained question-answering model from Hugging Face's transformers library.
  - Input: The extracted text from the PDF and the user's question.
  - Output: The answer from the relevant section of the text.

Step 4: Build the Streamlit Web Interface

- Use Streamlit to create a user-friendly web interface.
  - Allow users to upload a PDF file using st.file_uploader().
  - Display a text input for users to type their question.
  - Display the PDF text extraction result and allow users to ask questions.

Step 5: Integrate Everything

- Combine the PDF extraction and question-answering components into the Streamlit app.
  - Once the user uploads the PDF, extract the text and store it.
  - When a user submits a question, the chatbot will process the question and provide an answer based on the extracted text.

Step 6: Run and Test the App

- Run the app using the Streamlit command:

  streamlit run your_file_name.py

- Upload different PDFs and ask questions to ensure the chatbot accurately answers based on the content of the PDF.

Step 7: Expand and Optimize (Optional)

- Add error handling for unsupported PDFs or empty text.
- Add caching to avoid re-extracting text from the PDF for every question.
- Optionally, fine-tune the question-answering model for domain-specific PDFs (legal documents, academic papers, etc.).