Project Title: Chat with PDF using Streamlit and LLM

Objective:
The objective of this project is to create a Streamlit web application that allows users to upload PDF documents and engage in a conversational interface to query the content of those PDFs using large language models.

Components:

1. **Streamlit Interface:**
   - The Streamlit interface provides the user with options to upload PDF documents and enter queries.
   - It includes a sidebar for inputting the OpenAI API key and displays uploaded documents.
   - The user can input queries about the documents, and the system generates responses based on the content.
2. **PDF Processing:**
   - The `get_text` function extracts text from uploaded PDF documents using `PdfReader`.
   - The extracted text is preprocessed into chunks using a `CharacterTextSplitter`.
   - These text chunks are then converted into embeddings and stored in a vector database using `OpenAIEmbeddings` and `FAISS`.
3. **Conversation Handling:**
   - A conversational chain is established using OpenAI's language model (`ChatOpenAI`) and the vector database.
   - Conversation history is managed using a `ConversationBufferMemory`.
4. **User Interaction:**
   - User queries are handled by the `handle_userinput` function, which retrieves responses based on the input prompt and updates the conversation history.
   - The UI provides feedback to the user during processing, such as displaying a spinner.

Improvements:

1. **Error Handling:** Add robust error handling for PDF extraction failures and missing API keys.
2. **State Management:** Ensure proper management of session state for seamless user experience.
3. **Security:** Implement secure handling of API keys to prevent exposure.
4. **Efficiency:** Consider optimizing PDF processing for large documents to enhance performance.
5. **UI Enhancement:** Enhance the UI with better styling and user feedback elements.