1. **How do you deploy this ChatBot in production?**
   Deploying this ChatBot in production involves several steps to ensure it is robust, scalable, and maintainable. Here's a concise overview of the process:
   1. Containerization: Use Docker to create a container with the app and all dependencies.
   2. Deployment: Deploy the Docker container to a cloud platform (AWS, Google Cloud, Azure).
   3. Use orchestration tools like Kubernetes or cloud-native services to manage and scale the deployment.
   4. Monitoring and Logging:Implement monitoring to track performance and usage.
   5. Use logging tools to capture and analyze logs for troubleshooting.
   6. Automated Testing and CI/CD: Set up automated tests to ensure functionality.
   7. Establish a CI/CD pipeline to automate building, testing, and deploying new versions.
   8. Security: Implement security best practices such as SSL, authentication, and authorization to protect the application.

   2. Diagram of the implemented solution
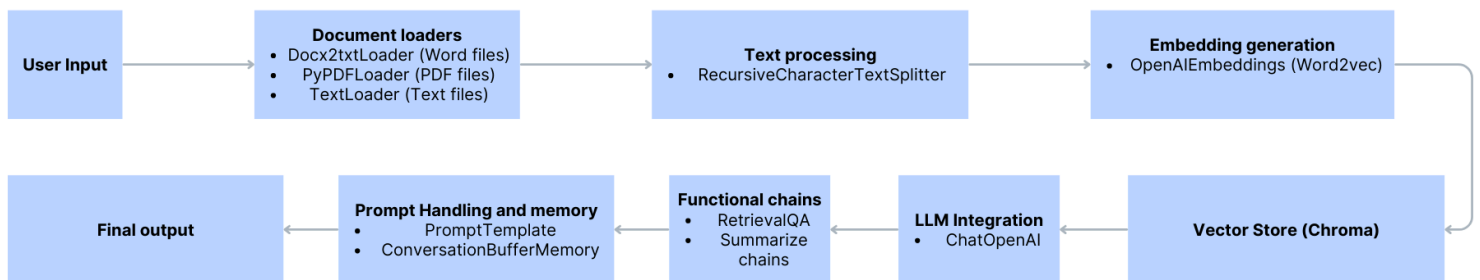
Explanation

1. **User Input**: The starting point where user queries or documents are received.
2. **Document Loaders**: Different loaders are used to handle various document types (Word, PDF, text).
3. **Text Processing**: The text is split into manageable chunks using RecursiveCharacterTextSplitter.
4. **Embedding Generation**: Text is converted into embeddings using `OpenAIEmbeddings`.
5. **Vector Store (Chroma)**: The embeddings are stored in a vector store for efficient retrieval.
6. **LLM Integration**: The chatbot uses a language model from OpenAI (`ChatOpenAI`) for understanding and generating responses.
7. **Functional Chains**: Chains are built for specific tasks like question-answering (`RetrievalQA`) and summarization.
8. **OCR & PDF to Image**: OCR is performed on images (extracted from PDFs) using Tesseract, and PDFs are converted to images using `pdf2image`.

9. **Prompt Handling & Memory**: Prompts are managed using `PromptTemplate`, and conversation history is maintained with `ConversationBufferMemory`.
10. **Final Output**: The response or result is generated and provided to the user.

Example of use:

Questions:

```
questions=["¿Que colonias tienes de disney",
           "¿Cual es la religión más practican en colombia?",
           "¿Hay alguna colonia o producto que sea dermatológicamente probado?",
           "Que venden en la última página del pdf"]
```

Answers:

```
['Tenemos disponibles las colonias Disney Minnie Mouse y Disney Princesa en presentaciones de 150 ml cada una. La Colonia Minnie Mouse tiene un precio regular de $18.900 y la Colonia Disney Princesa
tiene un precio regular de $25.900. Ambas colonias son dermatológicamente probadas y están libres de parabenos.',
 'Lo siento, pero no tengo información sobre la religión más practicada en Colombia en el contenido del PDF proporcionado. ¿Tienes alguna otra pregunta relacionada con el archivo?',
 'Sí, la Colonia Minnie Mouse de Disney y la Colonia Barbie Spray Desenredante Barbie son productos que han sido dermatológicamente probados, según se indica en el archivo PDF proporcionado.',
 'En la última página del PDF se encuentra la oferta de un gel para el cabello Spider-Man de Marvel de 100 g, con un precio regular de $24.990 y un descuento especial a $13.900. Este producto ha sido
dermatológicamente probado y está basado en estudios técnicos realizados.']
```