**Challenges and Solutions in the AskDocs Project**

### 1. Efficient PDF Processing

**Challenge:**

* Extracting text from PDFs while handling different formats and structures.
* Managing errors when loading corrupted or encrypted PDFs.

**Solution:**

* Used UnstructuredPDFLoader for robust text extraction.
* Implemented exception handling to catch and log errors during PDF loading.

### 2. Chunking Strategy

**Challenge:**

* Balancing chunk size for better retrieval accuracy without losing contextual meaning.
* Preventing overlapping information loss during chunking.

**Solution:**

* Used RecursiveCharacterTextSplitter with a **7500-character chunk size** and **100-character overlap** to maintain coherence.

### 3. Vector Database Management

**Challenge:**

* Ensuring fast and accurate search in the vector database.
* Managing document embedding storage efficiently.

**Solution:**

* Integrated **ChromaDB** with **Ollama embeddings (nomic-embed-text)** for optimized retrieval.
* Implemented methods for creating and deleting collections dynamically.

### 4. Multi-Query Generation

**Challenge:**

* Overcoming limitations of simple keyword-based retrieval.
* Improving the accuracy of document search.

**Solution:**

* Implemented **MultiQueryRetriever** to generate diverse question variations, enhancing retrieval relevance.

### 5. Ollama Model Performance

**Challenge:**

* Generating contextually relevant answers from retrieved document chunks.
* Adapting different Ollama models for optimal performance.

**Solution:**

* Designed a **custom prompt template** to refine query reformulation.
* Provided users the ability to **select different Ollama models** for experimentation.

### 6. User Interface Optimization

**Challenge:**

* Creating an intuitive and user-friendly UI for document interaction and Q&A.
* Enabling better document visualization in Streamlit.

**Solution:**

* Developed a **split-panel UI** in Streamlit with a **PDF viewer and chat interface**.
* Added **zoom controls** in pdf\_viewer.py for improved readability.

### 7. Session State Management

**Challenge:**

* Retaining chat history, document selections, and vector database states across interactions.

**Solution:**

* Utilized **Streamlit’s session state** to store:
  + Chat message history
  + Vector database instance
  + PDF document selection and page rendering

### 8. Deployment Issues

**Challenge:**

* Ensuring smooth setup and compatibility across different systems.
* Handling missing dependencies and ensuring Ollama models run correctly.

**Solution:**

* Created a **requirements.txt** for simplified dependency installation.
* Verified **Ollama was running locally** before application startup.