**Name-Vidwesh Choudhary**

**1. Installing Required Libraries**

**Purpose**: Install essential Python libraries like langchain, transformers, faiss-cpu, and gradio for text processing, embeddings, vector search, and user interface development.

**2. Extracting Text from PDFs**

**Steps**:

-Used pdfplumber to read and extract text from PDF documents.

-Listed all files in a folder to process multiple PDFs.

**Purpose**: Extract raw text from legal documents to create a searchable database.

**3. Preprocessing Text**

**Steps**:

-Cleaned the text by removing special characters and extra spaces using regex.

**Purpose**: Improve the quality of data for better embeddings and retrieval accuracy.

**4. Splitting Text into Chunks**

**Steps**:

-Used CharacterTextSplitter to divide the text into smaller, overlapping chunks (e.g., 1000 characters with a 100-character overlap).

**Purpose**: Handle large documents effectively, ensuring context retention in retrieval.

**5. Creating Embeddings**

**Steps**:

-Generated embeddings for text chunks using the HuggingFace model (sentence-transformers/all-mpnet-base-v2).

**Purpose**: Convert text into numerical vectors for efficient similarity search.

**6. Building a Vector Store**

**Steps**:

-Stored embeddings in a FAISS (Facebook AI Similarity Search) index for fast retrieval.

**-Purpose**: Enable quick and efficient searching of relevant document chunks.

**7. Querying and Document Retrieval**

**Steps**:

-Queried the FAISS index to retrieve the top relevant text chunks for a user’s query.

**Purpose**: Identify relevant legal text efficiently based on the input query.

**8. Generating Answers**

**Steps**:

-Used a pre-trained T5 model to generate answers from the retrieved context.

**Purpose**: Provide concise and accurate answers based on the legal documents.

**9. Developing a User Interface**

**Steps**:

-Created a Gradio interface to allow users to input queries and view responses interactively.

**Purpose**: Build an accessible front-end for end-users to interact with the chatbot.

**Why These Steps?**

**Data Handling**: Extract and preprocess text for better model performance.

**Embedding & Retrieval**: Use vector stores like FAISS to ensure fast, scalable, and relevant document retrieval.

**Answer Generation**: Leverage NLP models to transform retrieved text into user-friendly answers.

**Interface**: Provide a user-friendly interface for legal professionals or users to interact with the chatbot easily.