Source code Documentation

1. System Overview

This project is a web-based PDF Q&A application that allows users to upload a PDF, ask questions about its content, and receive relevant responses based on embeddings and vector storage. The system uses FastAPI as the backend API framework, and a React frontend for user interactions.

2. Architecture Components

1.1 Backend - FastAPI

The backend handles PDF processing, vector embedding, and question-answering functionalities. Key components include:

- File Upload Endpoint (/upload/): This endpoint accepts a PDF file, extracts text, and generates a vector store using FAISS (Facebook AI Similarity Search) for text retrieval.
- Ask Question Endpoint (/ask/): This endpoint receives a question and uses the generated vector store to retrieve the most relevant answer from the PDF text.

1.2 Frontend - React

The frontend interface includes components for file uploading, question input, and displaying Q&A responses. Key frontend components include:

- FileUpload Component: This handles PDF file selection and upload to the FastAPI server.
- App Component: Manages user interactions like entering questions, sending requests to the backend, and displaying responses.

3. Detailed Component Interactions

Backend Interactions

- PDF Processing and Vector Creation:
  1. PDF is uploaded to /upload/, where extract_text_from_pdf() parses the file text.
  2. Text is split into manageable chunks using CharacterTextSplitter.
  3. process_pdf_and_create_vector_store() generates embeddings and stores them in FAISS, a vector similarity library, for efficient information retrieval.
- Question Processing:
  1. When a question is sent to /ask/, the stored vector database retrieves relevant content using embedding similarity.
  2. The response includes the top relevant text from the PDF.

Frontend Interactions

- File Upload: FileUpload component triggers a POST request to /upload/ with the PDF file, initializing processing and vector storage.
- Question and Answer Display: The user enters a question, which triggers the handleAsk() function to send a POST request to /ask/, receiving and displaying an answer.

4. Key Files and Functions

Backend - FastAPI (Python)

- extract_text_from_pdf(): Extracts all text from a PDF file using PyMuPDF.
- process_pdf_and_create_vector_store(): Chunks PDF text and builds a vector database using FAISS for retrieval.
- upload_pdf() Endpoint: Manages PDF file upload and vector database creation.
- ask_pdf_question() Endpoint: Handles question requests, retrieving the answer from the vector store.

Frontend - React (JavaScript)

- FileUpload Component: Uploads the PDF and initiates backend processing.
- App Component: Manages Q&A interactions, storing each question-answer pair for display.
- handleAsk(): Sends the user’s question to the backend and retrieves the response.

5. Data Flow

1. Upload Phase:
   - PDF is uploaded and processed on the server.
   - Vector embeddings are created, allowing semantic search in the document.
2. Question-Answer Phase:
   - User enters a question, which is sent to the server.
   - The server retrieves the answer by finding relevant sections in the PDF text.

6. Tech Stack

- Backend: FastAPI, PyMuPDF, FAISS, LangChain (for embeddings), HuggingFace embeddings.
- Frontend: React with Material-UI, Axios for API requests.
- Database: In-memory vector storage with FAISS for similarity search.