**Build a QnA Chatbot over Custom PDFs**

Resume-based Job Recommender

1. Objective

The objective of this task is to build a PDF-based Question Answering system using LangChain and OpenAI.

The system should:

* Read text from a PDF file.
* Break the text into manageable chunks.
* Generate vector embeddings for each chunk using OpenAI.
* Store these embeddings in a FAISS vector database.
* (Optional for future) Use these vectors to answer user queries based on PDF content.

2. Tools & Technologies

The following tools and technologies were used to build the PDF QnA chatbot:

* Python – Programming language to build the solution
* VS Code – Code editor used for development
* LangChain – To manage text chunking and retrieval
* OpenAI Embeddings – For converting text into vector embeddings
* FAISS – In-memory vector database for efficient semantic search
* PyPDF2 – To extract text from PDF files
* OpenAI GPT-3.5 – To generate natural language answers from retrieved content

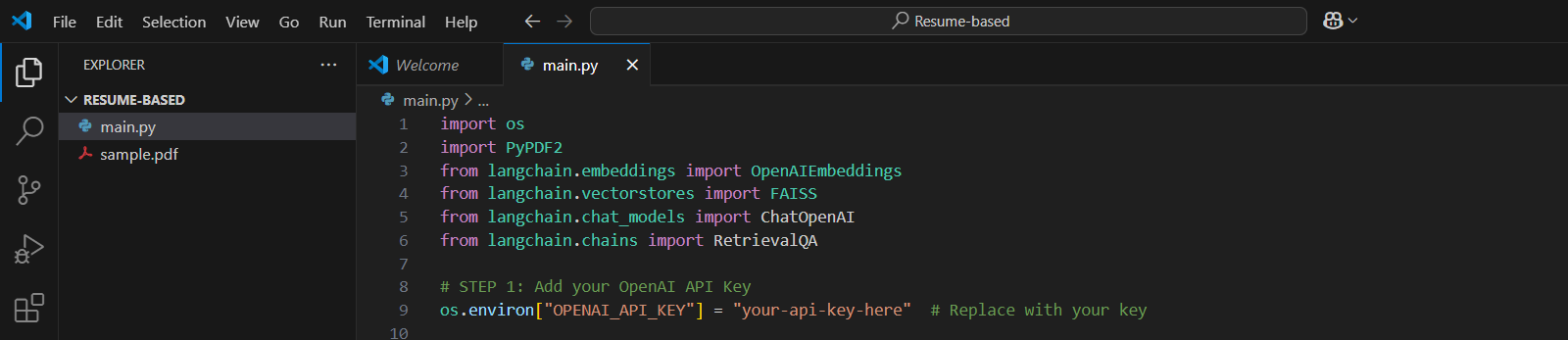
3. Step-by-Step Process

Step 1: Setup Environment

Install Python and necessary libraries:

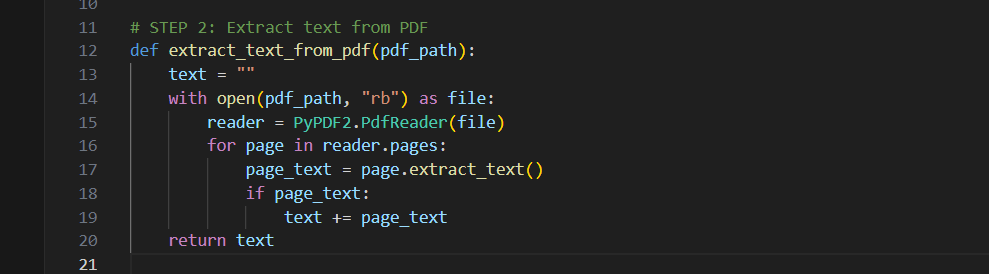
* openai,
* langchain,
* faiss-cpu,
* PyPDF2.

Setup your OpenAI API key to authenticate requests.



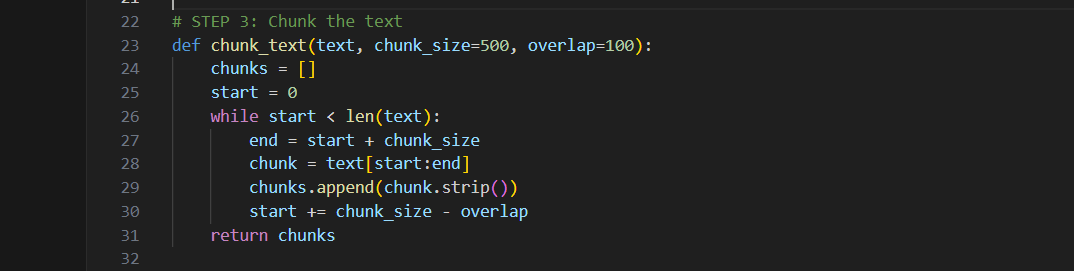
Step 2: Extract Text from PDF

* Use PyPDF2 to open and read the PDF file.
* Extract text page by page and concatenate it into a single string.



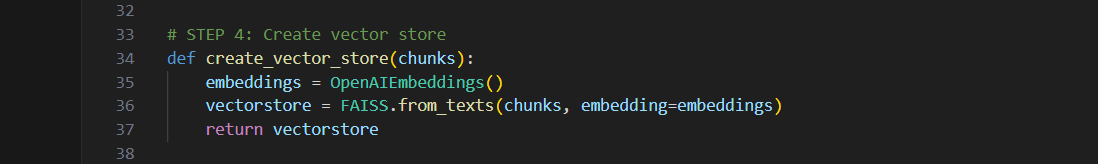
Step 3: Chunk the Text

* Split the extracted text into smaller chunks of about 500 characters each, with an overlap of 100 characters between chunks.
* This improves context retention when generating embeddings.



Step 4: Generate Vector Embeddings

* Use OpenAI's embedding model to convert text chunks into vector representations.
* These vectors numerically capture semantic meaning of the text.



Step 5: Store Embeddings in FAISS Vector Store

* Load the vectors into FAISS, a fast vector similarity search library.
* This allows efficient retrieval of relevant text chunks based on query similarity.

Step 6: Build the Chatbot Query Interface

* Use LangChain’s RetrievalQA chain with OpenAI Chat model.
* The chatbot takes a user question, finds the most relevant text chunks using FAISS, and generates a natural language answer.

Step 7: Run and Interact

* Run the Python script.
* Ask questions in the console about your PDF document content.
* The bot responds with answers based on the embedded PDF text.