PDFs Chatbot using Langchain, GPT 3.5 and Llama 2

This is a Python GUI application that demonstrates how to build a custom PDF chatbot using LangChain and GPT 3.5 / Llama 2.

Important Files and Folders

1. `app/.env`: File with environment variables.
2. `app/requirements.txt`: File with dependencies to be installed for the project.
3. `app/`: Directory containing the source code for the streamlit app.
4. `docs/`: Documentation including meeting minutes and README in PDF format.

How it works (GPT 3.5)

1. The application gui is built using streamlit
2. The application reads text from PDF files, splits it into chunks
3. Uses OpenAI Embedding API to generate embedding vectors used to find the most relevant content to a user's question
4. Build a conversational retrieval chain using Langchain
5. Use OpenAI GPT API to generate respond based on content in PDF

Requirements

1. Install the following Python packages:

```
pip install -r requirements.txt
```

2. Create a `.env` file in the root directory of the project and add the following environment variables:

```
OPENAI_API_KEY= # Your OpenAI API key
```

Note

> Make sure you are in the `/app` directory

Code Structure

The code is structured as follows:

- `app.py`: The main application file that defines the Streamlit gui app and the user interface.
  - get_pdf_text function: reads text from PDF files
  - get_text_chunks function: splits text into chunks
  - get_vectorstore function: creates a FAISS vectorstore from text chunks and their embeddings
  - get_conversation_chain function: creates a retrieval chain from vectorstore

    - handle_userinput function: generates response from OpenAI GPT API
- `htmlTemplates.py`: A module that defines HTML templates for the user interface.

How to run

```
streamlit run app.py
```

> **Note**: Make sure you are in the `/app` directory

Update to use Llama 2 running locally

1. Install Python bindings for llama.cpp library

   ```
   pip install llama-cpp-python
   ```

2. Download the llama 2 7B GGML model from https://huggingface.co/TheBloke/LLaMa-7B-GGML/blob/main/llama-7b.ggmlv3.q4_1.bin and place it in the models folder
3. Switch language model to use Llama 2 loaded by LlamaCpp
4. Switch embedding model to MiniLM-L6-v2 using HuggingFaceEmbeddings