## Brance ML Intern Task

Name:Nithesh R

Linkedin Profile: <https://www.linkedin.com/in/nithesh-r-9166b0215>

Date Challenge Received:29/07/2023 2:00 pm

Date Solution Delivered:31/07/2023



## 1. Problem Statement

What was the task and how you understood it.

The task was to build a RAG(Retrieval Augmented Generation) chatbot. For user question RAG module would retrieve context from knowledge document and generation phase llm would personalize answer using retrieval knowledge. I searched with llms, google, research papers about RAG which helped me to have good understanding of the problem.

2. Approach

Langchain Integration:

Langchain is a decentralized data exchange network that enables secure sharing of data across various organizations and systems. Integrate Langchain into the chatbot's architecture to facilitate secure data exchange between the chatbot, OpenAI, and other relevant parties.

OpenAI Integration:

Integrate OpenAI's RAG model (or any other suitable language model) into the chatbot to enable natural language understanding and generation. OpenAI's API can be used to interact with the RAG model and fetch relevant information based on user queries.

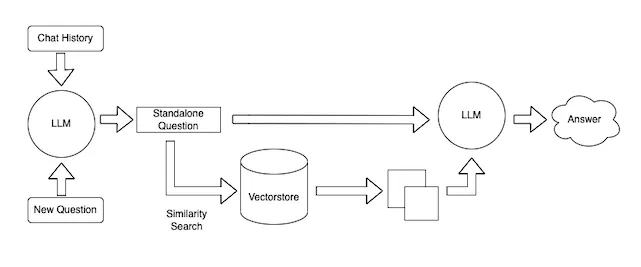
User Authentication and Authorization:

Implement a robust user authentication mechanism to ensure that only authorized users can access the chatbot and their Pan Card details. This may involve using multi-factor authentication (MFA) or other secure authentication methods.

Chatbot User Interface:

Develop a user-friendly interface for the chatbot that allows users to interact naturally and ask questions related to Pan Card details.

## 3. Solution



Data Loading and Splitting: The chatbot loads the uploaded text file and splits it into separate documents for efficient processing. To achive this Langchain was used to load the text document

After Loading the data needs to be splitted into useful chunkz MarkdownHeaderTextSplitter

Is used as the text file consist of

("#", "Header 1"),

    ("##", "Header 2"),

    ("###", "Header 3"),

The total chunkz were 25

Embedding and Database Creation: OpenAI's embedding technique is used to create a vector database from the loaded documents. This database allows for quick and accurate information retrieval.

OpenAIEmbeddings

from langchain.vectorstores import Chroma

These were used to create a vectorDB to store the chunkz

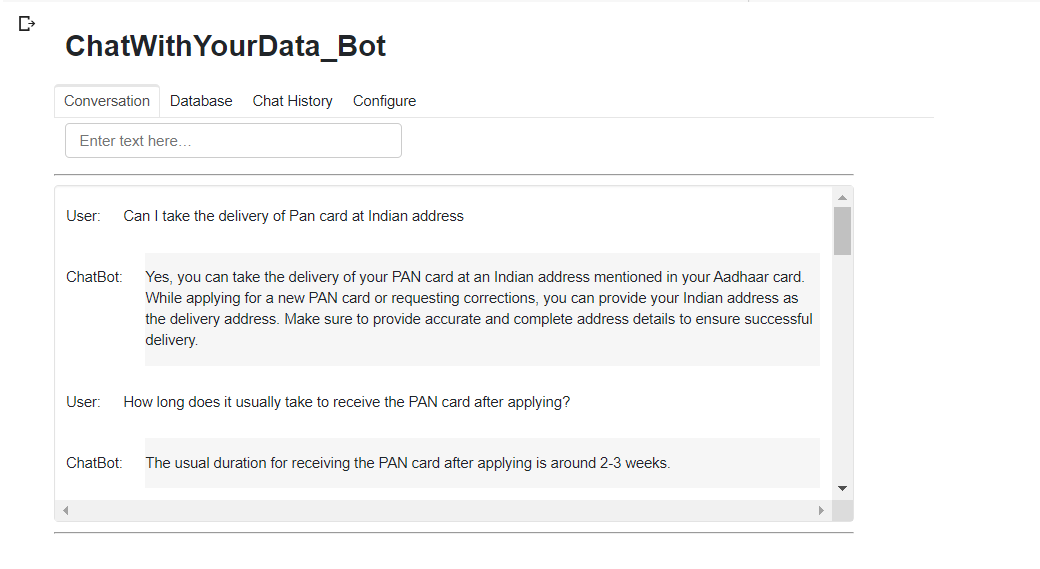
Question Answering: When you enter a question, the chatbot utilizes the Langchain and OpenAI RAG model to understand your query and find the most suitable answer from the database.

To find the chunkz most relevant for the question similarity\_search

Is used where relevant chunkz are retrived for the given question

Conversation Management: The chatbot maintains a chat history to keep track of the ongoing conversation, ensuring context-aware responses.

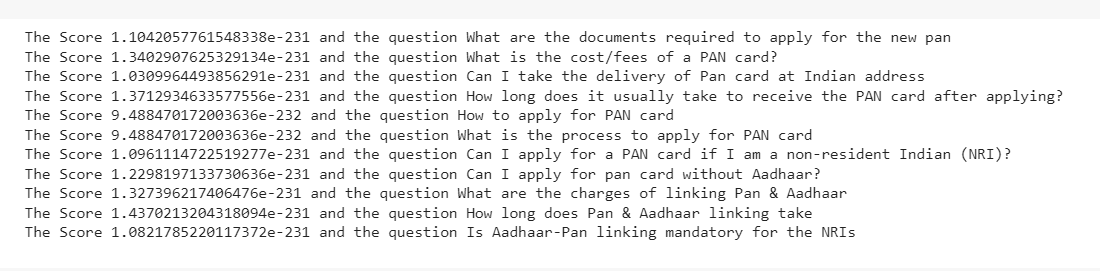
To achieve a interface Panel was used which actually intergrated with langchain so the above steps created as a function and Added to the interface and a chat history is also maintained



BONUS:

To check the Ideal Answer’s similarity with the Answer give by the model BLU score is used

where the Value is between 0 to 1



## 4. Future Scope

1. Instead of using chatgpt api a model can be fine tuned and intergrated with LangChain
2. Instead of Using VectorDB  such as TF-IDF or SVM can be used.