**DOCUMENTATION OF VIBELINKMODEL-EC2-DEPLOY**

**Introduction:**This project showcases the use of Retrieval-Augmented Generation (RAG) to build a sophisticated chatbot. The chatbot leverages OpenAI embeddings and Chroma for document storage and retrieval, providing an enhanced question-answering experience. This project is containerized using Docker Compose for easy deployment.

**Working:**

First we need to build a docker image for this model. We have Dockerfile containing all the required dependencies setting up the python environment.

We up the docker container by using this command:

“**docker-compose up --build”**

**Errors Encounter During the first run:**

But during first run quite a problems occur as listed below:

* The error I countered First when I built Dockerfile is that openapi codec and pypika are not compatible with the backend PES 27.
* So First manually installed them with no PES 27 backend, both installed successfully.
* But again the same error pops up because these dependencies are present in requirements.txt file.
* So the next step is that I manually enter them in Dockerfile and through that they directly install when docker-compose build command runs.

**Output and Results:**

After these when the docker container is up, we have to test send requests to the API to interact with the chatbot.

**Case 1:**

Querying the Chatbot

You can use curl or any API client (like Postman) to send a POST request to the /api/excerpt/transcript endpoint with your prompt.

curl -X POST http://localhost:8000/api/excerpt/transcript -d "user\_file=vocabularies.txt" -d "user\_id=dfgh-gfjfjk-hjdkk-jdjjgg" -d "username=Akin"

Example: Expected Response

The API will respond with a JSON object containing the answer.

{

"result": "The capital of France is Paris."

}

API Endpoints

POST /api/excerpt/transcript: Sends a query to the chatbot and returns the answer.

**Case 2:**

curl -X POST v -d "user\_file=vocabularies.txt" -d "user\_id=dfgh-gfjfjk-hjdkk-jdjjgg" -d "username=Akin"

Response: JSON object containing the result.

{

"username": "Akin",

"file\_id": "fgh-gfjfjk-hjdkk-jdjjgg",

"text":"The capital of France is Paris."

}

POST /api/excerpt/chatdata: Sends a query to the chatbot and returns the answer.

Request Body: JSON object containing the prompt.

curl -X POST http://localhost:8000/api/excerpt/chatdata -d "user\_id=dfgh-gfjfjk-hjdkk-jdjjgg" -d "username=Akin"

Response: JSON object containing the result.

{

"username": "Akin",

"file\_id": "fgh-gfjfjk-hjdkk-jdjjgg"

}

POST /api/excerpt/bot: Sends a query to the chatbot and returns the answer.

Request Body: JSON object containing the prompt.

**Case 3:**

curl -X POST http://localhost:8000/api/excerpt/bot -d "user\_id=dfgh-gfjfjk-hjdkk-jdjjgg" -d "username=Akin" -d "text=how are youy"

Response: JSON object containing the result.

{

"text": "I'm a language model AI, so I don't have feelings, but I'm here and ready to help you with any questions you have. How can I assist you today?",

"file\_id": "fgh-gfjfjk-hjdkk-jdjjgg"

}