# RAG ASSIGNMENT DOCUMENTATION

N26 AI Engineer role

02/11/2025

Gowtham B C
bcgowtham17@gmail.com

# RAG ASSIGNMENT DOCUMENTATION

## Declaration:

Since Gemini key wasn't given, I ran through **free tier quota** for embedding and LLM and couldn't develop or build using cloud. Hence local ollama docker was used to complete the assignment.

**While 90% of the code is same and 100% of concept is same, it is totally free as well.**

I will try to make the entire solution live as well: https://rag.solventumrnd.com/ (Not in scope of assignment)

## Prerequisite

Run all these commands to ensure, Cuda enabled Nvidia graphics card and docker is up and running on hardware. Stable internet and 20GB of hardware memory

1. Nvidia-smi
2. nvcc -V
3. docker
4. open docker desktop application

| |
|---|
| If Nvidia graphics card is not configured, please visit docker compose file and comment the line from 72 to 78 i.e. Deploy section<br>By default, GPU is enabled it since it is too slow on CPU. |

## Steps to run:

Please follow every step carefully:

1. have signed in GitHub account. https://github.com/Gowtham171996/N26-Gowtham-AI-Engineer
   open https://github.com/Gowtham171996/N26-Gowtham-AI-Engineer/archive/refs/heads/master.zip
2. Extract the zipped file
3. Open command prompt in folder path and hit DIR
4. Make sure n26rag-compose.yml file is visible. If not you are in wrong path.

5. Run "docker compose -f n26rag-compose.yml up -d"

(Make sure with Cuda GPU enabled or only CPU) Read prerequisites.

6. Wait for 10mins (pip requirements are very bulk in size, please wait)

7. Open the URLs:
   - http://localhost:8000/ (for main application)
   - http://localhost:6333/dashboard (for qdrant)
   - http://localhost:8000/docs (for docs)

8. The UI will load with filed ingested. if not click "Trigger full ingestion" button

9. Ask a query to the model and observe the links it pulled information from.

10. Want to modify ingestion? In "data" folder add a new file (txt, word doc, pdf with text)

11. Delete the vector collection by clicking on UI

12. Perform step 7 to ingest new file.

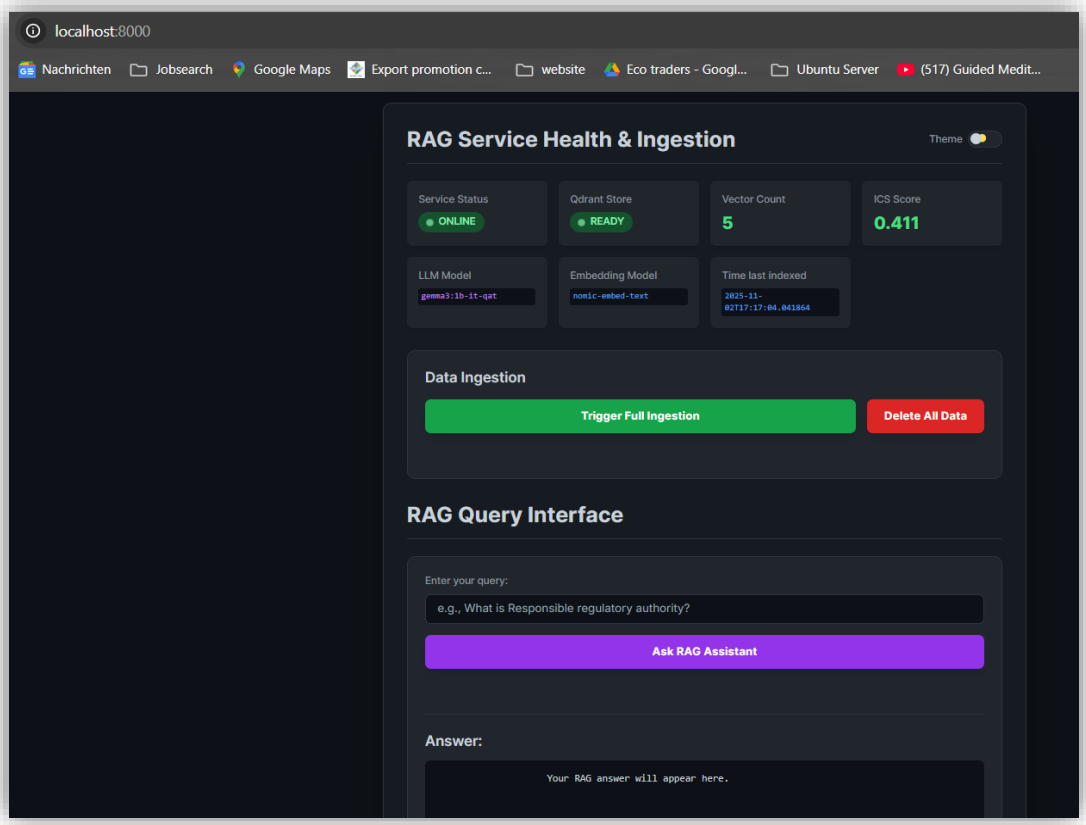
13. Perform step 8 with newly added file.

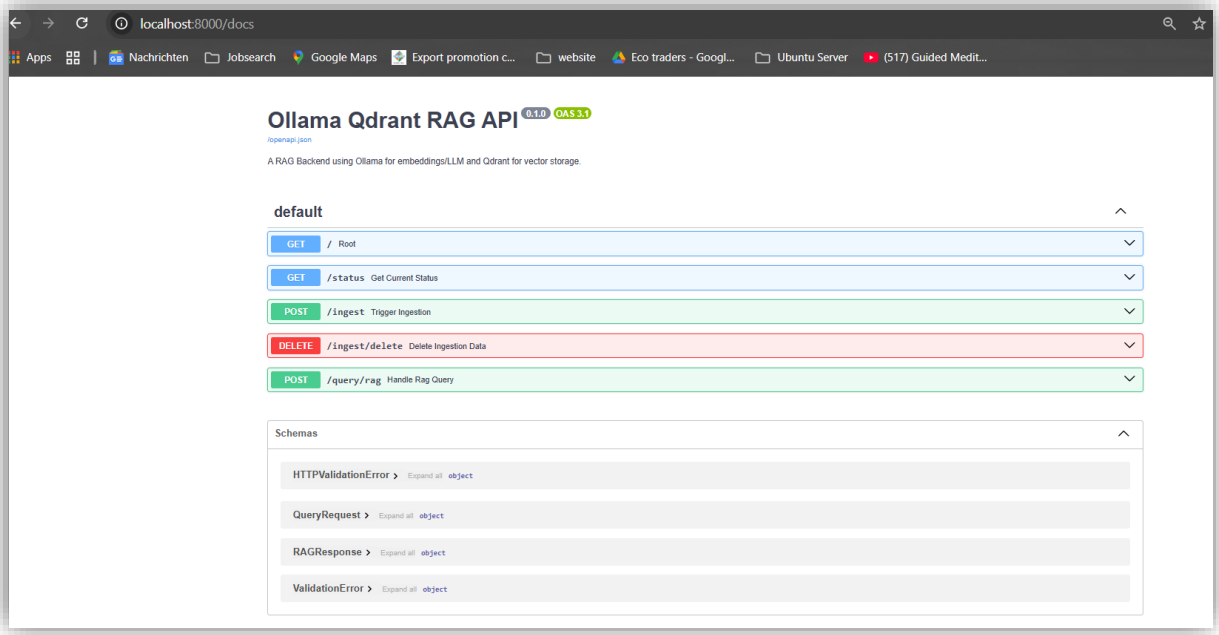
## Known Issues:

1. Twice ingestion is taking place, even though ingestion is persistent.
2. Docker build images are heavy. (For Cuda enabled environment, along with reranker algorithm. PIP requirements are too much.)

# RAG ASSIGNMENT DOCUMENTATION

## Gallery



# RAG ASSIGNMENT DOCUMENTATION