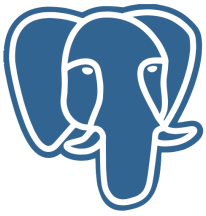
Project Architecture for General Advising Model

Tech Stack:

FastAPI
RESTful API Framework

LangChain
Language Model Orchestration Library



pgvector
Vector Database

PyTest
Testing Framework

OpenAI
LLM and Retiever Model

Llama
LLM and Retriever Model

Installation Guide:

- **New Users:**

Upon successfully adding the project on your local machine, start by creating a `.env`. We want to create an OpenAI API key. BOLT offers 2 options, if you have an ongoing OpenAI subscription or are willing to purchase credits, you are more than welcome to use the OpenAI stream. BOLT also offers a free version, powered by Ollama. You may enter your OpenAI API key like this:

```
OPENAI_API_KEY = "YOUR-API-KEY"
```

For Ollama, head over to Ollama and click the download button. Make sure you have around 20-30GB of storage for a successful installation. Once installed, our next step would be to download the model locally. On the top search bar, search for `llama3.1`. For demo purposes, we would be using the `8b` model with 8.03B parameters. On your command line, run this:

```
ollama run llama3.1
```

Congratulations, you have now successfully installed ollama. Next, we would set up the development system. Run this script to create a python virtual environment, activate it, install dependencies, run the uvicorn server and run backend tests:

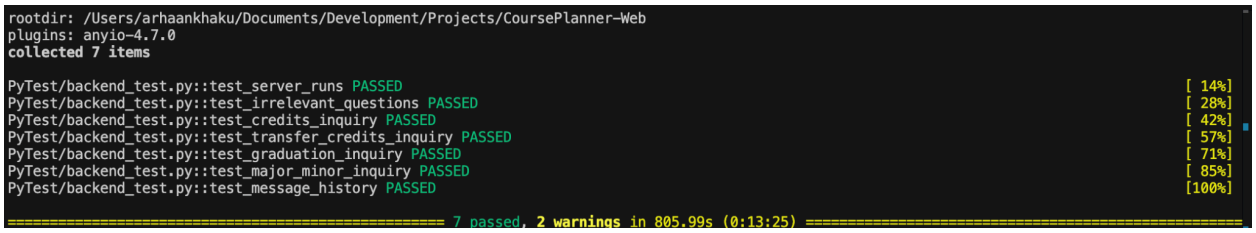
```
python3 src/setup.py
```

- **Existing Users:**

```
For MacOS:

# Create the virtual environment:

python3.12 -m venv venv

# Activate the virtual environment:

source venv/bin/activate

# Install dependencies:

pip install -r requirements.txt

# Start the uvicorn server at localhost:8000

uvicorn generalAdvising.main:app --reload

# Run backend tests:

pytest -v --disable-warnings

For Windows:

# Create the virtual environment:

python3.12 -m venv venv

# Activate the virtual environment:

venv\Scripts\activate

# Install dependencies:

pip install -r requirements.txt

# Start the uvicorn server at localhost:8000

uvicorn generalAdvising.main:app --reload

# Run backend tests:

pytest -v --disable-warnings
```

Important Notes:

1. Backend tests run using Ollama. High wait times for test completion is expected. Below is an image of test coverage done using PyTest as of 23rd December 2024, 2:00am.



The test covers 7 of the main features of general academic advising.

- First test checks if the server is running as expected.
- Second test checks for the model to not answer irrelevant questions asked by the user.
- Third test checks question-answer on general credits inquiry based questions.
- Fourth test checks question-answer on transfer credits based questions.
- Fifth test checks question-answer on graduation requirements based questions.
- Sixth test checks question-answer on major/minor based questions.
- Seventh test checks if message history is working perfectly fine and if the model can remember what you last asked.

2. Whenever you would like to use an OpenAI service, make sure to feed a valid OpenAI API key. You may go to localhost:8000/docs and using the /get_api API, you may feed in your OpenAI API key.

3. Make sure when you create a new virtual environment to check your python version. Many of the packages that have been used for the project work well with python 3.12. You may check python version using this:

```
python —version
```

4. If you receive an error such,

```
ERROR: [Errno 48] Address already in use
```

You should if something is running on port 8000. For the uvicorn server to run successfully, nothing should be running on the port. You may check what is running by using this:

```
lsof -i :8000
```

The following would give you an output of what is running on the port, as well as a PID associated with the task running on that port. To kill the task that is running on the port, use this command:

```
kill -9 <PID>
```

5. Another common error is,

```
httpx.ConnectError: [Errno 61] Connection Refused
```

The error is most likely to do with Ollama. You may want to fix such an error by pulling the latest version of llama3.1 by using this command:

```
ollama pull llama3.1
```

6. We have noticed significant reduces in performance and speed when using Ollama. Highly recommended to use a laptop with more than 8GB RAM.

7. A recommended start upon successful completion of running the server is to check if the uvicorn server is being hit correctly. You should see a message that mentions, `Welcome to Bolt!` After that you may begin by heading towards checking the

```
localhost:8000/docs
```

8. Comments have been added in major chunks of the code to help the feasibility and viability of future developers.

Areas of Future Improvement:

- Making the service student-personalized
- Adding support to more major such as support for Management students
- Improve session history management (maybe using LangGraphs)
- Improved local retriever model with enhanced performance
- File formatting into separate files such as utils.py, routes.py, model.py, main.py and more
- Improve API key encryption
- Hosting on a cloud server
- Add more test cases, testing on various fields
- Add verified answers feature, verified by real academic advisors
- Optimize the model by experimenting on tuning parameters such as k value, chunk size
- Feature for anti-spam messages to bot

Resources:

- https://python.langchain.com/docs/introduction/
- https://research.ibm.com/blog/retrieval-augmented-generation-RAG
- https://blogs.nvidia.com/blog/what-is-retrieval-augmented-generation/
- https://cloud.google.com/use-cases/retrieval-augmented-generation?hl=en
- What is Retrieval-Augmented Generation (RAG)?YouTube · IBM Technology836K+ views · 1 year ago
- How to set up RAG - Retrieval Augmented Generation (demo)www.youtube.com › watch
- What is RAG? (Retrieval Augmented Generation)www.youtube.com › watch
- Retrieval Augmented Generation (RAG) Explained in 8 Minutes!www.youtube.com › watch
- RAG vs. Fine Tuningwww.youtube.com › watch