Application Idea: AI-Powered Search Engine
Name: Anmol Agarwal
Email: aagarwal622@gatech.edu

Reference Application: https://github.com/GerevAI/gerev

Details of Topic: I am working on creating an AI-Powered Search Engine through EvaDB. I will make use of OpenAI models like ChatGPT (for prompt and output generation) and text-embedding-ada-002 (for embedding documents) to create a search functionality. The high level implementation is as follows:

1. Add new documents (pdf, word, text, etc.) and create their embeddings.
2. Store the content of original documents with title and other meta-data in the database.
3. Store the embeddings of original documents in the same database in a separate column.
4. Take the query as input and get its embedding.
5. Use similarity score to get documents with closest embeddings to query and return.

Why is it important?: In various enterprises, the number of documents stored by them is growing rapidly and more importantly, these documents are becoming scattered. Hence there is a pressing need for the search functionality in order to be able to identify those documents based on the query by the user. Moreover, there are currently existing approaches to fetch relevant documents like tf-idf techniques using inverted index, document language models like BM-25, etc. and they are also gradually shifting AI based approaches to get relevant documents. Hence, our solution is a step towards contributing to this trend by creating an AI-Powered Search Engine leveraging ChatGPT and embedding models.

Queries that will be supported:

1. Search: In the command line interface, users can enter the query for getting documents they need and our code in the process those queries in the backend.
2. Search with Min/Max Limit: User can specify how many documents do they need to be outputted.

Anticipated Challenges:

1. Parsing documents: EvaDB supports queries like “LOAD PDF” to load documents into tables. I will be leveraging queries of these types to read different types of documents.
2. User Defined Functions: EvaDB supports various types of functions to different tasks like MnistImageClassifier. So for generating embeddings, I plan to create a new user defined function (if needed) leveraging embedding model. Similarly, the UDF can be created for similarity score.

Development Environment: I can either develop the application on PyCharm IDE or use Jupyter Notebook.

Progress So Far: I have gone through EvaDB github repository, setup evaDB with PostgreSQL on my local machine and ran some sample tutorials on Google Colab. I have also explored Gerev repository to understand how they implement AI-powered search engine.

Testing Details: Once the application is complete, I will give multiple queries and check if the documents outputted the application are relevant or not, and I will fine-tune from there if there are additional opportunities for improving the search function.