Summary Report for the Test Generator Application for CodeMate

By: Tanmay Kumar | 00tanmay@gmail.com | +91 9588131922

Approach Taken:

1. I went through the Google Generative AI documentation that was provided in the assignment. It gave me a good idea of how to generate basic embeddings using the api.
2. I then went through the Chroma docs to understand how to store the embeddings in the db and run queries on it.

   This is how I build the application:

3. I took the input of multiple pdf files. Extracted all the text into a single variable.
4. Created chunks out of the combined text.
5. Created embeddings of these chunks using Google's models/embedding-001
6. Created a vector store and stored it on ChromaDb
7. Took input from the user for the specific topic. Based on the topic, ran a query on chromadb to extract relevant text passages only.
8. Took prompt from the user for the type and no. of questions to generate.
9. Then used Google's LLM model 'gemini-pro' to run the query on the relevant passage.
10. Finally, the QA test can be extracted in HTML or PDF format.
11. I also tested the application with different prompts and different types of pdf and question types thereby improving the code.

Challenges Faced:

1. I had never used Google generative API before. So learning it from scratch from the documentation was challenging.
2. There is no direct listed in the docs for creating and storing embeddings in a vector store. So figuring it out took a lot of attempts.
3. Finding the right prompt and testing it took a good amount of time, and I got an error of API limit reached so I had to create a new API key.

In the end, I solved all these challenges successfully and I totally enjoyed the whole process.

Suggestions:

A few more features such as changing difficulty settings can be added and the prompts can be refined even more. But solving the problem in the minimum time possible is a major criterion so I did my best of what can be done as fast as possible. **I completed the task in a day.**