Documentation Thinking

There are a total of 3 files in the JSON format, consisting of a Quiz Bank, Recent Quiz and Past Quiz.

So, the first step was to look deep into the data and remove the unwanted keys and flatten the hierarchy. This was done in the file Study.py

Thinking - The thought process behind this code is to take raw data from multiple APIs, clean and structure it to make it more useful for analysis or reporting. First, it fetches the data, then it simplifies complex, nested JSON data by flattening it. Next, it removes unnecessary or irrelevant information, keeping only the important data like quiz results and scores. Finally, it saves the cleaned data in formats (JSON and CSV) that are easy to work with, ensuring it’s ready for further analysis or use in reports. The goal is to make the data more accessible and manageable.

Then Data Visualization was completed, which showed the track record of the user to further use it to validate the generated report by LLM. This was done in the file Data\_viz.py

Thinking - The thought process behind the code is to first load and explore the data to understand its structure and key metrics (like quiz titles, final scores, and accuracy). Then, the goal is to visualize the data through different charts, helping to identify patterns, trends, and relationships, such as how quiz scores relate to speed or the distribution of accuracy across students. This analysis provides insights into overall quiz performance.

Then AI\_pattern.py was created to invoke the chatgroq module class from the langchain framework which will use the open source model of llama which will take the cleaned JSON files(1 and 2) as an input and using a customized prompt an output will be given which will provide the recommendations for the user. The clean data decrease the input token significantly, which leads to more optimal usage of the LLM and reduced Hallucinations.

Thinking - The thought process behind this code is to leverage AI to analyze a student's quiz performance and generate actionable insights for improvement. It starts by loading two sets of data: one with quiz questions and options, and another with the student’s responses and performance metrics. The code then processes this information to identify trends, such as correct answers and weak areas where the student struggled. Using this data, it creates a prompt that instructs the AI model (Groq) to analyze the student’s performance and suggest specific steps to improve, like focusing on certain topics or question types. The goal is to provide a comprehensive feedback report that helps the student understand their strengths and weaknesses and guides them on how to improve.

Finally, Persona.py was created, which collected all the data in the files and summarized it for the user, currently the system is based on the accuracy metric and only consists of 2 simple logic of less than and greater than. Some random quotes and motivations were dropped as a cool idea which could provide a refresher to the user. Also, on the accuracy metric goals were also designed.

Thinking - The thought process behind this code is to create a detailed, data-driven student persona based on quiz performance data. The code first loads student performance data from a CSV file, calculates key statistics such as average accuracy and speed, and identifies the student's strengths and weaknesses. Then, it uses this information to generate a unique persona with a biography, goals, motivations, challenges, and even relevant quotes, offering a holistic view of the student’s academic profile. This approach combines data analysis with narrative creation, making the persona both data-informed and relatable.

A notebook with writing on it

Description automatically generated

A notebook with writing on it

Description automatically generated