

E-Exam Generator App (agentic framework)

A. Introduction

Problem – Educators, trainers, and content creators for crafting high-quality questions for assessments, quizzes, and practice tests invest significant time to create assessment papers.

Objective –

- To create an AI solution that will generate multiple-choice questions (MCQs) and short questions through large language models and enable to download the instant solution for assessment purposes.
- Aim to implement this functionality by leveraging multiple Crew AI agents in the backend and stream lit user interface for the front end. The agents are further utilizing the large language models to create the output required.

B. Technical Approach

Methodology – Used agentic framework to assign different tasks to different agents such as mcqs are generated by one agent while short questions are created by another agent.

- User Input: The user uploads a pdf.
- MCQ Generation Agent: Generates multiple-choice questions based on the provided pdf content.
- Short Question Generation Agent: Generates short questions based on the provided pdf content.
- Select question type: User can select the type of questionnaire (mcqs/short question) to generate assessment questions from the provided content.
- Select download file format (pdf/word): - User select the desired file format to download the questions generated by the relevant agents.

Workflow –

- Create the MCQ Generation Agent:
 - a. Use an LLM to generate MCQs from the retrieved content.
 - b. Example prompt: "Generate 5 MCQs based on the retrieved text".
- Develop the Short Question Agent:

- a. Use an LLM to generate short questions.
 - b. Example prompt: " Generate 5 short questions based on the retrieved text ".
- o Integrate Crew AI:
 - a. Define the roles and tasks of each agent.

Tools- PyCharm IDE

Frameworks –

- o Groq – Groq platform and groq API is used to access large language models.
- o Crewai – is used for establishing agentic framework to assign different tasks to multiple agents and a crew is established.

C. Development Process

1. Firstly, I tried to create agentic framework by utilizing crewai and groq API through ngrok tunnel while using google colab, but that doesn't worked well. The reason was the tunnel created by ngrok creating issues from colab cloud.
2. Then, I started implementing the functionality by installing the dependencies on local system and adopted pycharm as my IDE for python.
3. On local setup, I implemented the functionality of uploading pdf and generating questions through large language model without using crewai agents.
4. After then, I tried to create a crew to assign tasks to multiple agents.
5. Also, created a function to customize groq model prompt and assign custom groq_llm to all the agents so that they do not use their default openAI llm. And instead utilize the customized groq_llm function.
6. The agents worked well with the assigned llm parameter to their relevant tasks.
7. User is now able to download the generated content in pdf or word file format.

D. Outcomes and Results

- o The pdf is successfully uploaded to generate mcqs and short questions from provided pdf content.

- User can select the type of questions such as mcqs and short questions to generate from the uploaded pdf content.
- After creating the content, the user can download the generated content file as per the selected file format such as pdf or word file format.
- **Improvement:-** currently, I have tested the functionality with 10 pages pdf only. Because of the Rate Limit of llm used. I would like to improve this.

E. Challenges and Solutions

- While using google colab, I tried to hit groq api using ngrok tunnel and it didn't work well due to the tunnel related issues.
- The second challenge is - Rate limit constraint of almost each freely available language models based on tokens per minute (TPM) as mentioned below: -
-

```
litellm.exceptions.RateLimitError: litellm.RateLimitError: RateLimitError: GroqException - {"error":
{"message": "Rate limit reached for model `llama3-8b-8192` in organization
`org_01jjnsxgzne8vstxya22m0ftf` service tier `on_demand` on tokens per minute (TPM): Limit
6000, Used 3548, Requested 3939. Please try again in 14.863s. Need more tokens? Upgrade to Dev
Tier today at
https://console.groq.com/settings/billing", "type": "tokens", "code": "rate_limit_exceeded"}}
```

F. Future Improvements

After uploading a file, there need to be a selection container or RAG search to extract segment of the file in order to generate questions from the selected topics only.
