**Saad Tariq (AI Intern)**

**Submitted To: Robx.ai**

**Task Report: Using Groq API for Chat Completions with Enhancements**

**Task Objective**

The purpose of this task is to integrate the Groq API into a Python application for generating AI-driven chat completions. The implementation fetches an API key securely from a .env file, constructs a chat completion request, and processes user inputs dynamically to provide responses. Additionally, enhancements were implemented to improve usability, maintainability, and robustness.

**Tools and Technologies Used**

* **Programming Language**: Python
* **Libraries**:
  + python-dotenv: To manage environment variables securely.
  + Groq: For interacting with the LLM API.
  + os: To fetch environment variables.
  + logging: For debugging and tracking usage.

**Step-by-Step Implementation**

1. **Environment Setup**:
   * Installed necessary libraries using pip install python-dotenv and configured the .env file to store the API key securely.
2. **Groq API Integration**:
   * Used the Groq library to interact with the LLM (Llama3-8b-8192 model).
   * Constructed a chat completion request with a system prompt and user query.
3. **Code file are attached within zip file**

**Challenges and Solutions**

* **Challenge**: Handling API errors or missing keys.
  + **Solution**: Implemented robust error handling with descriptive messages.
* **Challenge**: Improving user interaction.
  + **Solution**: Allowed user-driven queries through dynamic input.
* **Challenge**: Ensuring reusability.
  + **Solution**: Encapsulated core logic in a reusable function.

**Conclusion**

This task demonstrates how to integrate the Groq API with Python to generate high-quality chat completions. By incorporating enhancements, the application became more interactive, robust, and user-friendly. These improvements make the code suitable for both standalone use and as a foundation for larger applications.