Assignment: CLI-based Chat Application using Gemini API

Role: SDE Intern

Objective:

Build a Command Line Interface (CLI)-based chat application using Python and Google's Gemini API. The application should simulate a chatbot interaction where a user can send messages and receive intelligent responses from the Gemini language model. (We selected Gemini because you can use the APIs for free)

Key Requirements:

1. **Chat Flow:**
    - The application should allow the user to enter messages in a CLI.
    - Messages should be sent to the Gemini API, and responses should be printed in the CLI.
2. **Exit Detection:**
    - The LLM should detect when the user wants to end or exit the conversation.
    - Example prompts that should trigger this behavior: "bye", "exit", "end chat", "I want to leave".
3. **Review & Rating Collection:**
    - When an exit intent is detected, the LLM should politely ask the user for a short review and a rating (from 1 to 5).
    - The review and rating should be captured in a structured format using **Function Calling** from Gemini API.

- Example:

  ```
  {
      "review": "Really liked the way it answered my questions.",
      "rating": 5
  }
  ```

4. **Saving to File:**
   - Save the extracted review and rating in a text file (e.g., `feedback.txt`) in a human-readable format.

Technologies:

- Language: Python
- API: Google Gemini (Text Generation + Function Calling)
- File I/O for saving review and rating

Evaluation Criteria:

- Code quality and readability
- Proper API integration and error handling
- Logical flow of conversation
- Correct use of Gemini's Function Calling
- Handling of edge cases (e.g., invalid rating, no response)
- Review and rating must be saved accurately on conversation exit

Deliverables:

1. `cli_chat.py` – Main Python script
2. `feedback.txt` – Output file to save review and rating (should be auto-generated)
3. `README.md` – with the following sections:
   - Setup instructions
   - How to run the application
   - Example chat sessions
   - Gemini API setup guide (including link to free API key registration)

Useful Resources:

- Gemini API - Text Generation: https://ai.google.dev/gemini-api/docs/text-generation
- Gemini API - Function Calling: https://ai.google.dev/gemini-api/docs/function-calling?example=meeting

Bonus (Optional):

- Maintain a log of all conversations in a `chat_history.txt`

Submission Deadline:

Please submit your solution as a GitHub repo or a zipped file by 27/04/2025.

We look forward to seeing your creative implementation!