Generative AI Text Completion Project Report

1. **Project Overview**

   This project implemented a text completion application using Cohere’s Chat API in a Jupyter Notebook (text_completion_app.ipynb). The app accepts user prompts, sends them to the AI model, and displays generated responses. It also records the conversation history for evaluation.

2. **Experimentation and Evaluation**
   - Relevance and Coherence
     - The AI generated highly relevant and coherent responses for general prompts.
   - Inaccuracies or Biases
     - No major factual inaccuracies were observed in general knowledge queries.
     - However, there is potential for bias in subjective topics or if prompts include biased phrasing. This was not tested deeply due to time constraints.
   - Impact of Changing Settings
     - The temperature parameter controls creativity vs factuality:
       - Lower temperature (e.g. 0.2) produced direct, factual responses.
       - Higher temperature (e.g. 0.9) produced more creative, elaborate, or poetic outputs.
     - Due to Cohere Chat API limitations in this student free tier, some parameters had minimal observable effect.

3. **Reflection on Limitations**
   - When does the model perform well?
     - General knowledge explanations (e.g. science topics simplified for children)
     - Creative tasks like continuing a story or writing haikus
     - Generating clear summaries of short informational texts
   - When does it struggle?
     - Logical reasoning or multi-step calculations (not tested deeply in this project)
     - Very niche programming or technical questions may result in vague or partially correct answers

    - Length limitations: Sometimes the model cuts off before fully finishing a creative story

4. **Suggested Improvements**
    - Filter outputs for appropriateness or toxicity if deployed publicly
    - Add fact-checking by integrating external APIs or knowledge bases to verify factual responses
    - Allow users to set additional parameters (e.g. max tokens, top_p) for deeper experimentation
    - Handle rate limits and API errors gracefully with retry mechanisms for robustness

5. **Challenges and Errors Encountered**
    - API errors:
        - 403 Forbidden errors when using Hugging Face Inference API due to insufficient permissions on free read tokens
        - StopIteration errors from incompatible InferenceClient usage
        - InvalidRequestError in Cohere when using generate() on command-nightly (Chat API required)
    - Environment variable issues:
        - API keys not loading initially until the terminal or Jupyter Notebook kernel was restarted.
    - Switching providers:
        - Initially attempted OpenAI, but free trial credits expired.
        - Tried Hugging Face Inference API, but free tier limitations blocked model access.
        - Finally used Cohere, which worked after switching to Chat API instead of the outdated generate() function.

6. **Final Reflections**

    This project provided hands-on experience with:

    - Prompt design and experimentation
    - Debugging API integration errors

- Understanding Generative AI capabilities and limitations in real-world applications

I learned the importance of reading updated documentation, handling API security securely (e.g. environment variables), and evaluating AI outputs critically for relevance, coherence, and potential biases.