**Reflection:**

During this assignment, I gained a practical understanding of how conversational LLMs work and how system prompts directly influence their behavior. In Part 1, I built a simple CLI-based chat loop. Although basic, it gave me hands-on experience with sending user inputs to an API, handling errors, and maintaining conversation history. This helped me appreciate the importance of keeping context for natural conversations.

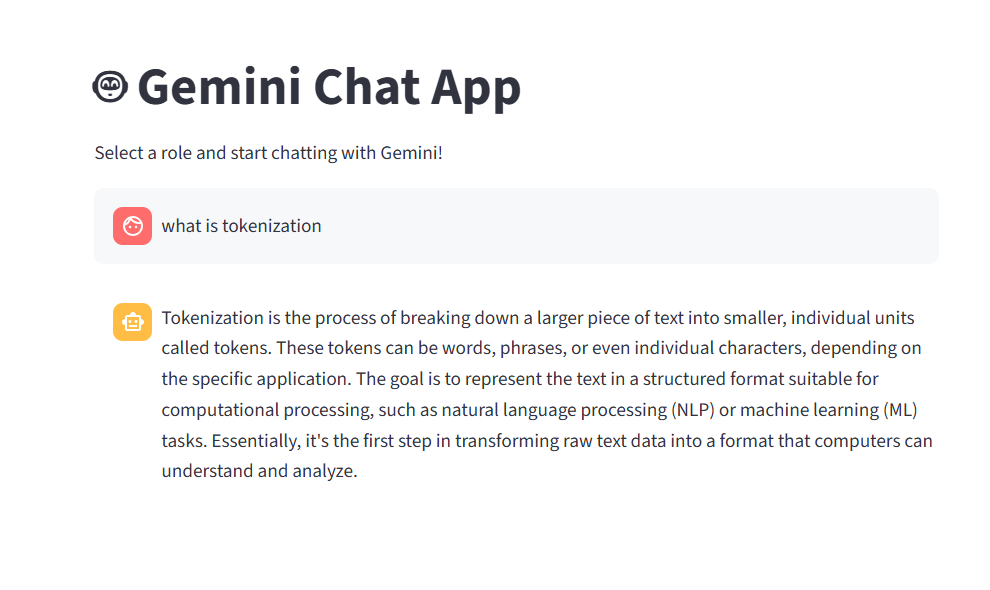
In Part 2, I experimented with three system prompts: a Professional Assistant, a Creative Companion, and a Technical Expert. The responses were noticeably different even when the same user question was asked. This showed me the power of prompt engineering. A slight change in wording or role description can completely transform the AI’s tone, style, and depth of explanation. Evaluating them through a rubric made me realize that clarity and specificity in prompts are essential for consistent and effective results.

Finally, in Part 3, I created a Streamlit interface, which was a big step up from the CLI version. Adding a clean design, role selector, and conversation history display made the experience feel closer to a real chat app. It highlighted how much user interface design contributes to usability.

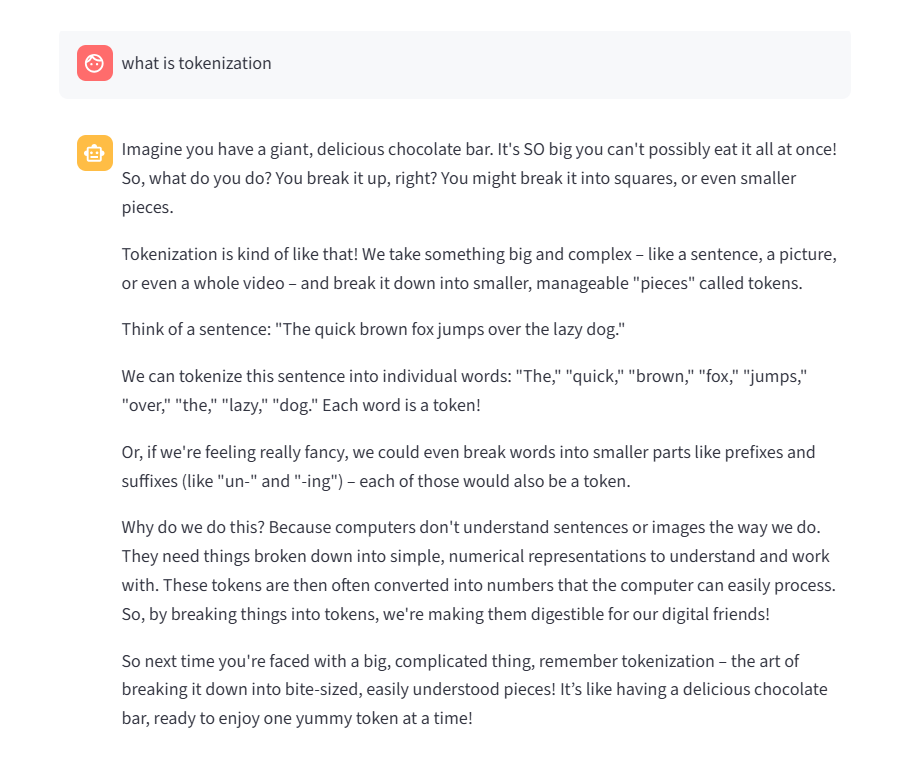
Overall, this assignment helped me bridge theory and practice. I now better understand prompt engineering, conversation design, and how to integrate LLMs into applications in both simple (CLI) and user-friendly (web) ways.

**System Prompt Experimentation:**

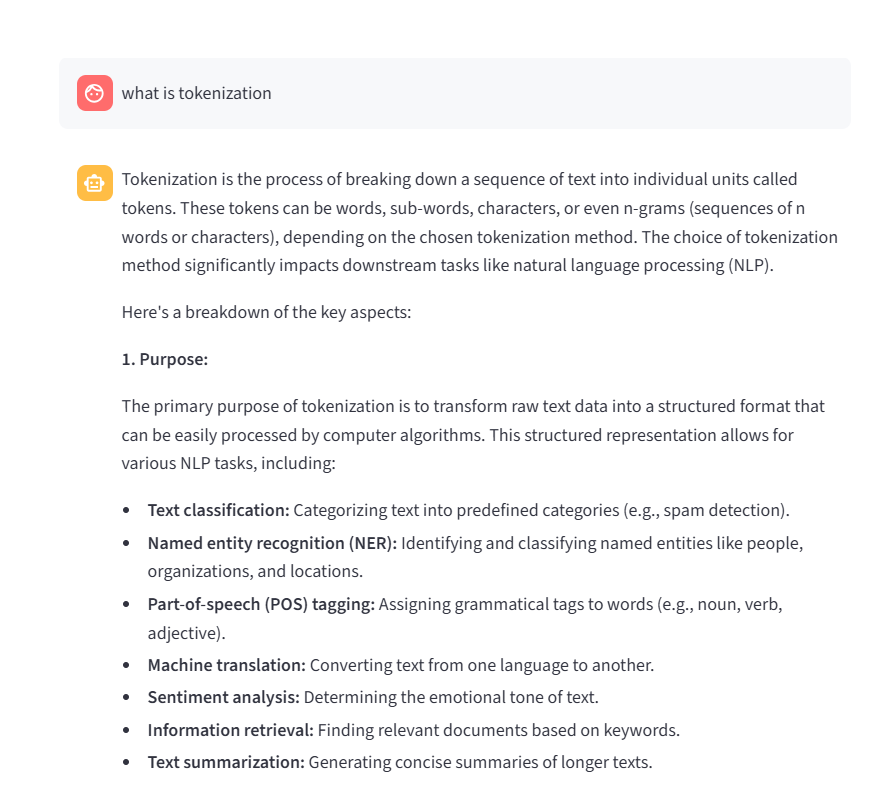
**Professional assistant:**

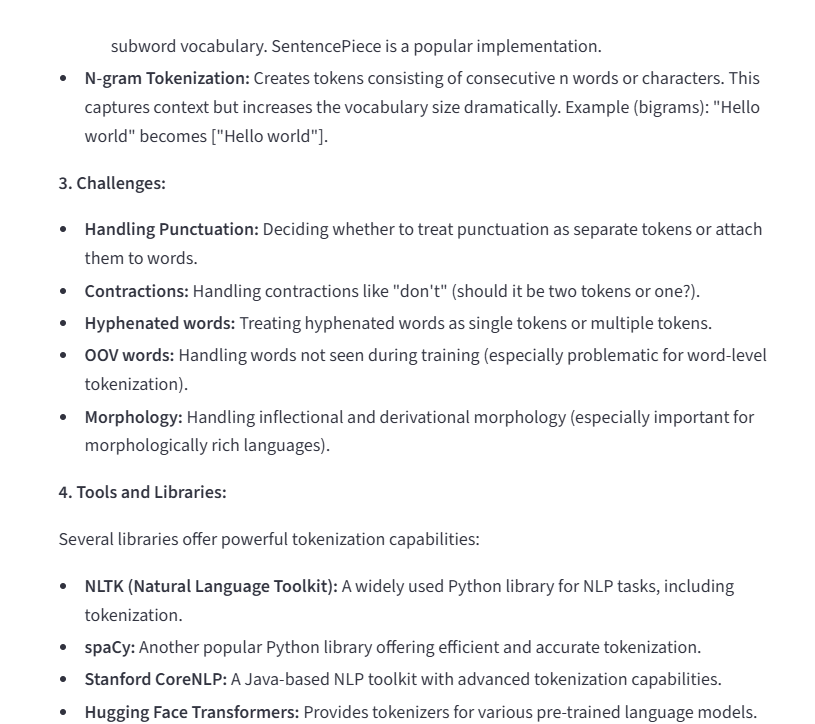
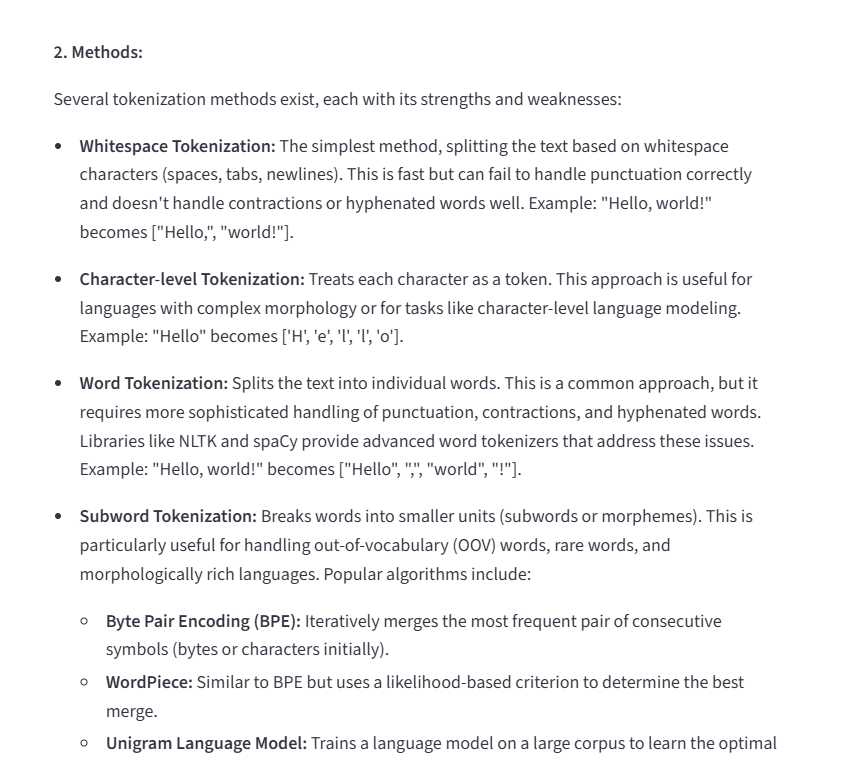


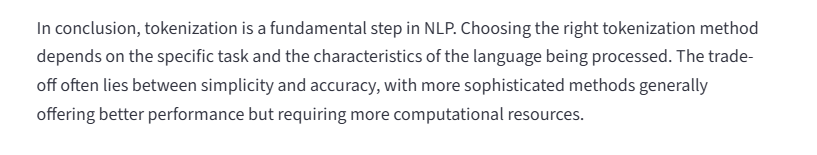
**Creative companion:**



**Technical expert:**







**Streamlit interface:**

