**AI Text Companion App**

**Project Report**

**Troy May**

**June 1, 2025**

***Objective:***

The goal of this project was to gain hands on experience working with generative AI by building a simple text completion app that interacts with a pre-trained llm via API.

***Setup and Tools:***

Library: openai, python-dotenv

Model: gpt-3.5-turbo

API Key Management was securely stored in a .env file

IDE: VS Code

***Application Features:***

Accepts user prompts through the terminal.

Calls the OpenAI API to generate responses based off the specified temperature and token count.

Handles empty or overly long inputs.

Included error handling for invalid API keys, rate limits, and connectivity issues.

***Prompts and Evaluation:***

1. “Hello, how are you?”

***Response****:* Hello! I'm just a computer program, so I don't have feelings, but I'm here and ready to assist you. How can I help you today?

1. “Once upon a time, there was a wizard named gandalf who lived in the U.S. for awhile and...”

***Response****: traveled across the country, using his magical powers to help those in need and spreading wisdom wherever he went. He was known for his long grey beard, tall pointed hat, and his trusty staff. Gandalf was a kind and wise wizard, always ready to lend a helping hand to those who needed it.*

*One day, Gandalf heard of a village that was plagued by a terrible curse. The villagers were living in fear, their crops were failing, and no one knew how to break the curse. Gandalf knew that he had to help, so he set out on a journey to the village.*

*When Gandalf arrived, he immediately sensed the dark magic that was causing the curse. He spoke to the villagers and learned that the curse had*

1. “Explain APIs like I'm a toddler”

***Response****: Okay, imagine you have a toy box full of different toys. Each toy has a special way to play with it, like pushing buttons or pulling strings. APIs are like the instructions that tell you how to play with each toy. They help different programs or apps talk to each other and share information, just like how you share your toys with your friends. So, APIs are like the rules that help things work together and have fun!*

1. “What did the chicken say to the duck?”

***Response****: Quack quack, I mean, cluck cluck!*

1. “In Warhammer 40k, who is Tyberos the Red Wake?”

***Response****: Tyberos the Red Wake, also known as the "Carcharodon Chapter Master," is a fearsome Space Marine who leads the brutal Carcharodon Astra Chapter. He is known for his ruthless tactics and merciless nature in battle, earning him a fearsome reputation among both enemies and allies.*

*Tyberos is a hulking figure, clad in ornate Terminator armor adorned with the teeth of his fallen enemies. He wields a massive power axe known as "The Red Wake," which is said to be imbued with the rage and fury of the Carcharodons themselves.*

*Tyberos earned his title of "The Red Wake" through his brutal and relentless pursuit of his enemies. He is known for leading his Chapter in devastating boarding actions and brutal close-quarters combat, often leaving a trail of carnage in his wake.*

*Tyberos' descent into madness is believed to have begun during the Badab War, a conflict that saw the Carcharodons Astra fight against their fellow Space Marine Chapters. It is said that the horrors of war and the betrayal of former allies drove Tyberos to the brink of insanity, leading him to embrace his savage nature fully.*

*Despite his madness, Tyberos is a formidable leader and warrior, inspiring fear and respect in equal measure. He is a master tactician and strategist, able to outmaneuver and outfight even the most skilled opponents.*

*In battle, Tyberos is a whirlwind of destruction, cutting through enemy ranks with*

**Thoughts:**

The prompting went very well given the requirements. The higher the temperature, the more creative and playful the model got. Lowering the temperature resulted in responses being more concise. The model didn’t seem to struggle much regardless of what was asked of it. There were no crashes or unexpected exceptions during my testing.

***Limitations and Improvements:***

Factual verification is not built in. The app could benefit from an optional fact-checking option. A big upgrade would be to include some sort of GUI or web interface with a pleasing UI.

***Conclusion:***

This project helped me understand the fundamentals of generative AI and prompt engineering. Through the API and application building process, I learned how to create a functional model through the terminal that could help any user with any topic. This also helped me learn a fundamental practice of hiding API keys. While researching APIs and implementing the OpenAI key into my VS Code, I learned the proper key hiding technique to ensure security for myself and others.