Hands-On Exploration of Generative AI with OpenAI's GPT-3.5

Objective

The aim of this project is to gain practical experience with Generative AI by building a simple Python-based text completion application that interacts with a pre-trained language model via API.

Tools and Technologies

- **Model:** OpenAI GPT-3.5 Turbo
- **Environment:** Jupyter Notebook
- **Languages:** Python
- **Libraries:** `openai`, `dotenv`
- **Platform:** OpenAI API

Implementation

The application accepts user prompts, sends them to the GPT-3.5 model, and displays the generated response. The system includes:

- Secure API key handling
- Adjustable response parameters (`temperature`, `max_tokens`)
- Multiple test prompts across different genres
- Error handling for empty input or API issues

Experimentation

Five prompts were tested using varying temperatures:

Prompt	Temperature	Max Tokens	Notes
Once upon a time…	0.8	100	Creative and fluid story

Explain photosynthesis…	0.5	100	Clear and age-appropriate
Haiku about the ocean	0.7	50	Correct structure, evocative
Explain recursion…	0.3	60	Concise, but could use metaphors
Summarize Romeo and Juliet	0.6	150	Accurate summary, minor poetic liberties

Observations

- **Strengths:** High fluency, good general knowledge, creative outputs
- **Weaknesses:** Occasional factual inaccuracies, difficulty with deep reasoning
- **Temperature Effects:** Lower values gave more factual, controlled responses; higher values were more creative but less predictable

Reflection

This project revealed the impressive language abilities of AI, especially for storytelling, summarizing, and rephrasing. However, limitations emerged in factual reliability and logical consistency. Future improvements could include:

- Output verification with retrieval models
- Advanced input validation
- UI-based frontend for broader accessibility