**PROJECT REPORT**

**Title: AI-Based Horror Story Generator Using OpenAI GPT API**

**Introduction**

Storytelling is an essential element of human expression. With the evolution of AI, machines can now assist or even create stories autonomously. This project focuses on generating **horror-themed short stories** using the **OpenAI GPT (ChatGPT)** language model via API integration in Python.

**Objective**

To build a Python-based application that:

Accepts user-defined inputs like **setting**, **protagonist**, and **fear type**.

Sends these inputs to the **OpenAI GPT model**.

Returns a complete horror story with optional **twist endings**.

Demonstrates the use of **generative AI in creative writing**.

**Tools & Technologies Used**

|  |  |
| --- | --- |
| **Tool/Tech** | **Description** |
| Python 3.x | Programming Language |
| OpenAI GPT-4 API | Core language model for story generation |
| openai Python SDK | Library to interact with the OpenAI API |
| pip | Package manager to install dependencies |
| Terminal / IDE | To run the Python script |

**System Workflow**

**Step 1**: User inputs horror story elements:

Setting (e.g., haunted house, asylum)

Protagonist (e.g., Clara the ghost hunter)

Fear Type (e.g., psychological, supernatural)

Optional: Twist Ending

**Step 2**: These inputs are added to a crafted **prompt**.

**Step 3**: Prompt is passed to the **OpenAI ChatCompletion API** using GPT-4.

**Step 4**: GPT-4 generates the horror story and sends it back.

**Step 5**: The story is displayed in the terminal or printed in output.

**Source Code**

import openai  
  
# Replace with your OpenAI API key  
openai.api\_key = "YOUR\_API\_KEY"  
  
def generate\_horror\_story(setting, protagonist, fear\_type, twist=True):  
    prompt = f"""  
    Write a horror story based on the following inputs:  
    - Setting: {setting}  
    - Protagonist: {protagonist}  
    - Type of fear: {fear\_type}  
    - Include a twist ending: {"Yes" if twist else "No"}  
  
    The story should be eerie, suspenseful, and highly atmospheric. Keep the reader on edge.  
    Begin now:  
    """  
  
    response = openai.ChatCompletion.create(  
        model="gpt-4",  
        messages=[  
            {"role": "system", "content": "You are a horror story writer."},  
            {"role": "user", "content": prompt}  
        ],  
        temperature=0.9,  
        max\_tokens=800  
    )  
  
    story = response['choices'][0]['message']['content']  
    return story  
  
# Example usage  
if \_\_name\_\_ == "\_\_main\_\_":  
    setting = "abandoned mental asylum during a thunderstorm"  
    protagonist = "a lone paranormal investigator named Clara"  
    fear\_type = "psychological and supernatural"  
  
    story = generate\_horror\_story(setting, protagonist, fear\_type)  
    print("\n Generated Horror Story:\n")  
    print(story)

**Output Sample**

Generated Horror Story:  
  
Clara stepped into the asylum as lightning forked across the sky. Each creak of the rotting floor echoed like whispers from the past. But what she didn't know was that the mirror in the hallway didn’t reflect the present — it showed the exact moment she'd die…

**Key Features**

Interactive input for dynamic story elements

Supports twist endings and various fear genres

Uses GPT-4 for realistic and eerie storytelling

Can be adapted for GUI or web versions

**Future Enhancements**

Build a **Tkinter GUI** interface

Add **voice narration** using text-to-speech (TTS)

Export stories to PDF or share on social media

Support multiple horror subgenres like Gothic, Paranormal, Slasher, etc.

**Applications**

Content creation for games, books, or films

Creative writing assistance for authors

Entertainment or storytelling bots

AI in education for narrative learning

**Conclusion**

This project demonstrates the integration of powerful generative AI (GPT-4) into storytelling applications. It successfully automates the generation of horror stories based on minimal user input and showcases the future potential of AI in the creative industry.