

AI-Powered Storyteller

AI-Powered Storyteller - Detailed Explanation

The **AI-Powered Storyteller** is a creative writing tool that generates **story continuations** based on user input and selected **genres** (Fantasy, Sci-Fi, Horror, Romance, Mystery, or General). Users can:

1. **Generate multiple story continuations** based on their input.
2. **Refine a generated story** to enhance or modify the content.
3. **Engage in a collaborative storytelling chat** where the AI interacts dynamically with user input.

This tool leverages **DeepSeek AI via Ollama** to provide **engaging and genre-specific story development**.

File Structure & Detailed Breakdown

This project consists of three key Python files:

app.py - The User Interface (Gradio)

This file is responsible for **setting up the user interface** using **Gradio**, allowing users to:

- **Generate a story continuation** from a given prompt.
- **Choose a genre** for AI-generated content.
- **Refine existing story content**.
- **Engage in a collaborative storytelling chat**.

Key Functionalities:

- `story_generator(user_story, genre)` → Generates **three different story continuations**.
- `refine_story(existing_story, genre)` → Enhances and **rewrites a story**.
- `chat_mode(user_input, history)` → Simulates a **real-time storytelling experience**.

How It Works:

- Users **input a story prompt** and **select a genre**.
- The system generates **three different continuations** using `generate_story()`.

- Users can **refine a story** or **collaborate interactively** with AI.

◆ Example UI Code:

```
generate_btn.click(  
    story_generator,  
    inputs=[user_input, genre],  
    outputs=[output_1, output_2, output_3]  
)
```

2 `ollama_interface.py` - AI Story Generation

This file is responsible for **communicating with Ollama's DeepSeek AI model** to generate creative stories.

📌 Key Functionalities:

- `generate_story(prompt, genre, max_tokens)` → Sends a **formatted prompt** to DeepSeek and **returns a generated story**.

🔍 How It Works:

1. **Formats a prompt** based on **user input and genre**.
2. **Sends the request** to DeepSeek AI using Ollama.
3. **Retrieves the AI-generated story** and returns it.

◆ Example AI Interaction Code:

```
def generate_story(prompt, genre="General", max_tokens=200):  
    full_prompt = f"Write a creative continuation in {genre} style:\n{prompt}  
\n\nResponse:"  
  
    response = ollama.chat(model="deepseek-r1:1.5b", messages=[{"role":  
"user", "content": full_prompt}])  
  
    return response['message']['content'] if 'message' in response else "Err  
or generating story."
```

3 `templates.py` - Genre-Based Story Prompts

This file contains **predefined prompts** for different genres, ensuring that AI-generated content **aligns with user expectations**.

Key Functionalities:

- `GENRE_TEMPLATES` → Stores story settings for **Fantasy, Sci-Fi, Horror, Romance, Mystery, and General**.
- `get_genre_prompt(user_story, genre)` → Combines **user input** with the **genre-specific setting**.

How It Works:

- If the user selects "**Fantasy**", the AI receives this instruction:

"A mystical land filled with dragons and magic. Write a continuation where..."

- If the user selects "**Horror**", the AI is guided with:

"A haunted house with dark secrets. Continue the eerie tale..."

◆ Example Genre Prompt Code:

```
GENRE_TEMPLATES = {
    "Fantasy": "A mystical land filled with dragons and magic. Write a continuation where...",
    "Sci-Fi": "A futuristic world ruled by AI overlords. Continue the story where...",
    "Horror": "A haunted house with dark secrets. Continue the eerie tale..."
}

def get_genre_prompt(user_story, genre):
    return f"{GENRE_TEMPLATES.get(genre, 'General')} {user_story}"
```

How Everything Works Together

- 1 User enters a story idea & selects a genre in the UI (`app.py`).
- 2 The system formats a genre-based prompt (`templates.py`).

- 3 AI generates multiple story continuations (`ollama_interface.py`).
 - 4 Users can refine their story or interactively collaborate with AI.
-

How to Run the Project

Run the application:

```
python app.py
```

Enter a story prompt, choose a genre, and generate stories!
