**AI Story Generator**

**1. Project Overview**

The AI Story Generator is a web application designed to generate creative stories based on user prompts. The application uses OpenAI's GPT-3.5 API to generate stories, providing a simple and intuitive interface for users to interact with.

Note: Due to an insufficient API quota, this demonstration uses mock data to simulate the story generation process.

**2. Installation and Setup**

2.1 Prerequisites

* Python 3.7 or higher
* Git (optional, for version control)
* OpenAI API Key (if real-time generation is required)

2.2 Installation Steps

1. Clone the repository:

git clone https://github.com/your-username/ai-story-generator.git

cd ai-story-generator

1. Create a virtual environment:

python -m venv venv

1. Activate the virtual environment:

venv\Scripts\activate

1. Install dependencies:

pip install -r requirements.txt

1. Set up the OpenAI API key:
   * Replace the placeholder in app.py with your actual OpenAI API key:

openai.api\_key = "your\_openai\_api\_key\_here"

1. Run the Flask application:

python app.py

1. Access the application:
   * Open a browser and navigate to <http://127.0.0.1:5000>.

**3. Usage Guide**

3.1 User Interface

* Prompt Field: The user enters a story prompt (e.g., "Once upon a time, in a magical forest").
* Generate Story: Clicking the "Generate Story" button processes the prompt and displays the generated story below the input field.

3.2 Example Prompts and Outputs

* Prompt: "Once upon a time, in a magical forest"
  + Generated Story: "A brave adventurer named Leo entered the enchanted woods, where mystical creatures roamed and hidden treasures awaited."
* Prompt: "A knight slays a dragon"
  + Generated Story: "Sir Galvin stood tall, sword raised high, as the fearsome dragon circled above. With one swift strike, he defeated the beast and saved the kingdom."
* Prompt: "A journey through space"
  + Generated Story: "Captain Zara and her crew embarked on a daring voyage to the farthest reaches of the galaxy, encountering alien civilizations and discovering new worlds."

**4. API Integration Details**

4.1 Overview

The application integrates with OpenAI's GPT-3.5 API to generate creative stories based on prompts.

4.2 API Endpoint

* Endpoint: /generate\_story
* Method: POST
* Request Body:

{

"prompt": "Your story prompt here"

}

* Response:

{

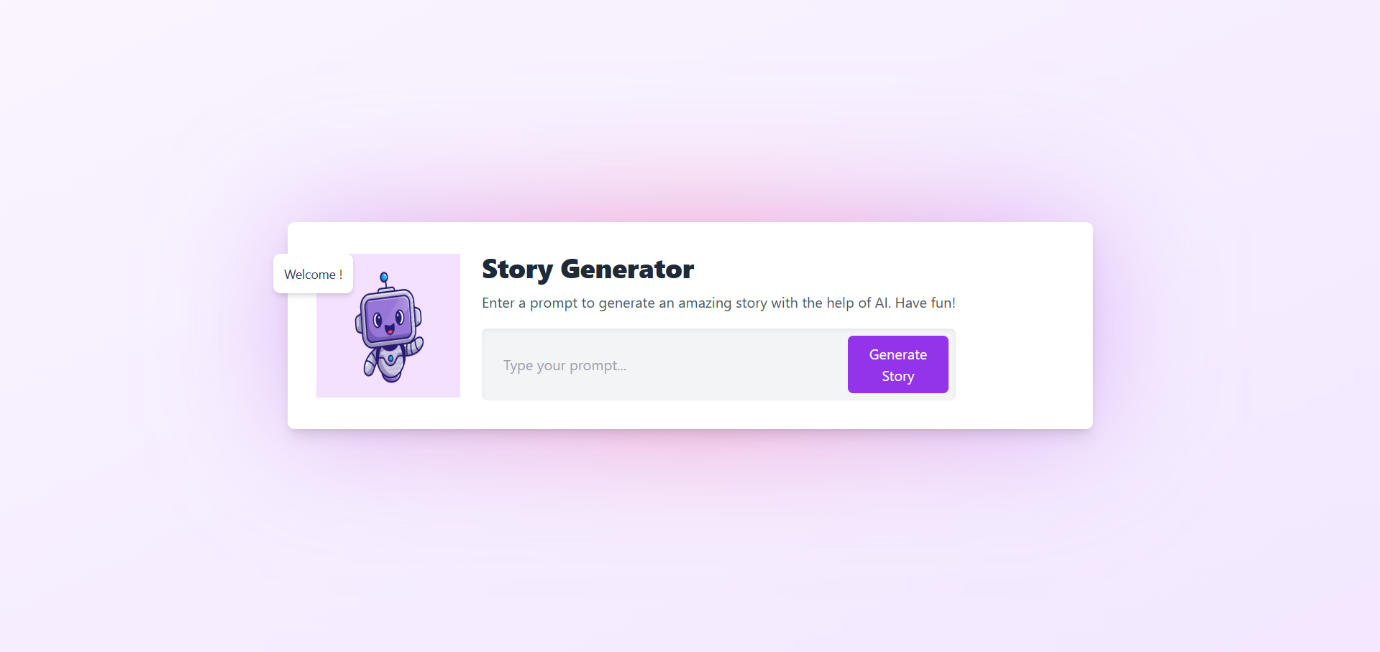
"story": "Generated story based on the prompt"

}

4.3 Mock Data for Quota Issue

Due to an insufficient API quota, mock data is used for demonstration purposes. Prompts are matched to pre-defined stories.

**5. Screenshots**



**6. Key Learnings and Challenges**

6.1 Key Learnings

* Integration of OpenAI API with Flask.
* Implementation of Tailwind CSS for frontend styling.
* Handling API errors and providing fallback options.

6.2 Challenges

* API quota limitations required switching to mock data for the demo.
* Debugging the interaction between the frontend and backend.

**7. Future Work**

* Resolve API quota issues for live story generation.
* Deploy the application online for wider accessibility.
* Add features like customizing story length, genre, and tone.

**8. Conclusion**

The AI Story Generator leverages GPT-3.5 to simplify creative writing. The app demonstrates the potential of AI in generating unique content based on user inputs, even with the current constraints.