

# Story Generation using Natural Language Processing

## An user-Friendly Application of Story Generation

N. Tejitha Sai Sridevi<sup>1</sup>

Department of Computer Science and Engineering (CSE)  
Sri Vasavi Engineering College, Tadepalligudem,  
Andhra Pradesh, India

M V V Gopala KrishnaMurthy<sup>2</sup> (Assistant Professor)

Department of Computer Science and Engineering (CSE)  
Sri Vasavi Engineering College, Tadepalligudem,  
Andhra Pradesh, India

**Abstract:-** This project centers on an innovative approach to real-time story generation, harnessing the capabilities of Flask, GPT-2 (Generative Pre-trained Transformer 2), and web technologies. Its primary objective is to create an intuitive platform where users can input prompts, triggering GPT-2 to generate unique, contextually relevant stories instantly. The seamless integration of Flask serves as the project's backbone, facilitating interactions between the user interface and the GPT-2 models. The core objective lies in demonstrating the fusion of AI-driven content creation and web development. The user-friendly interface, developed using HTML, CSS, and JavaScript, empowers users to input prompts and receive generated stories dynamically. By showcasing the potential of AI in creative content generation, this project illustrates the seamless integration of Flask with GPT-2, enabling real-time story creation for a diverse range of prompts. The ultimate goal is to offer an interactive and engaging storytelling experience that highlights the synergy between AI capabilities and user interaction within a web-based framework.

### I. INTRODUCTION

This task leaves on an imaginative excursion, meeting the domains of man-made reasoning and web improvement to make a state of the art stage for continuous story age. At its center, this try use the force of Flask, GPT-2 (Generative Pre-prepared Transformer 2), and different web innovations to lay out a natural space where clients can easily include prompts, setting off GPT-2 to in a flash create logically significant stories. The integration not only emphasizes the symbiotic relationship between artificial intelligence and web development, but it also demonstrates the technical prowess of AI-driven content creation.

The primary objective of this project is, in essence, to offer users an immersive and captivating storytelling experience that defies conventional boundaries. By wedding the abilities of Transformers, GPT-2, and web advances, this try looks to reclassify the scene of continuous story age, representing the agreeable mix of computerized reasoning and client driven intuitiveness.

### II. LITERATURE SURVEY

#### ➤ *Writing Support Tools :*

Many instruments support various parts of composing. These reach from spelling and sentence structure remedy, to thesauruses, to swarm fueled editors. As composing undertakings and styles are many times space explicit, apparatuses can be correspondingly particular: email message stating, help demands in proficient settings, mental help, affectionate messaging, schooling, and news coverage. Our objective is to use these language models however give an elective control technique utilizing both message and unique visual portrayals and collaborations.

#### ➤ *Machine Story Generation :*

Story age is Generation of the fantastic difficulties in man-made reasoning (man-made intelligence) with numerous useful applications. These range from entertainment and education to our goal, writing support. We focus on visual encodings that can both express high-level changes in the story's progression and be manipulable.

#### ➤ *Visualizing and Visually Expressing Stories :*

To permit clients to determine successive properties of the story and catch how texts are created with controls. Visual procedures have been utilized to show undeniable level credits of reports and stories. Our objective isn't just to address the story through the perception yet to take into account control through that portrayal.

#### ➤ *Sketching Interactions:*

To outwardly communicate consecutive traits of stories, utilizes outlining cooperation. With its roughness, uncertainty, and ambiguity, sketching is a simple, flexible way to convey the user's high-level intentions. In view of these highlights, different outlining devices have been worked to typify these attributes or go about as components to change harsh thoughts into exact portrayals. We influence late work on intuitive perception instruments for information age. However, these primarily focus on numerical methods rather than text (e.g., matching a sketched histogram with a dataset).

➤ *Problem Statement :*

Traditional methods for story generation frequently miss the mark on dynamism and intuitiveness wanted by users. Utilizing Natural Language Processing (NLP) strategies and Flask for ongoing story age presents its own arrangement of difficulties. The current arrangements frequently struggle with client commitment, setting significance, and a consistent reconciliation of NLP capacities inside web applications.

➤ *Disadvantages in Existing System :*

• *Limited User Interaction:*

Numerous story generation devices miss the mark on easy to use interface, ruining clients from effectively taking part in the inventive strategy. The test is to make a natural stage that supports client info and collaboration.

• *No Scalability:*

As the client base develops, versatility turns into a worry. It's hard to build a system that can handle more users and different computational loads without sacrificing performance.

• *Fixed Narrative Structures:*

It may be difficult to generate various story formats or adapt to various storytelling styles because many current systems may be restricted to fixed narrative structures.

• *Difficulty in Generating Realistic Dialogues:*

Story generation frameworks might battle to make practical and drawing in exchanges, restricting the general trustworthiness and submersion of the created stories.

➤ *Proposed System :*

Utilizing Flask, a lightweight web framework, and NLP (natural language processing) methods, the proposed system aims to create an innovative platform for story generation. Through a web interface that is easy to use, users of this system will be able to easily create stories that are engaging and coherent. Flask will be utilized to construct the web application, giving a responsive and versatile stage for story creation. The incorporation of NLP calculations will upgrade the framework's capacity to break down and comprehend regular language inputs, empowering it to produce logically significant and very much organized accounts.

➤ *Advantages in Existing System:*

• *Web Combination with Flask:*

Flask is a lightweight and adaptable web system in Python. Coordinating story age with Flagon permits you to make web applications where clients can connect with and produce stories through an easy to understand interface.

• *Personalization:*

You can tailor the generated stories to each user's interests by analyzing user preferences and behaviors with NLP. This personalization can altogether further develop the client experience.

• *Scalability:*

Flask is intended to be adaptable. You can convey your Flask application on different facilitating stages, making it open to countless clients. This versatility is pivotal, particularly in the event that your story age application acquires prevalence.

• *Adaptability and Customization:*

Flask gives adaptability in planning the UI and considers customization in view of your particular necessities. When users interact with the story generation system, you can create a one-of-a-kind and customized experience for them.

### III. METHODOLOGY

➤ *Set up your Environment:*

• *Introduce Flask:*

Use 'pip install flask' to introduce Flask, a lightweight web structure for Python.

• *Introduce Fundamental NLP Libraries:*

Introduce libraries like NLTK or SpaCy for normal language handling. You could likewise require explicit language models for cutting edge NLP undertakings.

➤ *Design the Web Interface:*

• *Make HTML Layouts:*

Foster HTML layouts for the UI. Use structures to take input from clients (e.g., prompts for story age).

• *Create Flask Routes:*

Characterize courses in Flask to deal with various URL ways and render the proper HTML formats.

➤ *NLP Model Reconciliation:*

• *Pick a NLP Model:*

Select a pre-prepared language model or train your own contingent upon the intricacy of story generation. Famous models incorporate GPT (Generative Pre-prepared Transformer) models.

• *Make use of the NLP Model:*

Utilize the fitting library (e.g., Embracing Face Transformers for GPT models) to stack the pre-prepared model into your Flagon application.

➤ *Handle user Input:*

• *Catch client input:* Extricate client input from the submitted structures in your Jar courses.

• *Preprocess the info:* Clean and preprocess the client contribution to make it appropriate for taking care of into the NLP model.

➤ *Story Generation:*

- Feed contribution to the NLP model: Pass the preprocessed contribution to the stacked NLP model to produce story content.
- Post-process the produced story: Tidy up the result, design it fittingly, and guarantee it seems OK linguistically.

➤ *Show the Outcome:*

- Return the produced story: Render another HTML layout or update the current one to show the created story.
- Handle mistakes: Manage cases in which the NLP model fails to generate a story or encounters issues by implementing error handling.

➤ *NLP Components:*

- ✓ NLP Tasks: Specific NLP functionalities integrated into the backend, such as:
- ✓ Text preprocessing: Cleaning, normalizing, and tokenizing user input.
- ✓ Named entity recognition (NER): Identifying and extracting entities (people, places, organizations) from text.
- ✓ Text classification: Categorizing stories based on genre, theme, or other relevant criteria.
- ✓ Sentiment analysis: Understanding the emotional tone of user input or generated text.
- ✓ NLP Tools/Models: Libraries or pre-trained models used for NLP tasks, such as: NLTK, spaCy, Hugging Face Transformers: Popular Python libraries for NLP.
- ✓ Backend: The NLP tasks are implemented as Python functions or classes within the Flask application.
- ✓ GPT-2 Model: NLP techniques can be used to preprocess user input before feeding it to GPT-2 or to refine generated text.
- ✓ Database (optional): NLP results can be stored in the database for further analysis or personalization.
- ✓ Key Considerations:
- ✓ NLP expertise: Building and integrating NLP components requires knowledge of NLP concepts and tools.
- ✓ Computational resources: NLP tasks can be computationally intensive, especially for large models. Ensure adequate resources for processing and model storage.
- ✓ Data quality: The performance of NLP tasks depends on the quality of training data. Use high-quality, relevant datasets.
- ✓ Ethical implications: Be mindful of potential biases in NLP models and the impact on story generation.

**IV. SYSTEM ARCHITECTURE**

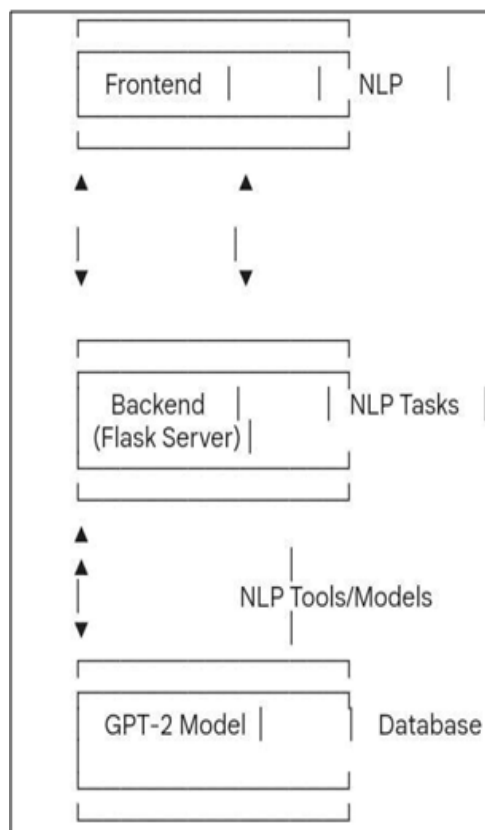


Fig 1 System Architecture

➤ *Additional Recommendations:*

- Continuous Evaluation: Regularly evaluate the performance of NLP components and make adjustments as needed.
- Experimentation: Explore different NLP techniques and models to find the best fit for your application's specific needs.
- Stay Updated: Keep up with advancements in NLP research and tools to improve your application's capabilities.

### V. EXPERIMENTAL RESULTS

In the wake of exploring to the web interface , client or user needs to choose say a story and afterward client will be approached to choose a genre that has romantic ,funny and scary genres. In the wake of choosing the genre ,story length should be determined and the regardedstory will be produced.

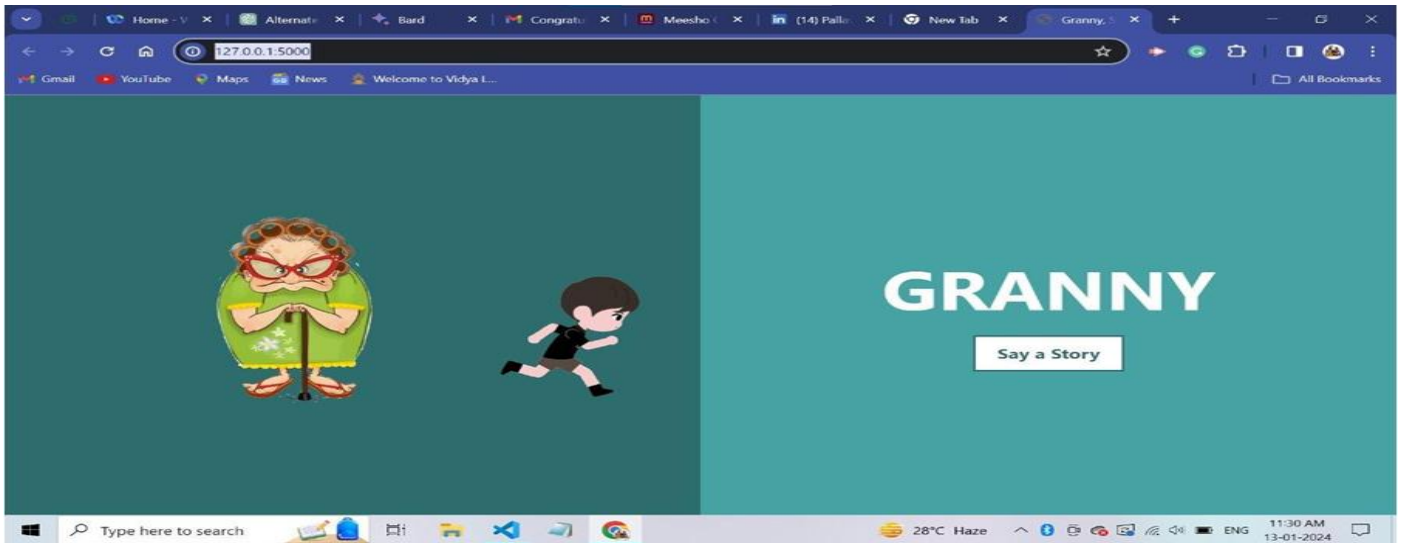


Fig1 Interface of Proposed Model

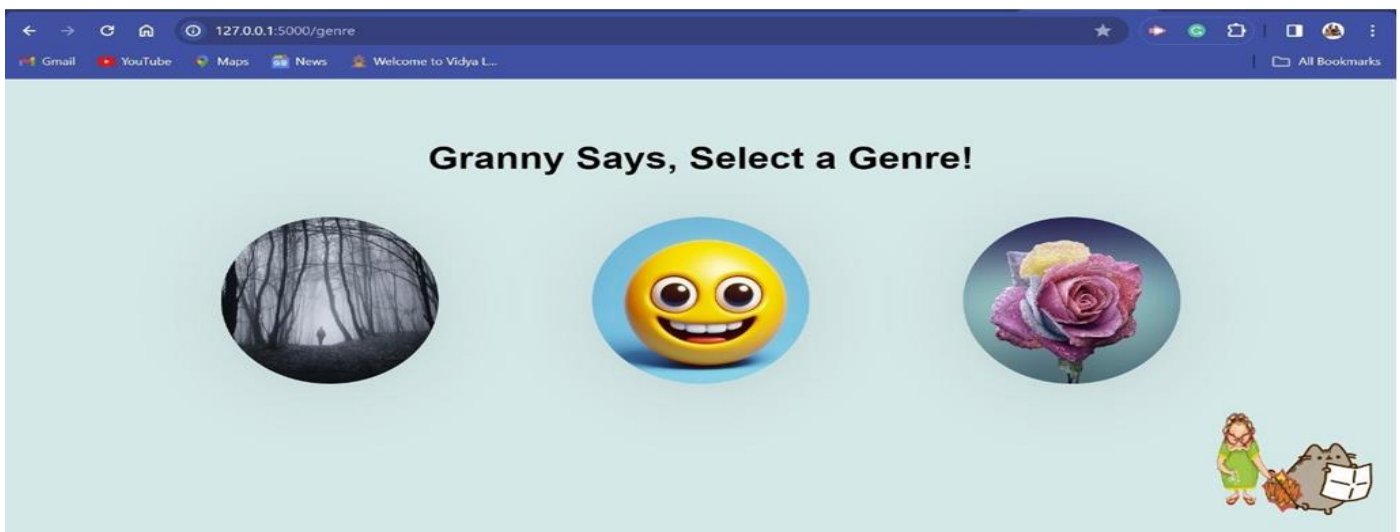


Fig 2 Selecting Genre

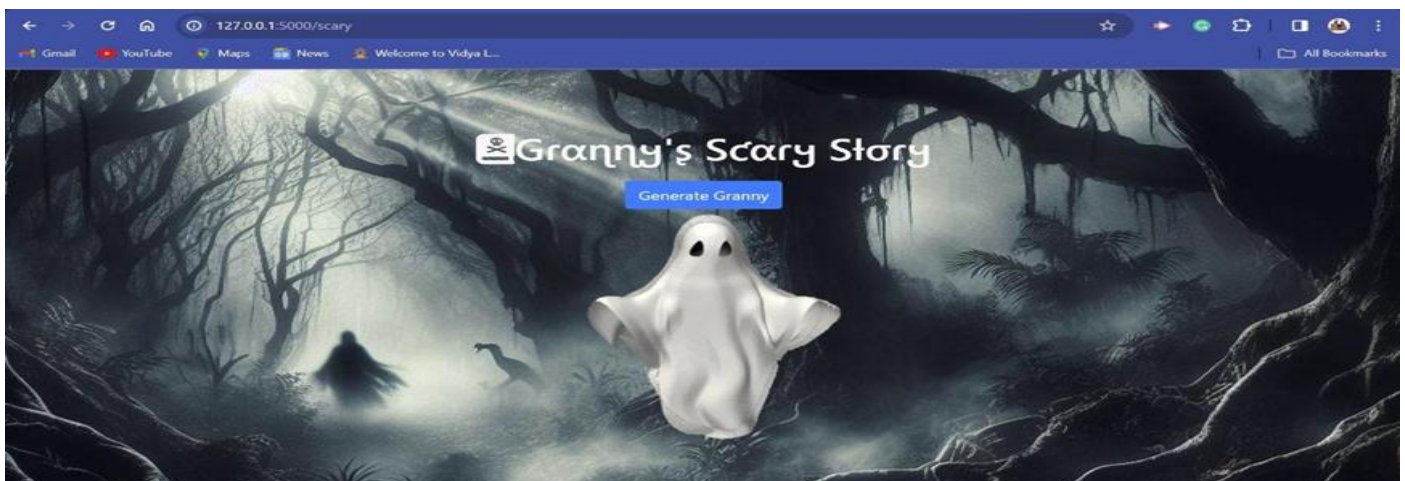


Fig 3 Scary Genre



Fig 4 Funny Genre

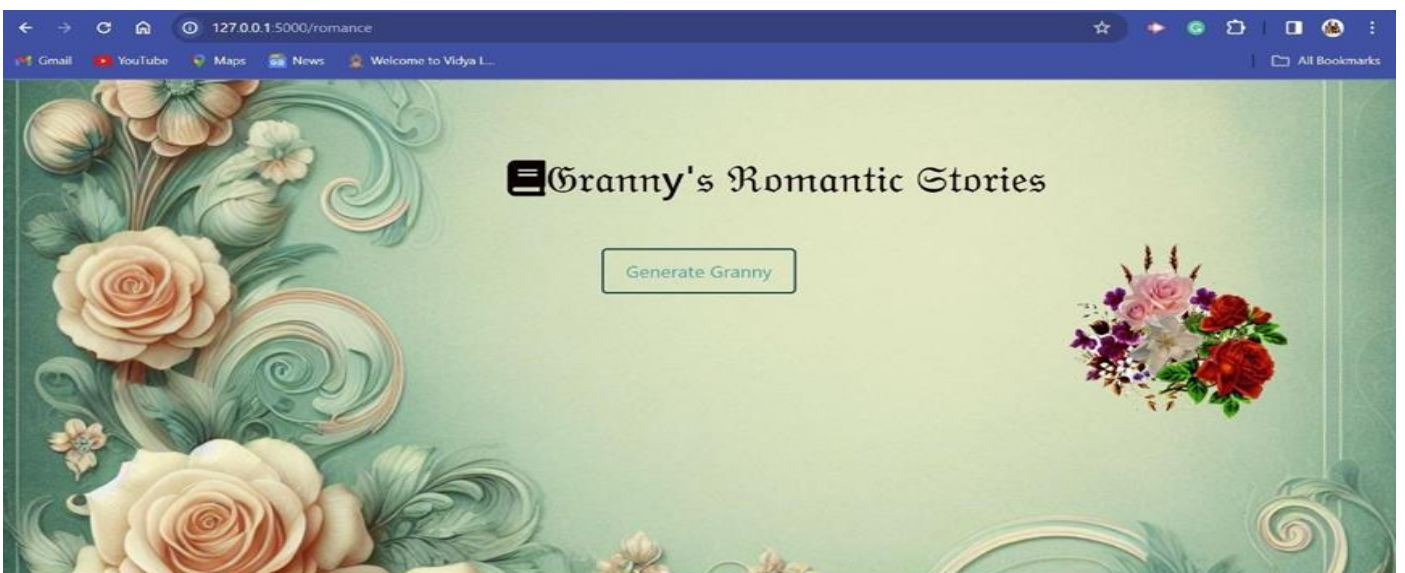


Fig 5 Romantic Genre

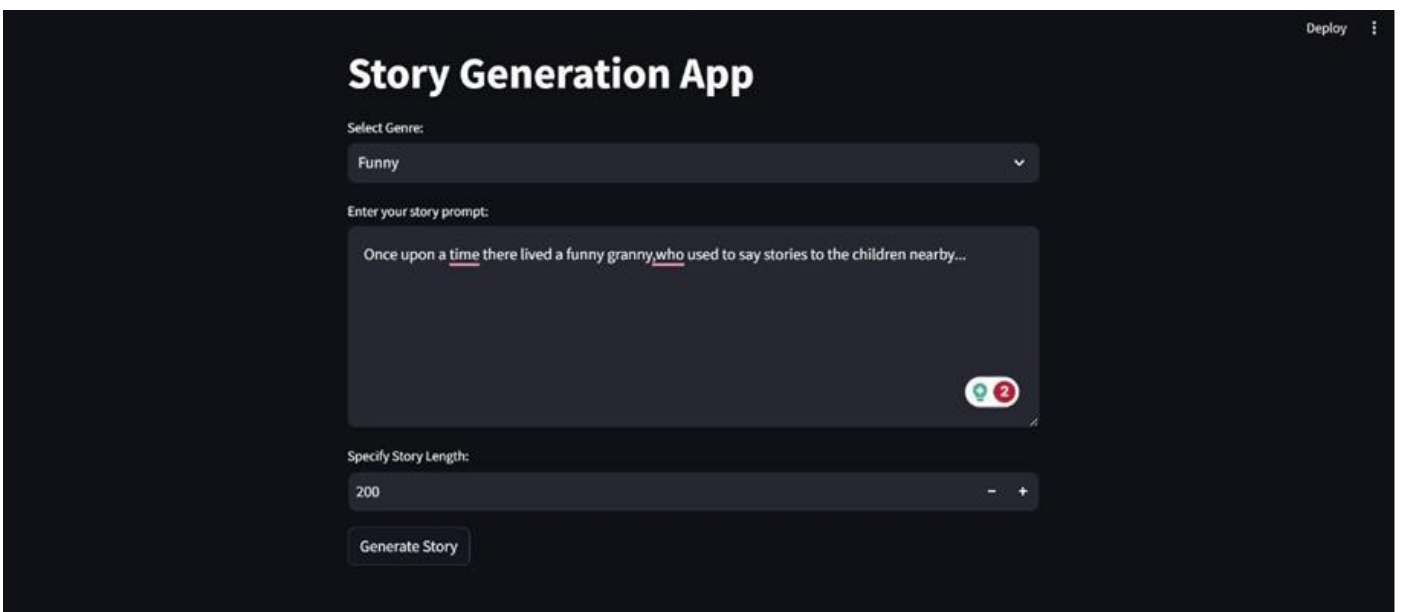


Fig 6 User Inputs

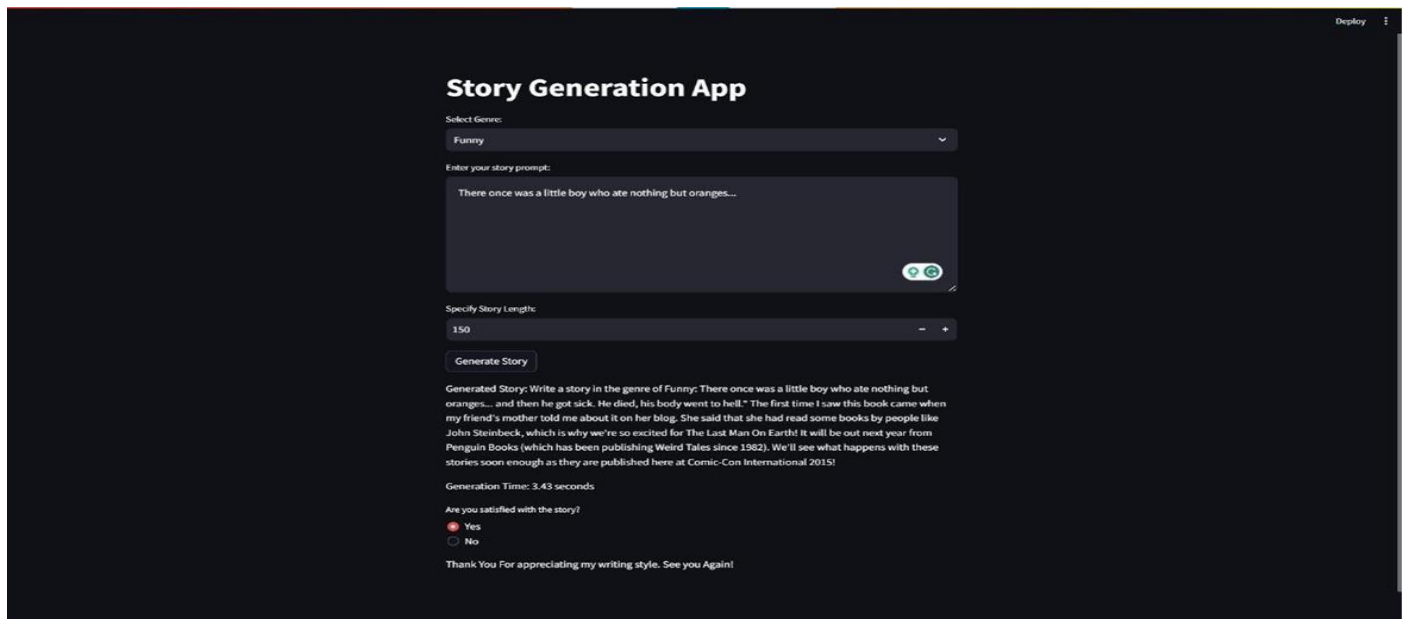


Fig 7 Generated Story

## VI. CONCLUSION

All in all, the coordination of Flask and NLP for story age presents a thrilling an open door to make intelligent and dynamic substance for clients. This framework architecture use the qualities of both web improvement and regular language handling to convey a consistent and drawing in experience. By joining Flask's straight forwardness and adaptability with strong NLP libraries, the application can create stories in light of client prompts or inclinations. In outline, the story age framework utilizing Flask and NLP makes the way for imaginative and intuitive substance creation. This architecture can be used for entertainment, education, or personalized experiences. It can be used to build innovative web-based applications that use the power of natural language processing.

## REFERENCES

- [1]. Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. 2011. Visualization of Time-Oriented Data (1st ed.). Springer Publishing Company, Incorporated.
- [2]. Taketo Akama. 2019. Controlling Symbolic Music Generation based on Concept Learning from Domain Knowledge. In Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019, Arthur Flexer, Geoffroy Peeters, Julián Urbano, and Anja Volk (Eds.). 816-823.
- [3]. Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 6470-6484.
- [4]. Prithviraj Ammanabrolu, William Broniec, Alex Mueller, Jeremy Paul, and Mark Riedl. 2019. Toward Automated Quest Generation in Text-Adventure Games. In Proceedings of the 4th Workshop on Computational Creativity in Language Generation. Association for Computational Linguistics, Tokyo, Japan, 1-12.
- [5]. Prithviraj Ammanabrolu, Wesley Cheung, William Broniec, and Mark O. Riedl. 2020. Automated Storytelling via Causal, Commonsense Plot Ordering. CoRR abs/2009.00829(2020).