AI Story Generator

Arshandh R S[1], Dr. Suresha D[2], Abhay V K[3]
[1,3]*Student, Srinivas Institute of Technology*
[2]*Professor, Srinivas Institute of Technology*
doi.org/10.64643/IJIRTV12I12-206765-459

***Abstract*—AI Story Generator is the most advanced and extensive web-based service carefully created to help users swiftly generate creative, exciting and interesting stories using high-end and advance Artificial Intelligence techniques. Traditional writing techniques have often required a great deal of creativity, preparation, and diligence, making the act of generating stories tedious and overwhelming for students, writers, and numerous content creators. The recommended system has solved this huge problem (a key challenge) by providing an intuitive and simple interface where users can easily type prompt, choose from the set of genres and in an instant create elaborate real stories from their inputs of prompts. It is built with a powerful React frontend, a stable Node. Built on a js and Express backend, tightly integrated with the also state-of-the-art OpenRouter API, the system is capable of generating coherent, meaningful and human-like narratives at mind-boggling speed and efficiencies by leveraging powerful Large Language Models. In addition, the system integrates multiple useful features in our service—genre-specific story generation, a wide variety of prompt customization options, and the ability to interact with users—all elements that have significantly provided more accessibility, interactivity and productivity for/ among different skill levels or types of creative writers.**

***Index Terms*—Artificial Intelligence, AI Story Generation, Natural Language Processing, Large Language Models, Creative Writing, Automated Storytelling, OpenRouter API, React.js, Node.js, Genre-Based Story Generation, Human–AI Collaboration, Narrative Generation, Web-Based Application, Content Creation, Prompt-Based Storytelling.**

I. INTRODUCTION

AI has revolutionized the content industry as far as creation is concerned by providing users with an efficient way of creating content quickly. With the rising role of story creation and writing in education, entertainment, games, and other media, many people find it difficult to write their stories due to issues such as writer's block and lack of motivation. Regular writing processes take up a lot of time and imagination; therefore, they can be quite difficult for students, writers, and others in terms of creating content. Conventional writing software does not assist users with intelligent writing processes. Modern Large Language Models and Natural Language Processing have enabled features like auto-text generation, context awareness, and storytelling based on genres. The technology can create coherent and human-like stories using only prompts. It is helpful for users who struggle to overcome creative barriers and enhance the process of writing. At the same time, there are numerous problems associated with the lack of adaptability and user-friendly interactions within existing solutions. Users find it hard to generate creative stories in different genres within one platform and ensure high-quality results.

The AI Story Generator application was created to facilitate the process of writing and allow users to create engaging stories effortlessly. This solution offers an intelligent and interactive approach to generating stories and uses advanced technologies such as React, Node.js, and Express. All these technologies enable fast communication with the OpenRouter API, which creates an easy-to-use web application. Using this app, users will be able to input various prompts and generate stories according to their preferences. The app enables the generation of stories based on different genres.

The primary aim of AI Story Generator is to develop an easy-to-use and secure platform that not only fosters creativity but helps generate stories efficiently. The combination of AI-based storytelling in one platform makes it easier for users to overcome their creative blockage and be more productive in developing meaningful stories in a hassle-free way. In the coming time, new features like story continuation, customization, multimedia storytelling, and

personalized storytelling can also be added to the platform.

II. LITERATURE REVIEW

AI technology has played an essential role in the domain of automated content creation. Automated narrative and story generation tools have been extensively investigated using Natural Language Processing and transformer-based large language models. Various methodologies have been developed for enhancing story coherence, creativity, personalization, and interactive storytelling with intelligent systems.

In their survey work on automatic story generation approaches, Lukin et al. [1] covered symbolic planning methods, neural network techniques, and case-based reasoning approaches. The authors have provided a thorough discussion on critical issues related to narrative generation, including logical coherence, plot continuity, and character development. The survey has concluded that transformer-based language models became a popular methodology for modern story generators due to the ability to create relevant narratives.

The article by Hou Zhou [2] focuses on studying deep learning techniques applied for story creation purposes. Recurrent neural network, sequence-to-sequence, and transformer methodologies have been analyzed for generating narratives. It has been concluded that transformers outperform other models for creating contextually coherent texts. On the other hand, some limitations, including repetition, loss of long-term memory, and computational issues, have been revealed.

Riedl et al. [3] concentrated on computational narrative generation and highlighted the significance of achieving a balance between human creative abilities and automatic storytelling. The researchers stressed that intelligent automation algorithms should help human imagination instead of fully automating the process of story creation. The authors concluded that interactive narrative systems with prompts, genre selection, and personalized outputs would promote the development of user creativity.

According to Martin et al. [4], event-driven story generation allows for representing story plots in the form of a series of events instead of text. In addition, their model enhanced the logical flow of events and narrative development by predicting them prior to creating readable sentences. However, the proposed approach had difficulty generating emotionally engaging narratives.

In their article, Fan, Lewis, and Dauphin [5] presented the idea of hierarchical neural story generation. The authors proposed creating hierarchies in advance and developing stories using them. As a result, thematic coherence was improved as well as the long-term consistency of the narrative.

The Plan-and-Write approach was proposed by Yao et al. [6], who suggested dividing story planning and story generation into two distinct processes. In this scheme, the AI first produces a storyline and subsequently generates a story based on this plan. The experimental results confirmed the superiority of the proposed technique in terms of coherence, topical relevance, and creativity compared to direct text generation methods.

In addition, Liu et al. [7] presented a character-based storytelling approach in which the AI produced narratives using character data and their personalities predefined by humans. It was found that the focus on characters increased consistency and interest for readers. Nevertheless, interactions between different characters were not easy to maintain.

Recently, advances in LLMs have led to new improvements in AI storytelling approaches. Yuan et al. [8] studied interactive human-AI writing tools that allow users to collaboratively edit AI-generated text. The research indicated that AI writing assistants could relieve writer's block and increase productivity.

Xu et al. [9] examined controllable text generation techniques, which allow the creation of narratives under control through parameters such as genre, tone, sentiment, and style. It was found out that controllable AI tools offer more customization capabilities to users and increase their satisfaction by providing narratives consistent with their preferences.

As can be seen from the literature review, there is a fast progress in developing intelligent technologies for story creation and increasing the scope of their application in various spheres. Current research emphasizes the role of coherence, personalization, controllability, and human-AI interaction in narrative generation. These results serve as the basis for designing the proposed AI Story Generator.

III. REQUIREMENTS

It is vital to establish the requirements for the system in order to successfully implement the AI Story Generator. This section presents the hardware and software requirements necessary for proper operation of the platform. The presented requirements will help the AI Story Generator to efficiently perform all its functions by communicating with users and processing their requests while ensuring performance, stability, and other characteristics.

I. Hardware Requirements

Moderate processing power is required for the computer used to run AI Story Generator, which must allow the system to perform frontend and backend processes simultaneously. A decent CPU, for example, Intel i5 or AMD Ryzen 5, is required. 8 GB of RAM is the minimum amount of memory needed for the system to handle React frontend, backend servers, and APIs. The computer must have at least 20 GB of free space on its drive for keeping project files and dependencies. It is important that the computer is connected to the Internet constantly as this tool needs to communicate with an external API (OpenRouter) in order to perform its function.

II. Software Requirements

The system is implemented through the use of current web technologies like React.js for front-end design and Node.js and Express.js for back-end implementation. The application can run seamlessly under Windows, Linux, and macOS systems. Axios is one of the libraries that will enable the application to communicate with APIs for data transfer from the front-end, back-end, and AI services. The system also utilizes the OpenRouter API that gives it access to powerful Large Language Models able to produce coherent stories. The application development is done in Visual Studio Code, while the use of Git and GitHub helps in managing code versions.

IV. METHODOLOGY

As mentioned above, the core concept behind the AI Story Generator methodology consists of creating an effective workflow for interacting with users, communicating between the AI model, performing backend tasks and presenting the generated story on one platform. Therefore, the aim is to create an efficient system that guarantees high performance and usability for the user, as well as high-quality results. The methodology describes all the key processes related to story creation.

I. Input and Validation

Firstly, the frontend starts with user inputs via interaction by entering story prompts, choosing from different genres, and specifying their preferences for story styles or lengths. The frontend checks whether the input data provided by the users makes sense before passing the request to the backend server. This process helps eliminate any invalid request and minimize unnecessary API calls.

II. Preprocessing

Following the validation of the input, the backend prepares the user's input before passing it to the AI algorithm. In the process of pre-processing, the formatting of the prompt, merging of the genre details, and organization of the prompt into a format ready for AI are done. All unnecessary characters are stripped out at this point to ensure that the output generated is high quality.

III. AI Model Integration

The AI Story Generator utilizes the OpenRouter API integration layer to connect with the advanced large language models used in story generation. In the backend, the prompt is sent after processing and waits for the output from the AI service. The AI model processes the user input by understanding the genre and context and then creates an intelligent story. The integration layer facilitates the process by managing request formation, API communication, response retrieval, and error handling.

IV. Output Processing

Once the story has been generated by the AI model, it goes through the backend process to ensure the story has an improved format and looks good on presentation. It is then checked whether the story has any repetitive parts, and proper structuring is done to make sure it makes sense to the reader once presented at the front end.

V. Frontend Rendering

The final version of the story is then presented to the

frontend interface, which displays it in an organized and easily readable manner. The frontend ensures that titles, paragraphs, and other elements of the story are properly formatted to make the reading experience more enjoyable. Readers have the ability to check the generated output, change the prompts, generate a new story, or write a new story using the interface.

V. DESIGN

The design of AI Story Generator takes into consideration how to design a systematic, scalable, and user-friendly framework which will help manage interactions with users, process the information, and generate stories using artificial intelligence. This way, the design of the software framework can ensure seamless communication between various components while allowing room for future improvements. This framework adopts a modular design whereby each module is responsible for performing unique functions.

I. User Interaction Design

The user interaction design enables an easy-to-use interface that enables the generation of stories by the users without any need for technical skills. The frontend interface enables users to provide inputs through interactive input forms and buttons. The frontend interface enables users to provide prompts, choose story types, and send their requests.

II. Frontend Design

The frontend part of the application is created with the help of React.js, making the interface interactive and responsive. The frontend part is responsible for processing the input from the user and outputting the results back to them along with handling communication between the interface and the backend via HTTP requests. The interface itself is designed with well-organized layouts and typography along with being responsive.

III. Backend Design

The backend is built using Node.js and Express.js in order to perform server-side actions and communicate with the AI smoothly. The backend accepts requests sent by the frontend, validates them, prepares prompts for sending to the OpenRouter API, and performs any additional actions required to establish a successful connection. This includes error handling, processing the response, and transferring data back and forth between the frontend and the AI.

IV. AI Integration Design

The integration layer of the AI component takes care of integrating the app with sophisticated LLMs using the OpenRouter API. This layer takes care of handling prompt submission, response extraction, and secure and efficient communications via the API. The AI model analyzes the prompts submitted by users and produces narratives depending on the genre chosen by the user.

V. Output and Display Design

Output design mainly emphasizes generating stories that appear clean and attractive to read. When an AI-generated story has been generated at the backend, the front-end ensures that it is presented in a neat form through the use of paragraphs and formatting. Output display allows users to view, copy, regenerate or change the generated story as required.

VI. System Architecture Design

The architecture of the whole system is based on layers, which include the frontend layer, the backend processing layer, and the AI model layer. The frontend layer receives user inputs and forwards requests to the backend layer, whereas the backend layer communicates with the OpenRouter API in order to produce stories. The generated stories using AI models are forwarded to the frontend layer, where they can be easily displayed to users.

VI. RESULTS

The successful development and testing of the AI Story Generator took place, and it proved to be able to produce meaningful stories through prompts provided by the users. It was found that the integration of the React frontend, the Node.js backend, and the OpenRouter AI models works well to enable a good user interaction experience.

I. Homepage Analysis

The home page of the Artificial Intelligence Story Generator is easy to navigate, allowing users to input prompts and select story types effortlessly.

The interface was created keeping in mind the convenience of users who lack technical skills. Testing

found that users could use the system without any difficulty.

II. Story Generation Evaluation

The narrative creation engine was successful in creating full narratives depending on different user inputs in the form of prompts and categories of stories that the users wished to generate. The output stories created by the AI engine had correct sentence structure, logical structure, and were creatively constructed. The AI tool was tested using various story genres like fantasy, horror, adventure, and science fiction, among others.

III. System Performance and Integration

The connection between the frontend, the backend, and the AI services worked seamlessly in all tests carried out. The users' requests were served efficiently, and stories were returned in a short while. The backend performed perfectly without failing in communicating through APIs, validation of inputs, and response formatting. Any errors or problems caused by the API were shown to the users in form of error messages.

IV. User Experience Analysis

The platform gave an interactive experience while undergoing the test. Users were able to create, read, and recreate the story using the application. There were improvements made in terms of the readability of the generated story because of the frontend formatting used. The feedback given by the users revealed that the system is user-friendly.

V. Overall Discussion

In summary, the results obtained demonstrate that the AI Story Generator is able to effectively accomplish its core purposes of helping users develop creative and valuable stories while minimizing efforts. Thus, one can see the practical implementation of Artificial Intelligence in the field of creative writing. Moreover, the developed flexible structure of the project may facilitate some possible future modifications that can include, but are not limited to, personalized storytelling, custom character design, multimedia inclusion, as well as developing story continuity tools.

VII. CONCLUSION AND FUTURE SCOPE

AI Story Generator is an intelligent and interactive platform that allows you to create stories with the help of Artificial Intelligence. The application combines modern web technologies with advanced language models to offer features like story generation according to the genre, prompts customisation, and quick and effective AI story creation in one place. Using the application makes it easier to write creatively and overcome problems associated with writer's block. The use of React, Node.js, Express, and OpenRouter API makes the application scalable, responsive, reliable, and maintainable.

There are various possibilities for future development of the system. For example, the application can be made even more personalized and interactive. There are many features that could be added to the platform in the future, such as story continuation, character customisation, voice-based story narration, story translation into several languages, and story illustration using AI. Another idea for future improvements could be adding user accounts, cloud story storage, collaboration, and mobile applications to the platform.

REFERENCES

[1] S. Lukin *et al*., "A Comprehensive Review of Automatic Story Generation Methods," *ACM Comput. Surveys*, vol. 54, no. 5, pp. 1–38, 2021.

[2] H. Zhou, "Deep Learning Techniques Applied in AI-Based Story Generation," J. Artif. Intell. Res., vol. 70, pp. 1–29, 2021.

[3] M. O. Riedl et al., "Computational Narrative Authoring and Story Generation Approaches," IEEE Trans. Comput. Intell. AI Games, vol. 11, no. 3, pp. 1–15, 2019.

[4] A. Alabdulkarim et al., "Challenges and Recent Developments in Automatic Storytelling Systems," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 4, pp. 112–120, 2021.

[5] L. Martin et al., "Event-Based Models for AI Story Generation," in Proc. AAAI Conf. Artif. Intell., vol. 31, no. 1, pp. 868–875, 2017.

[6] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical Approaches for Neural Story Creation," in Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL), pp. 889–898, 2018.

[7] L. Yao et al., "Plan-and-Write Framework for Improved Automatic Storytelling," in Proc. AAAI Conf. Artif. Intell., vol. 33, no. 1, pp. 7378–7385, 2019.

[8] A. Fan et al., "Structured Techniques for Long-Form Story Generation," in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol. (NAACL-HLT), pp. 2650–2660, 2019.
[9] D. Liu et al., "Character-Oriented Neural Networks for Automated Narrative Generation," IEEE Access, vol. 8, pp. 121001–121012, 2020.
[10] P. Xu et al., "Controlled Story Generation Using External Knowledge Sources," in Proc. Conf. Empirical Methods Natural Language Process. (EMNLP), pp. 970–980, 2020.
[11] N. Peng et al., "Reader-Guided Neural Narrative Generation Models," Trans. Assoc. Comput. Linguistics, vol. 9, pp. 1132–1148, 2021.
[12] M. Yuan et al., "Large Language Models for Interactive Story Writing," in Proc. CHI Conf. Hum. Factors Comput. Syst., pp. 1–15, 2022.
[13] "Survey of Large Language Models in Story Generation Applications," Findings of the Association for Computational Linguistics: EMNLP, pp. 1–22, 2025.
[14] W. Liang and Y. Wang, "Survey on Controlled Text Generation Using Large Language Models," Artif. Intell. Rev., vol. 57, no. 2, pp. 1–35, 2024.
[15] X. Wang et al., "Structured Knowledge-Based Open-World Story Generation," IEEE Trans. Knowl. Data Eng., vol. 35, no. 7, pp. 6541–6553, 2023.
[16] M. Araz, "Automated Story Generation Using Transformer Neural Networks," Int. J. Comput. Appl., vol. 176, no. 40, pp. 22–29, 2020.
[17] M. Roemmele et al., "Automated Evaluation Methods for Story Generation Systems," in Proc. AAAI Workshop Intell. Narrative Technol., pp. 1–8, 2017.
[18] J. Hong et al., "Character-Grounded Storytelling with Visual Writing Prompts," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1440–1450, 2023.
[19] H. Kim et al., "Controllable and Multi-Modal Narrative Generation Systems," IEEE Access, vol. 10, pp. 77452–77465, 2022.
[20] S. Megala et al., "Review of Personalized AI-Based Story Generation Systems," Int. J. Innov. Res. Comput. Sci. Technol., vol. 12, no. 1, pp. 45–52, 2024.