



Storybook Maker: A Generative AI Based Approach for Automated Storybook Generation

Prerna Agrawal¹, Arman Dholiya², Pooja Vani², Om Sathiya², Devvrat Vyas², Vikas Sheth², Harsh Makwana²

¹Assistant Professor, FCAIT-PG, GLS University, Ahmedabad, India. E-mail: Prerna013@gmail.com

²Student, FCAIT-PG (MCA), GLS University, Ahmedabad, India.

Abstract

Generative Artificial Intelligence (Generative AI) has rapidly changed digital content creation by allowing automated generation of text, images, and multimedia content from natural language prompts. Nevertheless, current storytelling platforms that are AI-based are typically limited to a single function, thus requiring the user to use multiple apps for illustration, translation to multiple languages, narration, and layout. In this contribution, we introduce Storybook Maker, a web-based generative AI architecture for the automated generation of digital storybooks that integrates several AI services into a common platform. The proposed tool employs Gemini 3 Flash Preview for story generation and FLUX.1-dev for text-to-image synthesis while allowing translation to any other language, text-to-sound conversion, and PDF storybook generation. Software implementation will be conducted using Python, Django, and MySQL with modular architecture. An extensive evaluation was performed using various LLMs and diffusion-based text to image models. The preliminary findings show that Gemini 3 Flash Preview is the best-performing system in terms of story generation, according to BLEU, ROUGE, BERTScore, and human scores. Conversely, FLUX.1-dev is better than any other known diffusion-based model if considered in terms of CLIP Score, ImageReward, SSIM, PSNR, and Human Evaluation. Thus, an efficient, complete website was created for AI supported digital storytelling that allows making story creation easier.

Keywords - Generative Artificial Intelligence, Digital Storybook, Story Generation, Large Language Models, Text-to-Image Generation, Gemini 3 Flash Preview, FLUX.1-dev, Django.

1. Introduction

Among the disruptive technologies that have arisen in recent times is generative artificial intelligence (Generative AI), which refers to intelligent systems that can produce high-quality text, images, sound, and other digital assets from simple prompts expressed in natural language. Research on Large Language Models (LLMs) and diffusion-based technologies for image creation has resulted in the increase in the scope of application of digital content generation methods facilitated by AI. Hence, Generative AI technologies allow for creation in different spheres such as education, medical treatment, entertainment, digital publishing, etc. Moreover, introducing these technologies leads to an enormous decrease of time and efforts for creating quality content with more options for people. Although many other domains are subjected to the implementation of the technology, AI-powered storytelling is the best known application of the technology, which provides users with an opportunity to create interesting stories just from user prompts. The modern AI-based language models Gemini, GPT, and LLaMA can process the information and produce texts, taking into account the styles and contexts of the stories and creating narratives that are relevant to the genre. The new technologies of text-to-image generation allow for the creation of such intelligent systems, which could produce complete digital picture books and develop the digital storytelling process. Nevertheless, the vast majority of the existing storytelling platforms usually focus only on storytelling and thus do not provide the customer with all possible features of the storytelling system. Thus, users should turn to using various independent applications for creating illustrations, translating stories, creating audio stories, formatting storybooks, and delivering the content to be shared. To fill this gap, the authors of the study proprietary application – Storybook Maker, a web application based on generative AI. Thus, the proposed application allows users to create digital storybooks just from a simple prompt using the selected genre of the story. Storytelling is conducted via the Gemini 3 Flash Preview model; illustrations are created with the help of FLUX.1-dev technology. Apart from storytelling and illustrations, the platform includes translation, audio storytelling, and generation of PDF versions of storybooks. The system is created with Python, Django, and MySQL, using a modular approach.

The proposed Storybook Maker offers several significant contributions:

- An end-to-end web-based system for automated digital storybook generation.

- Prompt-based story generation using the Gemini 3 Flash Preview model.
- Integration of AI-powered story generation, illustration generation, multilingual translation support, audio narration, and PDF export within a unified platform.
- Dedicated user and administrator modules for efficient content creation and management.
- A modular and extensible architecture that facilitates future integration of advanced generative AI technologies.

The outline of the rest of the paper is as follows: Section 2 describes recent studies on AI-backed storytelling methods and generative AI-based content generation. Section 3 explains the architecture of the Storybook Maker and the methodology used to develop it. Section 4 provides the description of the experimental setup that was used for conducting the experiments. Section 5 presents the results of the experiments and performance evaluation. Section 6 describes the GUI screens of the Story Maker application, and section 7 explains the conclusion and the future scope.

II. Related Work

The advent of generative artificial intelligence (generative AI) has accelerated the exploration of storytelling by using large language models (LLMs). Initially, storytelling systems revolved around structured story generation, using certain rules and templates, which would generate repetitive stories that were only correct grammatically [5]. While the stories produced in this way were correct from the grammatical perspective, they could not produce the necessary variation and interest in the content. The use of transformers has changed the game of story generation; now with models such as GPT, T5, LLaMA, and Gemini, AI understands the user's intention, and the story can be created for a variety of different genres [6]. With these new models, the creation of stories became accessible and in turn opened a wide variety of educational applications in the art of storytelling. Recent findings have been focused on multimodal storytelling since newly developed technologies make use of multimodal story creation, combining story generation with image generation and translation as well as TTS approaches [7]. Such methods significantly improve the quality of the end product and make stories visually engaging. At the same time, the existing technologies are mostly quite specific, meaning that users need to switch between technologies. As it can be easily concluded, the necessity of the existence of a single technology that can combine all of the above-mentioned functionalities is clear. This need has given rise to the creation of the Storybook Maker that will integrate the abovementioned functions into a single web application. The architecture of the system being developed is modular in its core and allows the application of the Gemini 3 Flash Preview system for story generation and the innovative system of FLUX.1 dev for illustration generation [8].

III. Proposed System Architecture

The proposed Storybook Maker is an intelligent online application that provides automated end-to-end digital story creation by integrating various generative artificial intelligence systems onto a single platform. It features a modular client-server architecture that allows its users to create stories, regenerate them in different variations, translate them into different languages, generate audio narration, and create PDF storybooks through a single web interface. The application is developed with the use of the Python programming language and Django framework, while MySQL serves as a means of storing both user data and story content generated [14, 15]. The architecture features multiple independent modules that interact with each other via the centralized application server. This modular architecture increases scalability, simplifies the process of maintaining the system, and allows the introduction of new AI services without affecting the existing features of the system. The application server assures the effective operation of the technology by providing the interaction between the interface of the project, the database, and the AI models, which assures the successful completion of all the tasks of storytelling.

The Storybook Maker launches when a person registers and logs into the application. The process of registration includes the verification of the user data before being granted access to the application interface. After the successful authentication, the user launches a natural language prompt while choosing the story genre from the assortment of options. The application server receives the prompt, checks its validity and forwards it to the Gemini 3 Flash Preview algorithm [4]. The result of the generated narrative is returned to the server, demonstrated through the web interface and saved in the MySQL database for further use. In case the output does not meet the user's expectations, the Regenerate Story feature can be used to generate other alternatives with the same prompt while keeping the content consistent. After the story is generated, the application also offers some additional services powered by AI that will help the user use the generated story as a complete writing piece that can be presented as a digital storybook. Thus, the user can receive illustrations for the key scenes of the narrative with the help of the FLUX.1-dev image generation model [12]. Besides, the generated story can be translated into different languages, turned into speech by means of the audio narration module, and converted into a well-formatted PDF. All the services function independently and in coordination with the application server, which allows the accomplishment of different tasks of content generation without changing the applications. The system also has a special

administrator module that enables secure access to administrative functions. After the authentication, administrators have a chance to view the registered users of the application and check the generated stories, thus being able to view, update, or delete the records of the users and generated stories. Therefore, the administrator interface is efficient in terms of content management functions. The centralized server is in constant touch with the database used to store and retrieve all the relevant user information as well as prompts for the generation of stories, translated results, and PDF files. It is also used for making requests and getting responses from the models and algorithms of AI, which provides for the effective use of the service for the future integration of AI systems.

A) System Architecture

Figure 1 provides an overview of how the Storybook Maker application works. The entire system consists of six essential components: the User Module, Administrator Module, Application Server, Database Module, Generative AI services, and the supporting service modules.

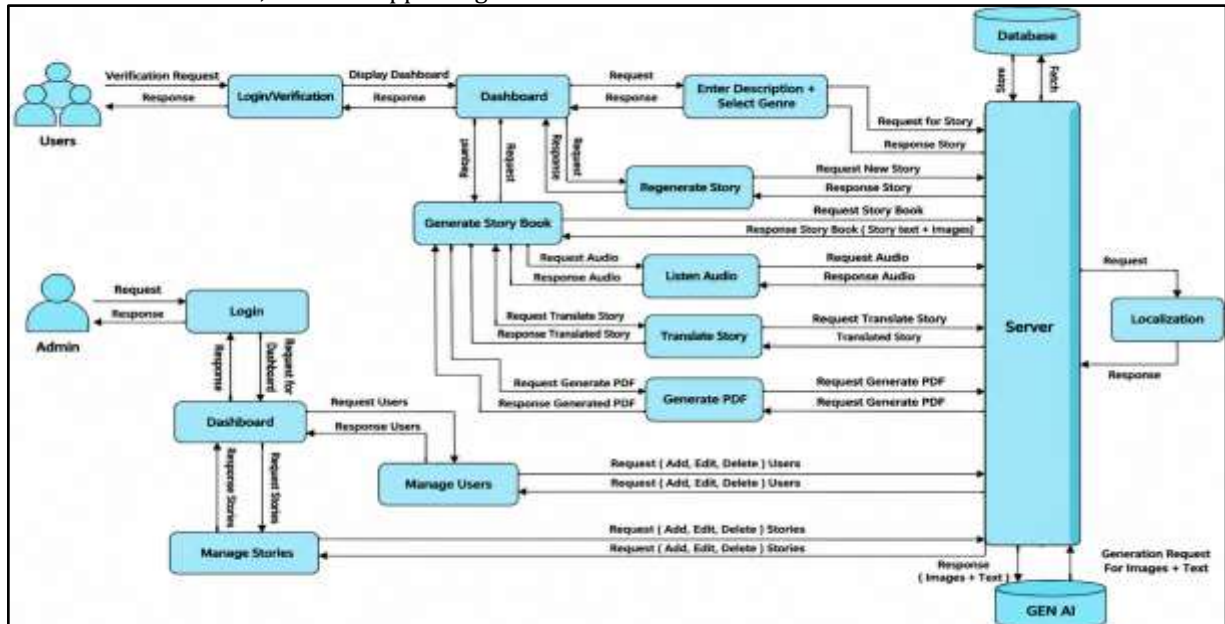


Figure 1. Overall architecture of the Storybook Maker application

The User Module gives users the ability to register, log in, access the main panel, and provide a story. prompts, and pick genres. It also allows users to generate stories, regenerate other stories, translate stories into other languages, create narrations for stories, and download the finished stories in PDF format. The application server serves as the heart of the system. It is responsible for validating incoming requests from users, managing communications with the external artificial intelligence services, working with the received responses, handling different transactions in the database, and providing the final result for users. The server was developed using the Django framework and makes use of the MySQL database for data storage [14, 15]. The Generative AI Module is responsible for the main generation process. The story generation is accomplished with the help of the large language model known as the Gemini 3 Flash Preview, which is capable of coming up with stories based on natural language prompts [4]. As for the illustrations, they are created thanks to the text-to-image model FLUX.1-dev. The images correspond to the significant parts of the story [12]. In addition, there are more AI modules responsible for the translation of the text into multiple languages, text-to-speech narrations, and PDF book generation. The database module stores information about users, such as credentials, prompts, stories generated, translations, sound files, PDFs, and admin records using MySQL technology [15]. Having centralized data storage makes it possible for users to retrieve necessary information more easily. The Admin Module provides an opportunity to carry out admin functions. This module allows control of what users are doing, management of stories created, storage of application records, and management of the content. Separating administrative and user functions contributes to improving overall application security and simplifying the entire application management process.

In this way, the new architecture includes a user interface, AI-powered content generation, database management, and administrative work in one unified application. The modular design of the system allows all competing parts to get in touch with each other and ensures scalability as well as flexibility of the application for future implementations of generative AI technologies. The next section describes the experimental setup used for performing the experimentation.

IV. EXPERIMENTAL SETUP

The Storybook Maker system has been developed as a web platform for testing the capabilities of generative AI in automatic digital storybook production. Python was utilized as the programming language and Django as the web framework for building this application. MySQL was used for the backend database for user information, created prompts, generated stories, and other system data [14, 15]. The interface was created with HTML, CSS, and JavaScript and offers a friendly and easy-to-use interface.

The core of the storytelling engine is based on the Gemini 3 Flash Preview model that is used for the generation of coherent stories with appropriate context based on user input and selected story genre [4]. After creation, stories can be displayed in the web interface and stored in the database for future processing and use. Another advanced component of this system is the image generative module based on FLUX.1-dev, which provides the functionality of generating illustrations for the text of the generated stories [12]. The application also includes the components responsible for translation, conversion of text into speech, and PDF generation, which means that the user can create the entire digital storybook using one tool. This modular architecture makes it possible for these automated services to function independently from one another while allowing the seamless implementation of new developments in the future.

In order to conduct an evaluation of the system, testing was performed based on the prompts belonging to different storytelling genres: fantasy, adventure, mystery, science fiction, romance, horror, and children's stories. The outputs produced in response to the prompts were evaluated against a number of automatic evaluation criteria as well as based on the views of the users to measure coherence, relevance of context, consistency of meaning, creativity, and quality of the produced story.

A) Development Environment

The planned Storybook Maker was created with the use of the Django web technology and Python. programming language, thereby establishing a powerful, consistent, and secure base to create web applications. The platform can handle requests, provide backend processing, allow routing of URLs, use authentication, and integrate easily with relational databases, being a good candidate for the development of applications using AI technologies. A MySQL relational database management system was selected for maintaining user data, prompts, original texts, translations, graphics created, and PDFs generated [15]. The storytelling generation module works with the help of the Gemini 3 Flash Preview LLM, capable of creating stories based on natural language prompts issued by users. The multilingual support for the translation of the story in other languages such as Hindi, English, Gujarati, and Marathi is done by the GTTS library. For PDF generation and download, the ReportLab library is used. Table 1 gives an outline concerning both software and hardware systems that have been used in the development process.

Table 1. Development Environment used for StoryBook Maker

Component	Specification
Programming Language	Python
Backend Framework	Django
Frontend	HTML, CSS, JavaScript
Database	MySQL
Story Generation Mode	Gemini 3 Flash Preview
Image Generation Mode	FLUX.1-dev
Multilingual Support	GTTS
PDF Generation and Download	Reportlab

B) Dataset Used

The testing of the suggested Storybook Maker framework was carried out by utilizing a rich mixture of stories, which included children's stories obtained from various free online sources. The dataset used consisted of the Children's Stories Dataset that was made available through Hugging Face [16], Project

Gutenberg [17], and similar platforms that provide open-access children's stories and those generated by LLMs (large language models). A total of 25000 stories were used for the Storybook Maker. The stories are classified under many genres, including fantasy, horror, adventure, mystery, science fiction, romance, and children's stories. Before using the dataset, it was preprocessed by removing duplicate entries, fixing formatting errors, filtering incomplete records, and sorting stories into genres. The processed dataset was primarily utilized in prompt engineering, qualitative comparison, and evaluation of generated stories, rather than for training purposes. Table 2 provides a summary of datasets used in the design and evaluation of the suggested Storybook Maker framework.

Table 2. StoryBook Maker Dataset

Source	Description	Approximate Number of Stories
Children Stories Dataset (Hugging Face) [16]	Public dataset containing children's stories across multiple genres	7,965
Project Gutenberg [17]	Public domain collection of classic literary stories	4,035
Synthetic Stories (LLM-generated)	Stories generated using Large Language Models	13,000
Total	Combined dataset used during development and evaluation	25,000

C) Evaluation Metrics

To find the best large language model (LLM) for automated story writing and the best model for text-to-image evaluations, comparisons were conducted between state-of-the-art LLMs. For story generation, metrics such as BLEU, ROUGE (1, 2, L), BERTScore (recall, precision, F1), perplexity, human evaluation score, and average quality scores were used. To find the best diffusion model for the text-to-image generation, metrics such as CLIP Score, ImageReward, LPIPS, SSIM, PSNR, and Human Evaluation were used. Table 3 and Table 4 describe the different evaluation metrics for story generation and text-to-image generation. The next section describes the results of the experimentation conducted based on the evaluation parameters.

Table 3. Evaluation Metrics for LLM-Based Story Generation

Metric	Purpose
BLEU	Measures lexical similarity between the generated story and the reference story using n-gram matching. Higher values indicate better word-level correspondence.
ROUGE-1	Measures unigram overlap between generated and reference stories, indicating content coverage.

ROUGE-2	Measures bigram overlap to evaluate phrase-level similarity.
ROUGE-L	Measures the longest common subsequence to evaluate structural and sequential similarity.
BERTScore (Precision, Recall, F1)	Measures semantic similarity using contextual embeddings from pretrained language models. Higher scores indicate better semantic consistency.
Perplexity	Evaluates the fluency and predictability of the generated stories. Lower perplexity indicates more natural language generation.
Human Evaluation Score	Stories are manually evaluated based on narrative coherence, creativity, readability, and relevance to the input prompt using a five-point rating scale.
Average Quality Score	Overall quality score computed by combining automatic and human evaluation results.

Table 4. Evaluation Metrics for Text-to-Image Generation

Metric	Purpose
CLIP Score	Measures the semantic alignment between the generated image and the input text prompt.
ImageReward	Evaluates the overall quality and prompt relevance of the generated image based on human preference learning.
LPIPS	Measures perceptual similarity between the generated image and the reference image.
SSIM	Measures the structural similarity between the generated image and the reference image.
PSNR	Measures the reconstruction quality and image fidelity relative to the reference image.
Human Evaluation Score	Assesses the generated image based on visual quality, prompt relevance, creativity, and overall aesthetics.

Average Quality Score	Represents the overall image quality obtained by averaging the human evaluation scores.
------------------------------	---

V. Results And Discussion

The effectiveness of the Storybook Maker was evaluated by means of studying how efficient it is in producing textual and visual stories. To this end, large-scale comparative analysis of different innovative large language models (LLMs) and text-to-image generation diffusion models was performed. The assessment was carried out using automatic evaluation metrics. In light of the obtained results, the best LLM models and image-producing models were picked up for integration into the Storybook Maker concept.

A) Automatic Story Generation Results

Table 5 presents details on the performance of the Large Language Models (LLMs) monitored along with the different measuring metrics from Section 4.3. The following comparison represents the Gemini 3 Flash Preview, Gemini 1.0 Pro, Gemini Pro, Phi-2, TinyLlama 1.1B Chat, GPT-2, Falcon RW 1B, and FLAN-T5 Base, which are the most commonly used language models in the process of text generation as well as natural language processing tasks [2, 4–7].

According to Table 5, the Gemini-3 Flash Preview model presents the best performance out of the rest of the evaluated models, with the biggest BLEU score (0.1915), ROUGE-1 (0.6004), ROUGE-2 (0.3008), ROUGE-L (0.4448), BERTScore-F1 (0.8745), and Human Evaluation Score (1.0) obtained by the model. Besides, the Gemini 3 Flash Preview has achieved very competitive perplexity (59.92), showing that the model produces quite fluid discourse. Furthermore, the processed model also attains the highest Average Quality Score (0.6684), which indicates its better lexical similarity, semantic consistency, contextual relevance, and overall quality in comparison to the other LLMs. Therefore, based on these experimental results, the Gemini 3 Flash Preview model was chosen as a story-generating model in the Storybook Maker.

Model	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BERT Score Precision	BERTScore Recall	BERTScore F1	Perplexity	Human Evaluation Score	Average Quality Score
Gemini 3 Flash Preview	0.1915	0.6004	0.3008	0.4448	0.8733	0.8759	0.8745	59.9174	1	0.6684
Gemini 1.0 Pro	0.0797	0.4750	0.1711	0.3187	0.8481	0.8322	0.8400	78.2569	0.9428	0.5837
Gemini Pro	0.0328	0.4192	0.1099	0.2496	0.8336	0.8133	0.8233	71.9294	0.8571	0.5391
Phi-2	0.0446	0.4409	0.1563	0.2904	0.8664	0.8098	0.8371	85.4422	0.6571	0.5237
TinyLlama 1.1B Chat	0.0085	0.3280	0.0991	0.2334	0.8653	0.7723	0.8162	56.9394	0.5714	0.4794
GPT-2	0.0053	0.2806	0.0637	0.2024	0.8043	0.7303	0.7655	61.7825	0.4286	0.4282
Falcon RW 1B	0.002	0.2183	0.041	0.161	0.8077	0.7428	0.7739	76.301	0.3429	0.3949
FLAN-T5 Base	0	0.1215	0.0346	0.1029	0.8375	0.6903	0.7568	220.8649	0.3714	0.3269

Table 5. Evaluation Results of Automatic Story Generation LLMs

B) Text to Image Generation Results

The analyzed text-to-image diffusion models are illustrated in Table 6 according to the metrics explained in Section 4.3. The comparison contains FLUX.1-dev, SDXL, Stable Diffusion 1.5, Kandinsky 2.2, DeepFloyd IF, and PixArt- α , which represent the famous diffusion-based models used in text-to-image synthesis [12, 13]. As illustrated in Table 6, FLUX.1-dev shows the best performance, obtaining the highest score according to CLIP (0.93), ImageReward (0.91), SSIM (0.94), PSNR (31.4), Human Evaluation (4.8), and Average Quality Score (0.94) while also obtaining the lowest LPIPS value (0.10), suggesting its great ability

for providing perceptual similarity. This proves that FLUX.1-dev is able to create images with great semantic matching, visual fidelity, and consistency with the prompts. This comparison leads to the selection of FLUX.1-dev as the text-to-image generation model used in the proposed Storybook Maker architecture. Table 7 represents different images generated by the diffusion models. The next section describes the GUI screens of the Story Maker web application.

Model	CLIP	ImageReward	LPIPS	SSIM	PSNR	Human Score (Out of 5)	Avg. Quality
FLUX.1-dev	0.93	0.91	0.1	0.94	31.40	4.80	0.94
SDXL	0.90	0.88	0.13	0.91	29.70	4.50	0.89
Stable Diffusion 1.5	0.85	0.82	0.19	0.86	27.10	4.10	0.83
Kandinsky 2.2	0.88	0.84	0.17	0.88	28.20	4.20	0.85
DeepFloyd IF	0.91	0.90	0.11	0.93	30.80	4.70	0.91
PixArt- α	0.89	0.86	0.15	0.89	29.00	4.30	0.87

Table 6. Evaluation Results of Text to Image Generation Diffusion models



FLUX.1-dev model



DeepFloyd IF model



SDXL model



PixArt- α model



Kandinsky 2.2 model



Stable Diffusion 1.5 model

Figure 2. Generated images from text by different diffusion models

VI. GUI SCREENS OF THE STORYBOOK MAKER

This section provides a number of example screenshots of the developed Storybook Maker web application. The user interface shows how the system architecture is put into practice. In particular, it allows for user registration and authentication, prompt-based story generation, automatic picture creation through AI, multilingual story translation, audio narration, and PDF storybook creation. The screenshots demonstrate how different generative AI components work together within one unified, interactive platform, creating a fully automated digital storybook production. Figure 3 represents the GUI screenshots of the StoryMaker web application. The next section explains the conclusion and the future scope.

127.0.0.1:5000



AI Storybook Maker

Turn your imagination into magical illustrated stories.

Story Description

Genre

Fantasy

Story Length

Short

Generate Story

Generated Story

Title:

The Moonlit Wolf and the Sapphire Sparrow

Page 1:

In the kingdom of Ever-Glow, where the trees hummed soft lullabies, lived Fenris, a giant wolf with fur of spun moonlight. Despite his size, Fenris was as gentle as a summer breeze, yet he carried a heavy secret in his heart. He had lost his howl to the Shadow-Mist, leaving him silent in a forest that thrived on music. He sat alone by the Crystal Brook, watching the glowing fish dance, wishing he could join the midnight chorus.

Page 2:

Suddenly, a tiny spark of blue fluttered down from the ancient Silver Oak. It was Pip, a sparrow whose wings were crafted from enchanted sapphire glass. Pip did not chirp in ordinary sounds; instead, he sang in tiny, shimmering bubbles of light that floated in the air. Seeing the wolf's sad eyes, the little bird landed right on Fenris's velvet ear. Pip knew of a place where lost sounds were kept: the Star-Catcher Peak, hidden behind the Veil of Dreams.

Page 3:

Fenris stood up, his massive paws treading softly on the glowing emerald moss. With Pip perched securely atop his head, the unlikely pair began their journey through the shifting fog. The forest changed around them, with flowers that blossomed into tiny lanterns and vines that moved like playful snakes. Pip used his glass wings to catch the stray beams of moonlight, reflecting them forward to light a path through the deepest, darkest thickets of the enchanted woods to find the way.

Page 4:

They reached the base of the mountain, where the wind sang in riddles. To pass, they had to cross the Bridge of Echoes, a shimmering path made of frozen whispers. The bridge was fragile and would shatter if any loud sound was made. Fenris held his breath, moving with the grace of a quiet shadow, while Pip tucked his wings tight to stay perfectly silent. Together, they balanced on the edge of the world, trusting each other completely while ancient magic swirled.

Page 5:

At the summit stood the Fountain of Melodies, a swirling pool of liquid starlight where every lost sound in the world gathered. Pip dove into the glowing water, his sapphire wings splashing bright droplets everywhere. He returned with a single, golden bubble held gently in his tiny beak. He pressed the bubble against the wolf's chest, and a warm, humming glow filled the air. Fenris felt a tingle in his throat, a sensation of joy he had forgotten long ago during his long silence.

Page 6:

Fenris opened his mouth, and instead of a simple bark, a magnificent, melodic howl echoed across the valley. The sound turned into a shower of falling stars that painted the sky in colors of violet and gold. The Shadow-Mist vanished, defeated by the pure music of a friend's kindness. From that night on, the giant wolf and the tiny glass sparrow became the guardians of the woods, proving that even the loudest voice sometimes needs the help of a small friend.

Moral:

True friendships can bridge any distance and help us find the magic we thought we had lost forever.

Regenerate Story

Generate Storybook

AI Storybook

Your imagination, our creation

The Moonlit Wolf and the Sapphire Sparrow



The Moonlit Wolf and the Sapphire Sparrow

The Journey begins...

Start Reading

The Silent Wolf



In the kingdom of Ever-Glow, where the trees hummed soft lullabies, lived Fenris, a giant wolf with fur of spun moonlight. Despite his size, Fenris was as gentle as a summer breeze, yet he carried a heavy secret in his heart. He had lost his howl to the Shadow-Mist, leaving him silent in a forest that thrived on music. He sat alone by the Crystal Brook, watching the glowing fish dance, wishing he could join the midnight chorus.

Read Page

Previous Spread

Pages 1 - 2

Language: English

Next Spread

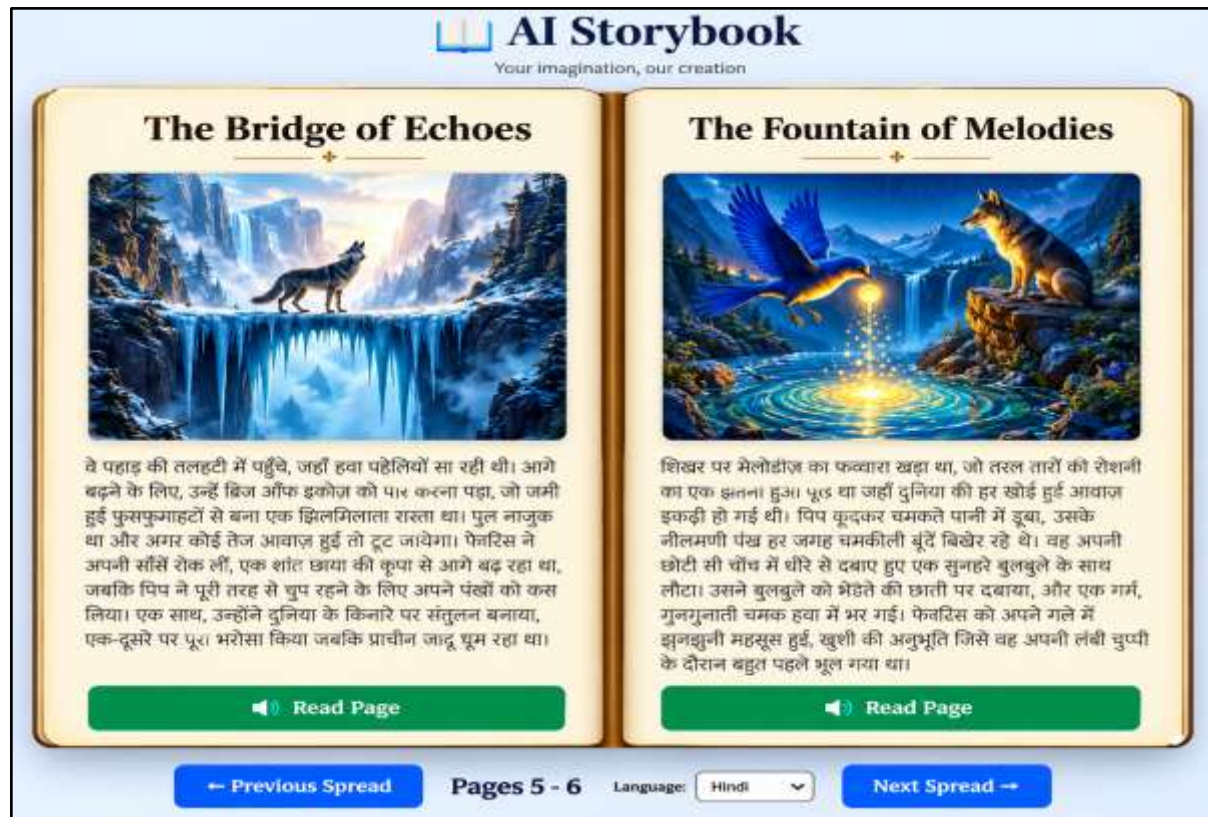


Figure 3. GUI Screen Shots of Story Maker

VII. Conclusion And Future Scope

This paper describes the Storybook Maker, a generative AI powered application for automatic creation of storybooks that uses processes such as story creation, process of narration, and text in multiple languages using AI technology and making it available in PDF format. Based on the use of Python, Django, and MySQL, the proposed method has a scalable architecture that allows end-to-end storytelling. It has been found out by comparison of several large language models and visualization models that Gemini 3 Flash Preview was the most effective approach for narrative creation and FLUX.1-dev showed the best result in illustrations generation. Everything proves the efficiency of the system for making storytelling accessible and successful; therefore, it can be used in education, children's learning, creative writing, and digital publishing. In the nearest future the focus will be on the usage of the best multimodal foundation models, age-adjustable and personalized storytelling, emotion-based narration, animated book creation, multilingual development, and Retrieval-Augmented Generation (RAG). The next step will include developing a mobile application of the framework and creating a collaboration among users in making content using cloud technology.

References

- [1] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 5998–6008. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [2] T. B. Brown, B. Mann, N. Ryder, et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020. [Online]. Available: <https://arxiv.org/abs/2005.14165>
- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), 2019, pp. 4171–4186. [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [4] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," arXiv preprint arXiv:2312.11805, 2023. [Online]. Available: <https://arxiv.org/abs/2312.11805>
- [5] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023. [Online]. Available: <https://arxiv.org/abs/2303.08774>

- [6] C. Raffel, N. Shazeer, A. Roberts, et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: <https://www.jmlr.org/papers/v21/20-074.html>
- [7] H. Touvron, L. Martin, K. Stone, et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2302.13971, 2023. [Online]. Available: <https://arxiv.org/abs/2302.13971>
- [8] R. Pérez y Pérez and M. Sharples, "Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA," *Knowledge-Based Systems*, vol. 17, no. 1, pp. 15–29, 2004. doi: 10.1016/S0950-7051(03)00048-0.
- [9] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in *Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002, pp. 311–318. [Online]. Available: <https://aclanthology.org/P02-1040/>
- [10] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Proc. ACL Workshop on Text Summarization Branches Out*, 2004, pp. 74–81. [Online]. Available: <https://aclanthology.org/W04-1013/>
- [11] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," in *International Conference on Learning Representations (ICLR)*, 2020. [Online]. Available: <https://arxiv.org/abs/1904.09675>
- [12] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10684–10695. [Online]. Available: <https://arxiv.org/abs/2112.10752>
- [13] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," arXiv preprint arXiv:2204.06125, 2022. [Online]. Available: <https://arxiv.org/abs/2204.06125>
- [14] Django Software Foundation, Django Documentation. [Online]. Available: <https://docs.djangoproject.com/>
- [15] Oracle Corporation, MySQL 8.0 Reference Manual. [Online]. Available: <https://dev.mysql.com/doc/>
- [16] G. Paul, Children Stories Dataset. Hugging Face Datasets. [Online]. Available: <https://huggingface.co/datasets/garethpaul/children-stories-dataset>
- [17] M. Bejan, 15000 Gutenberg Books [Data set]. Kaggle, 2021. [Online]. Available: <https://www.kaggle.com/datasets/mateibejan/15000-gutenberg-book>