StoryGPT : An LLM-based learning tool for specially-abled children

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

Sarang Pavanaskar D12A/51

Akshat Mahajan D12A/40

Tanmay Maity D12A/42

Mohit Vaidya D12A/60

Name of the Mentor

Dr. Sujata Khedkar

Vivekanand Education Society's Institute of Technology,

An Autonomous Institute affiliated to University of Mumbai

HAMC, Collector's Colony, Chembur,

Mumbai-400074

University of Mumbai (AY 2024-25)

CERTIFICATE

This is to certify that the Mini Project entitled **" StoryGPT : An LLM-based learning tool for specially-abled children "** is a bonafide work of **Sarang Pavanaskar (D12A-51), Askhat Mahajan (D12A-40), Tanmay Maity (D12A-42), Mohit Vaidya (D12A-60)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

(Dr. Sujata Khedkar)
Mentor

(Dr. Nupur Giri)
Head of Department

(Dr. J.M. Nair)
Principal

Mini Project Approval

This Mini Project entitled " **StoryGPT : An LLM-based learning tool for specially-abled children "** by **Sarang Pavanaskar (D12A-51), Askhat Mahajan (D12A-40), Tanmay Maity (D12A-42), Mohit Vaidya (D12A-60)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place:

Contents

Abstract

This project aims to revolutionize how educational narratives are crafted and delivered to teach social norms and appropriate behaviors to specially-abled children. By leveraging advancements in artificial intelligence, particularly Large Language Models (LLMs) and generative AI, this project seeks to automate the creation of personalized stories. These stories, designed to address specific situations and challenges, will serve as valuable tools for parents, educators, and caregivers. By offering personalized content in various formats such as text, audio, images, and videos, the project aims to provide an engaging and accessible learning experience for children, fostering their understanding of social norms in a creative and meaningful way.

Acknowledgement

We would like to sincerely thank everyone who helped us complete this project. First, we are very grateful to our mentor, Dr. Sujata Khedkar, for her constant support, guidance, and helpful advice throughout the project.

We also want to express our gratitude to our Head of Department, Dr. Nupur Giri, and Deputy Head of Department, Dr. Gresha Bhatia, for providing us with the opportunity to work on this project. We are thankful to our college, VESIT (Vivekanand Education Society's Institute of Technology), for providing us with a conducive environment for learning and innovation..

List of Figures

List of Tables

1. Introduction

This chapter introduces the StoryGPT project, focusing on the need for personalized tools to help specially-abled children learn social skills. It highlights the limitations of traditional methods and proposes an AI-driven system using Large Language Models (LLMs) and generative AI to create engaging, multimedia-rich stories. The chapter outlines the problem of manually creating social stories and presents the project's objectives, which include automating the story creation process and offering accessible, tailored content. It also provides an overview of the report's structure, detailing the key sections on research, system design, implementation, and future work.

1.1 Introduction

The ability to learn and understand social norms is a critical aspect of a child's development, yet for specially-abled children, acquiring these skills can be particularly challenging. Traditional educational methods often fall short in providing the tailored support and engagement these children need to thrive in social situations. As technology advances, there is a growing opportunity to transform the way we approach teaching social behaviors. This project aims to harness the power of Large Language Models (LLMs) and generative AI technologies to create personalized stories that are not only informative but also engaging for specially-abled children. These stories will be designed to address real-life scenarios and challenges, offering a more accessible and interactive learning experience in formats such as text, audio, images, and videos. By automating the story creation process and personalizing the content to each child's unique needs, this project seeks to provide a valuable resource for parents, educators, and caregivers, ultimately enhancing the educational support available for children with special needs.

1.2 Motivation

The motivation behind this project comes from the need to provide specially-abled children with more personalized, accessible, and effective tools for learning social norms and appropriate behaviors. Traditional teaching methods often lack the adaptability required to meet the unique learning needs of these children, leaving gaps in their ability to navigate social situations confidently. With the growing capabilities of artificial intelligence, particularly Large Language Models (LLMs) and generative AI, there is now an opportunity to create highly customized educational content that can engage children in ways that conventional approaches cannot. Personalized stories that incorporate relevant situations, challenges, and characters can help children better understand and practice social behaviors in a safe, supportive environment. By delivering these stories in various formats—text, audio, images, and videos—this project aims to make learning more immersive, enjoyable, and tailored to each child's abilities. This initiative not only seeks to support children's social development but also to empower parents, educators, and caregivers with a powerful, innovative tool to enhance learning outcomes.

1.3 Problem Statement

Manually creating Social Stories for children with special needs is a labor-intensive and challenging process that demands significant time and effort from parents and educators. Traditional Social Stories often fall short in providing the necessary customization to address specific situations, behaviors, and learning needs, which limits their effectiveness. Additionally, relying solely on text-based content fails to capture the attention of children with varying learning preferences, reducing the overall impact of these stories. The absence of integrated multimedia elements such as audio, images, and videos further diminishes the engagement and effectiveness in teaching social norms and behaviors. Moreover, parents and educators face difficulties in regularly creating and updating these stories, making it hard to consistently meet the evolving needs of the children. This problem underscores the need for a more efficient, customizable, and engaging approach to creating Social Stories that can adapt to the unique requirements of each child.

Objectives

- To develop an AI-driven system that automates the creation of personalized educational stories tailored to the unique needs of specially-abled children.
- To produce these stories in multiple formats—text, audio, images, and videos—to cater to different learning preferences and enhance accessibility.
- To address specific social challenges and situations faced by specially-abled children through the narrative content of the stories.
- To provide parents, educators, and caregivers with an effective tool to support the social development of specially-abled children.
- To innovate educational technology by integrating Large Language Models (LLMs) and generative AI for personalized learning solutions.

1.4 Organization of the Report

This report is structured to provide a comprehensive overview of the StoryGPT project:

Chapter 1: Provides a detailed introduction to the project, including the motivation, problem statement, and objectives, establishing the context for the rest of the report.

Chapter 2: Conducts a thorough literature survey, reviewing existing systems, identifying their limitations, and highlighting the research gaps that StoryGPT addresses.

Chapter 3: Describes the proposed system in detail, covering the architectural framework, algorithm design, process design, and methodologies applied, along with hardware and software specifications.

Chapter 4: Focuses on the implementation and testing of the StoryGPT system, presenting experimental results, validation, and verification processes.

Chapter 5: Concludes the report, summarizing the key findings and offering suggestions for future work to further enhance the system.

2. Literature Review

This chapter provides a comprehensive literature review, examining existing systems and methodologies related to automated storytelling, generative AI, and text-to-image synthesis. The chapter is divided into three sections: a survey of existing systems, identifying the methodologies, advantages, and limitations of recent projects; an analysis of the research gaps and limitations within current systems; and the unique contributions of this mini project, StoryGPT, which aims to address several identified gaps by offering a more personalized, multimedia-rich, and automated storytelling experience.

2.1 Survey of Existing System

Year	Title	Objective	Methodology	Advantages	Limitations
2024	Sarid: Arabic Storyteller Using a Fine-Tuned LLM and Text-to-Image Generation	To develop an Arabic story generation model using a fine-tuned LLM and text-to-image generator	Collected a dataset of 527 Arabic children's stories through web scraping, Preprocessed and fine-tuned the Davinci model using the dataset, Generated visual elements using Midjourney, Integrated user inputs into the story generation process through a website	Gives stories based on user inputs such as theme, tone, and character, Produces high-quality images corresponding to the stories, Ensures character consistency throughout the story.	Requires careful prompt engineering to maintain consistency in visuals., Challenges in character consistency without the use of specific seeds, Dependence on the quality of the fine-tuned model and dataset.
2024	A survey of Generative AI Applications	To survey and categorize over 350 generative AI applications, providing insights into state-of-the-art technologies across various domains.	Organized into 15 categories such as text, images, video, and more, detailing current technologies and tools used.	Extensive coverage; organized framework; aids in identifying current technologies and trends.	Rapid tech evolution might make some details outdated; broad scope may be overwhelming for beginners.
2022	Deep Learning Methodology Converts Text to Image	To explore the application of Generative Adversarial Networks (GANs) for text-to-image synthesis, focusing on the challenges of translating textual descriptions into visual content and the utilization of deep learning techniques to enhance image generation	Utilized GANs with TensorFlow, NLTK, and PySimpleGUI; employed GAN-CLS algorithm; trained on Oxford-102 flower dataset with generator and discriminator models	Effective text-to-image synthesis; improved user interaction with GUI; deep learning efficiency.	High computational cost; time-consuming training; challenges in text interpretation and coherence.

2024	LLM-DetectAIve: a Tool for Fine-Grained Machine-Generated Text Detection	To classify texts into four categories: human-written, machine-generated, machine-humanized, and human-polished.	- Creation of a new dataset combining human-written and various machine-generated texts. - Training multiple detectors using models like RoBERTa, DeBERTa, and DistilBERT. - Evaluation using domain-specific and universal detectors.	- Offers fine-grained detection, improving the accuracy and relevance of detecting AI intervention in text. - Incorporates a multi-way classification system, offering more nuanced detection compared to binary classification.	- Domain-specific detectors require users to select the domain before detection. - May face challenges in cross-domain scenarios due to varying text features.
2024	HEART-felt Narratives: Tracing Empathy and Narrative Style in Personal Stories with LLMs	To empirically examine the relationship between narrative style and empathy using LLMs and large-scale crowdsourcing studies.	The authors introduced a novel taxonomy (HEART) and used LLMs to extract narrative elements, followed by a large-scale crowdsourcing study with 2,624 participants.	The methodology allows for the analysis of complex narrative elements and their impact on empathy using advanced AI techniques.	The study relies heavily on LLMs, which may have limitations in fully capturing the nuances of human narrative empathy.
2024	How Well Can Vision Language Models See Image Details?	Investigate how well Vision-Language Models (VLMs) perceive image details beyond the semantic level using a pixel value prediction task (PVP).	Fine-tune VLMs on the PVP task by adapting the connection module, language model, and visual encoder, then evaluate performance on image-language tasks.	Significant improvement in detailed image perception tasks, with notable boosts in referring image segmentation and video game decision-making performance.	Existing VLMs struggle with precise pixel value prediction when only fine-tuning the connection module and language model.
2023	Muse: Text-To-Image Generation via Masked Generative Transformers	To develop Muse, a more efficient text-to-image Transformer model that achieves state-of-the-art (SOTA) image generation performance.	Muse is trained on a masked modeling task using discrete image tokens and text embeddings from a pre-trained large language model, enabling efficient parallel decoding and fine-grained language understanding.	Muse outperforms diffusion and autoregressive models in efficiency and image quality, supports advanced image editing tasks without fine-tuning, and achieves SOTA results on benchmarks like CC3M and COCO.	Performance and efficiency gains may vary across different datasets, and applicability beyond tested benchmarks needs exploration.

Table no. 01 : Literature Survey

2.2 Limitation in Existing system or Research gap

Aspect	Limitation/Research Gap in Existing System
Story Customization	Most existing systems lack the ability to personalize social stories based on age, comprehension level, or specific needs of the child, making stories less effective.
Multimedia Integration	Current tools typically focus on text-based stories and do not integrate multimedia elements like videos, images, and audio in a seamless, user-friendly way.
Cultural and Linguistic Adaptability	Current systems lack the ability to adapt social stories for diverse cultural backgrounds and multiple languages, limiting their global applicability.
Automation	The creation of social stories is largely a manual, time-consuming process involving specialists like psychologists, with minimal automation for content generation.
Real-time Feedback	Existing systems do not offer mechanisms for gathering real-time feedback from parents or educators to improve or adapt stories dynamically.
Dataset Availability	Publicly available datasets of social stories are limited, and there is no organized corpus to systematically train machine learning models for personalized story generation.
Story Categorization	Current solutions do not categorize stories by themes (e.g., hospital visits, school behavior) or emotional situations, limiting their applicability to diverse scenarios.
Cross-format Consistency	Systems lack consistency when generating stories across formats (text, video, audio), resulting in varying levels of quality and cohesion between story components.
User Interaction and Usability	Existing systems often have complex interfaces, making it difficult for non-technical users (parents, teachers) to create and adapt stories effectively.
Scalability	Tools for social stories generation are not scalable to handle diverse user bases, particularly when creating personalized, multimedia-rich stories for various use cases.

Table no. 02 : Limitations in the existing systems

2.3 Mini Project Contribution

This project contributes to the domain of automated storytelling by addressing the limitations found in existing systems:

Automated Story Generation: StoryGPT introduces an AI-based system that dynamically generates personalized stories based on user inputs, eliminating the need for manual story creation.

Image Integration: Unlike existing systems, StoryGPT seamlessly integrates AI-generated images into the story, ensuring that visuals are relevant to the narrative and enhance the overall storytelling experience.

Streamlined User Experience: The project utilizes Streamlit to provide an intuitive, user-friendly interface, allowing users to easily navigate between input, generation, and output pages without complex steps.

Dynamic Story Segmentation: StoryGPT automatically splits the generated stories into meaningful segments, inserting appropriate images between sections to create a more engaging and structured narrative flow.

Customization and Interactivity: The system allows for a high degree of customization, offering users the flexibility to tailor stories to their preferences and interact with the generated content seamlessly.

3. Proposed System

This chapter provides a detailed explanation of the system's architecture, algorithm, and design process. The proposed solution leverages advanced Large Language Models (LLMs) and generative AI technologies to generate personalized social stories, incorporating multimedia elements like text, audio, images, and video. The system enables caregivers to input specific behavioral challenges and preferences, and in response, creates tailored stories that are both engaging and accessible. This chapter also covers the technical aspects of the project, including hardware and software specifications, experiments conducted, and results achieved, with a focus on refining the system for improved efficiency and accuracy. Lastly, the chapter concludes with a discussion on future work, emphasizing scalability, multilingual support, and collaboration with educators to maximize the system's impact.

3.1 Proposed solution

The proposed solution involves developing a system that leverages Large Language Models (LLMs) and generative AI technologies to create personalized educational narratives for specially-abled children. This system will be designed to generate stories that address specific social challenges and situations, making it easier for children to understand and adopt appropriate social behaviors. The AI-driven platform will enable users—parents, educators, or caregivers—to input the child's unique needs, preferences, and behavioral challenges. The system will then generate a tailored story in various formats, including text, audio, images, and video, ensuring the content is accessible and engaging for the child.

By utilizing advanced AI technologies, the stories will be both highly personalized and contextually relevant, incorporating characters, scenarios, and language suited to each child’s learning level. These narratives will be designed to reinforce social norms through repetition and engaging storytelling, with interactive elements that encourage active participation. The platform will also allow users to create multiple stories based on different situations, helping children practice and internalize appropriate behaviors across a range of real-life scenarios.

Additionally, the system will include tools for continuous adaptation, where feedback from the child’s responses can be used to fine-tune the stories, making them more effective over time. This approach will not only enhance the learning experience for specially-abled children but also provide caregivers with a powerful tool for guiding social development. The solution aims to be scalable, with potential integration into educational apps or platforms for broader use, making personalized, AI-driven learning accessible to a larger audience.

3.2 Architectural Framework / Conceptual Design

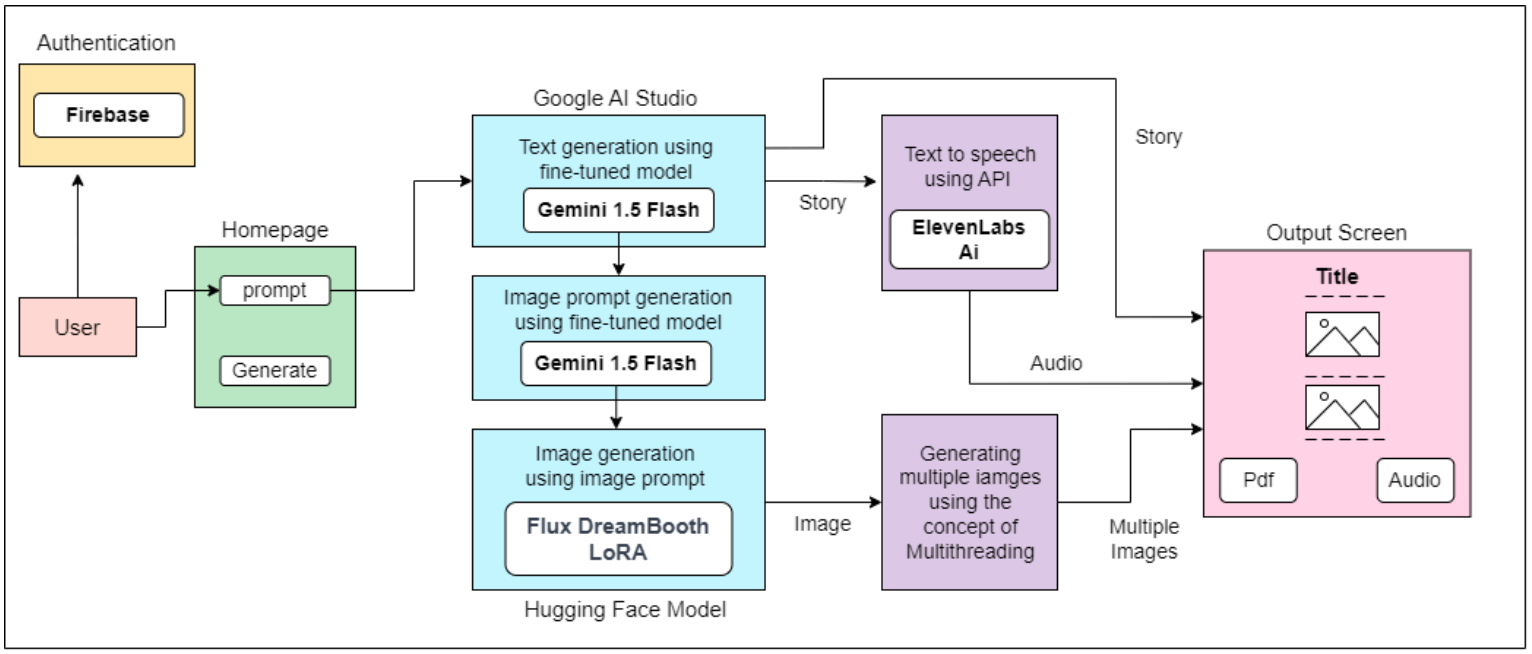


Fig. no. 01 : Block Diagram

3.3 Algorithm and Process Design

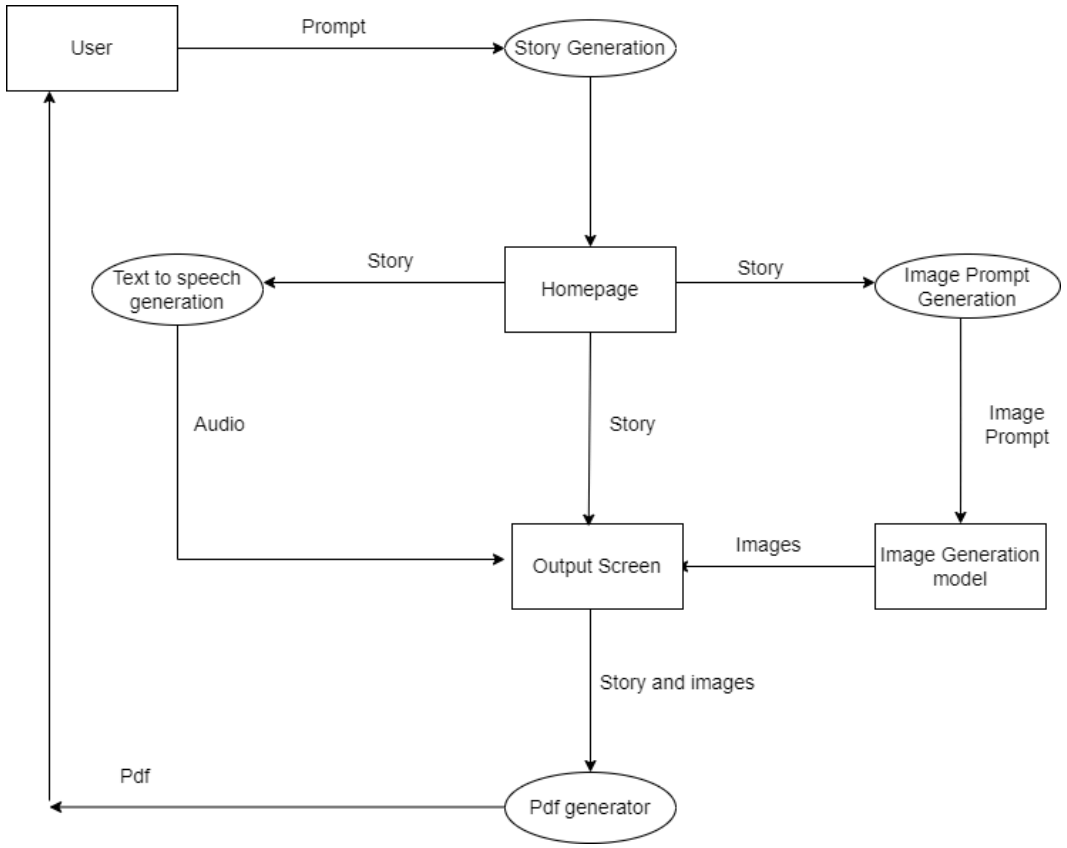


Fig. no. 02 : Data Flow Diagram

3.4 Methodology Applied

Problem Identification and Research : The first step involved conducting extensive research to identify the challenges faced by parents, educators, and specially-abled children in traditional learning environments. This research focused on understanding the limitations of current educational tools in teaching social norms and behaviors, particularly the lack of personalized content that can engage children with different learning abilities.

Data Collection and Preparation : A curated dataset of social stories was developed, focusing on real-world scenarios that specially-abled children often encounter. This dataset forms the foundation for both the story generation and image prompt creation processes, ensuring the relevance and effectiveness of the content generated by the system.

AI Model Development:

Gemini Model for Story Generation : The Gemini model was fine-tuned using the curated dataset to generate personalized social stories based on user input. This model enables the generation of narratives that address specific behavioral challenges and learning needs, ensuring each story is tailored to the child's context.

Gemini Model for Image Prompt Generation: A second instance of the Gemini model was fine-tuned to generate detailed image prompts from the curated dataset. These prompts are designed to be forwarded to the image generation model, ensuring that the images created are story-specific and visually aligned with the narrative.

1. **Image Generation:**
   Flux Dream Booth LoRA: The Flux Dream Booth LoRA model was employed for generating personalized, high-quality images based on the prompts provided by the Gemini model. These images enhance the visual storytelling experience for children.
2. **Multithreading for Multiple Image Generation:** To improve efficiency and handle the simultaneous creation of multiple images, multithreading techniques were implemented. This approach ensures faster image generation, allowing multiple story-specific images to be processed concurrently.
3. **Audio Generation:** Using the Eleven Labs API, the generated stories were converted into audio format, providing an additional modality for children who benefit from auditory learning. This step ensures that the stories are accessible to a wider range of learning preferences, offering a more engaging experience through voice narration.

4. **PDF Output Generation:** Python libraries were used to provide a downloadable PDF output of the generated story and images. This feature allows caregivers and educators to access the stories in a traditional, printable format, which can be shared or used offline as needed.
5. **Video Integration:** A system will be developed to combine the generated stories, images, and audio into a cohesive video format. This integrated approach creates immersive, multimedia stories that can effectively capture and maintain the child's attention, offering a richer learning experience.

3.5 Hardware & Software Specifications

High-end computer with the following configuration :

- Processor: Intel Core i5 or i7 (7th Generation or later)
- RAM: 8GB or more
- Storage: 256GB or higher
- Graphics card: Dedicated GPU with at least 4GB VRAM (for multimedia processing) or
- cloud GPU like Google Colab

Software Techstack :

- Gemini 1.5 Flash (fine-tuned) for text and image prompt generation
- Flux DreamBooth LoRA Hugging face model for image generation
- Frontend: Streamlit python library
- Authentication & Database: Firebase
- External libraries: Langchain, Streamlit etc.

3.6 Experiment and Results for Validation and Verification

Experiment	Result
Using llama 2 for story generation	● As llama 2 is a big model with a large amount of parameters, it takes long time for text generation ● Not producing expected output
Using Gemini fine tuning	● A lot of description was required along with story topic for getting desired output
Using fine tuned gemini model	● Achieved desired output after fine tuning the model with dataset containing around 100 stories
Using Stable Diffusion	● Took around 40-50 minutes for generating one single image
Using Midjourney	● Using their API key was paid and quite expensive as well as there were limits on number of images that could be generated on a single day
Using flux-lego-lora-dream booth	● Achieved desired output with accurate and good quality images ● Capable of generating multiple images at the same time ● Took around 1-2 minutes to generate all the images

Table no. 03 : Experiments & results

3.7 Result Analysis and Discussion

Story Generation : Using Gemini 1.5 Flash Model

This is the base model which generates the output in a long story format which is not desirable for the generation of social stories . Hence we fine tune it to get desired results.

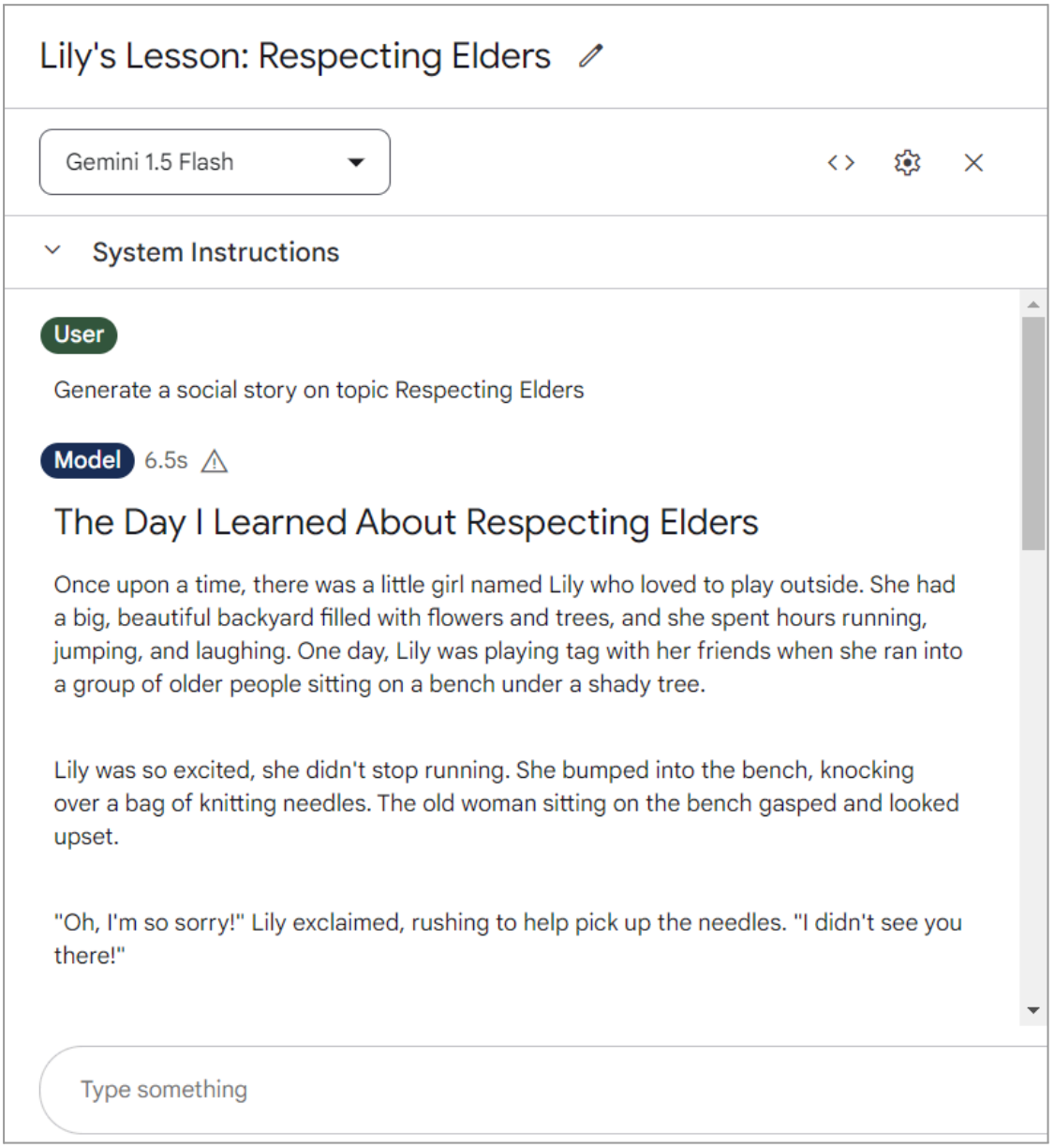


Fig no. 03 : Story Generation using Gemini model

This is the fine tuned version of the Gemini 1.5 Flash model and gives the desired output which can be used by other models for image and audio generation . This model is fine tuned using a dataset .

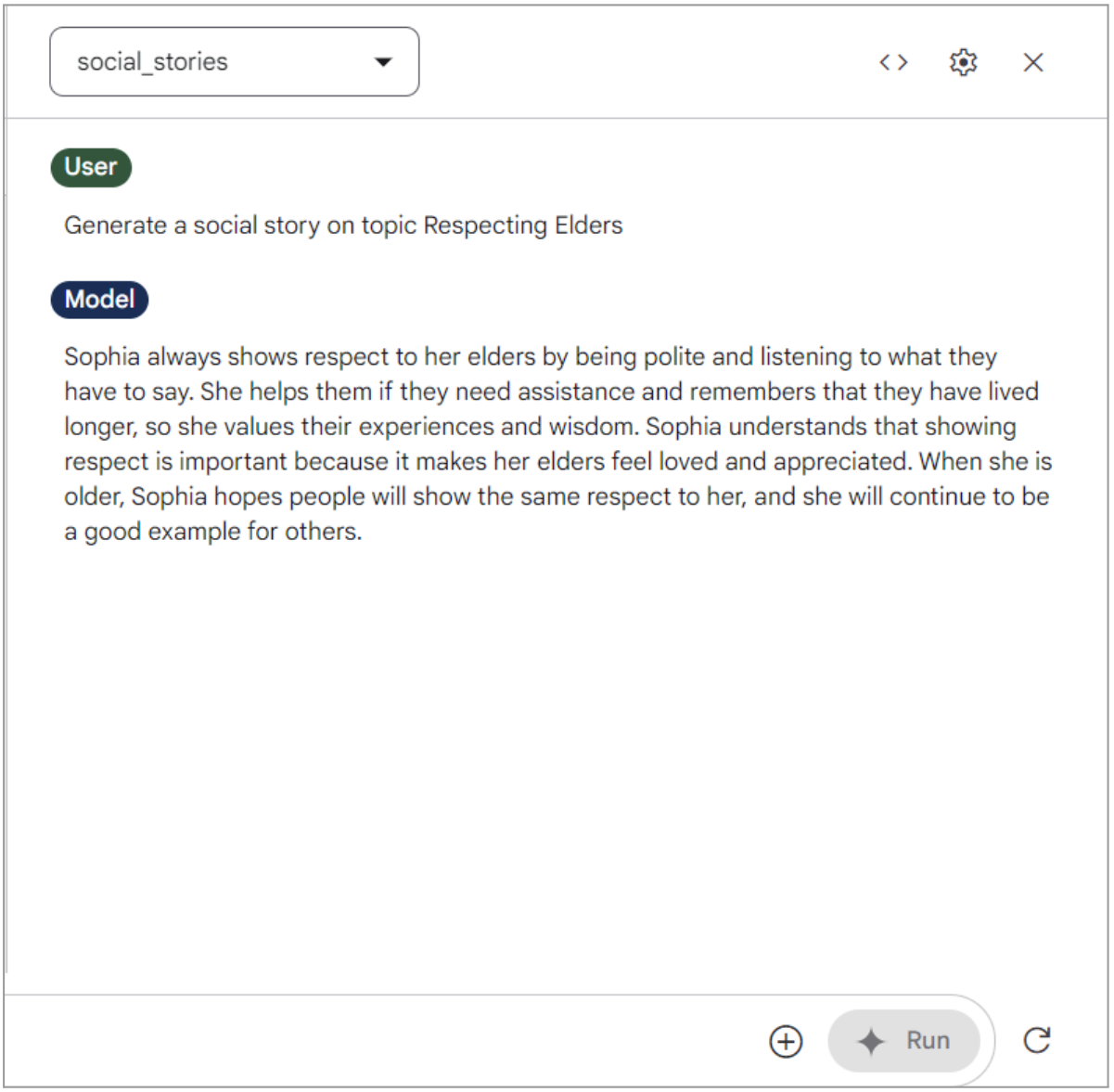
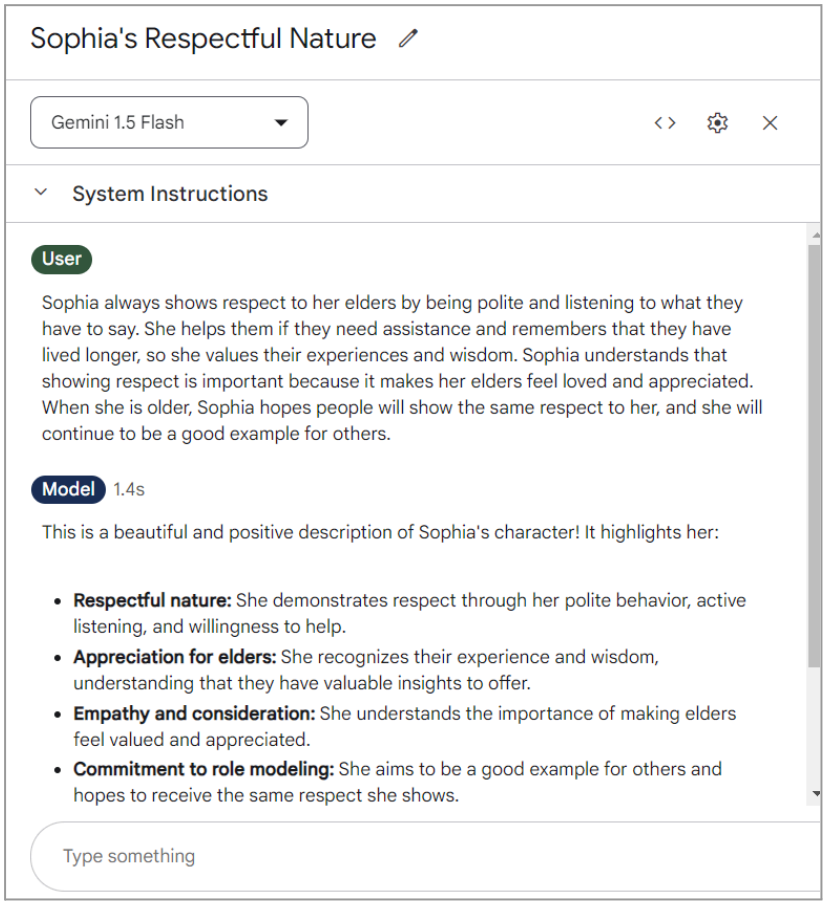


Fig no. 04 : Story Generation using Fine Tuned Model

Image prompt generation

This is the base model which generates the image prompts in a long descriptive format which is not desirable for the generation of relevant images as the image generating model expects short and descriptive image prompts . Hence we fine tune it to get better results.



This is the similar fine tuned version of the Gemini 1.5 Flash model and gives the desired image prompt which can be used by image model for image generation . This model is fine tuned using a dataset which consists of story and its corresponding image prompts.

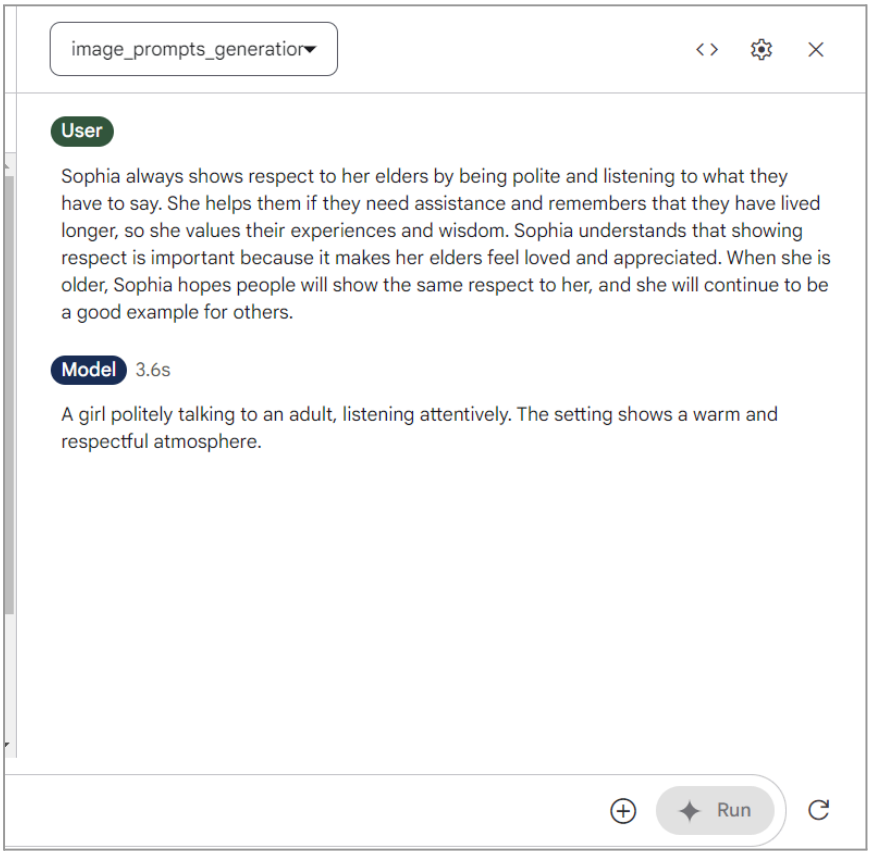


Fig no. 05 : Image prompt Generation

Implementation (Output)

This is the first screen that pops up when you click on the website link. We have also added an animation to this page, which enhances user engagement and captures attention right from the start.

Fig no. 06 : Starter Screen

This is the homepage where users can navigate to other pages and also can give input prompts and press the generate button to generate the story.

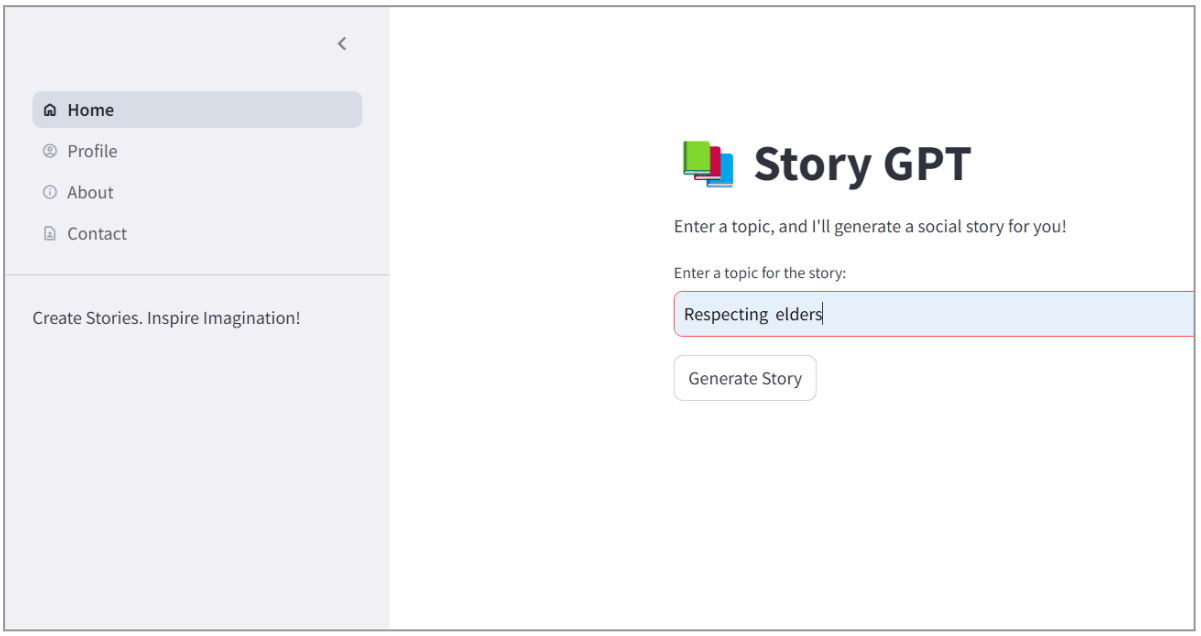


Fig no. 07 : Home Page

After clicking on generate story, We get directed to this page where our text generation model and image generation model combine to give a multiple image and text media output

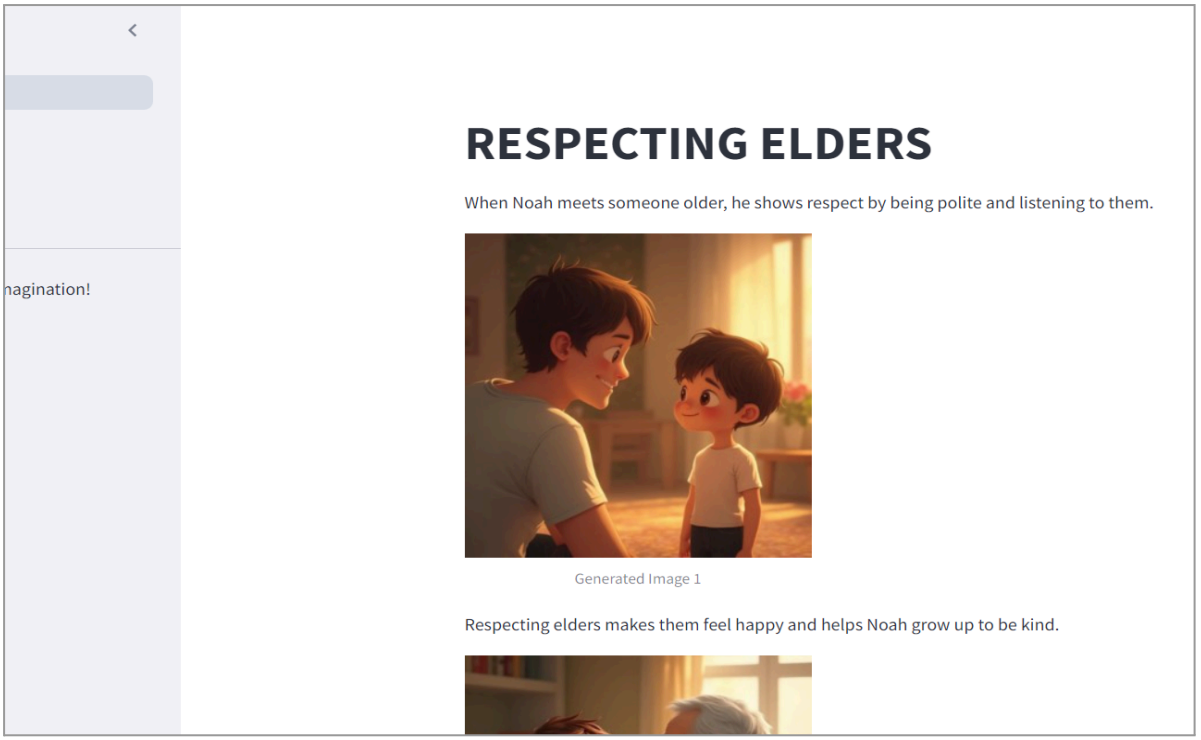


Fig no. 08 : Output Screen -1

This is the section within the output screen which contains the Audio generation button , Pdf download button and Video generate button which allows the user to have outputs generated in multiple media forms.

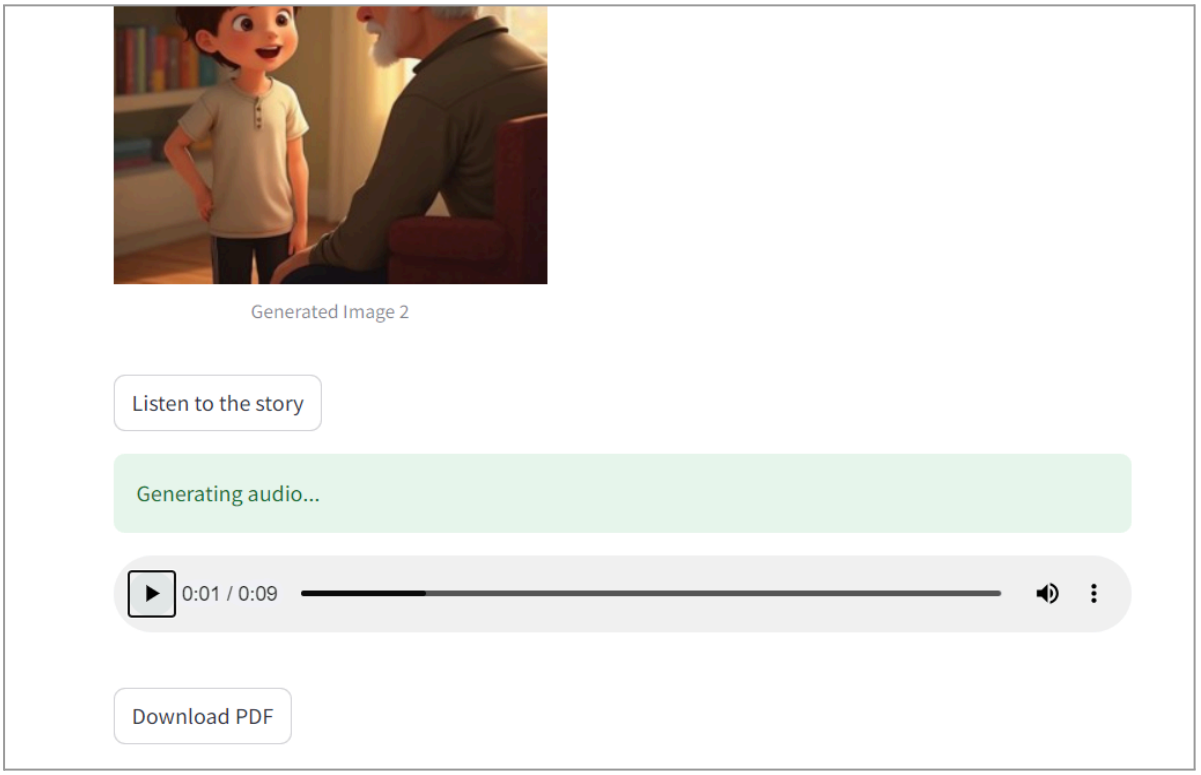


Fig no. 09 : Output Screen -2

This is the pdf output which is downloaded into the downloads folder of your pc.

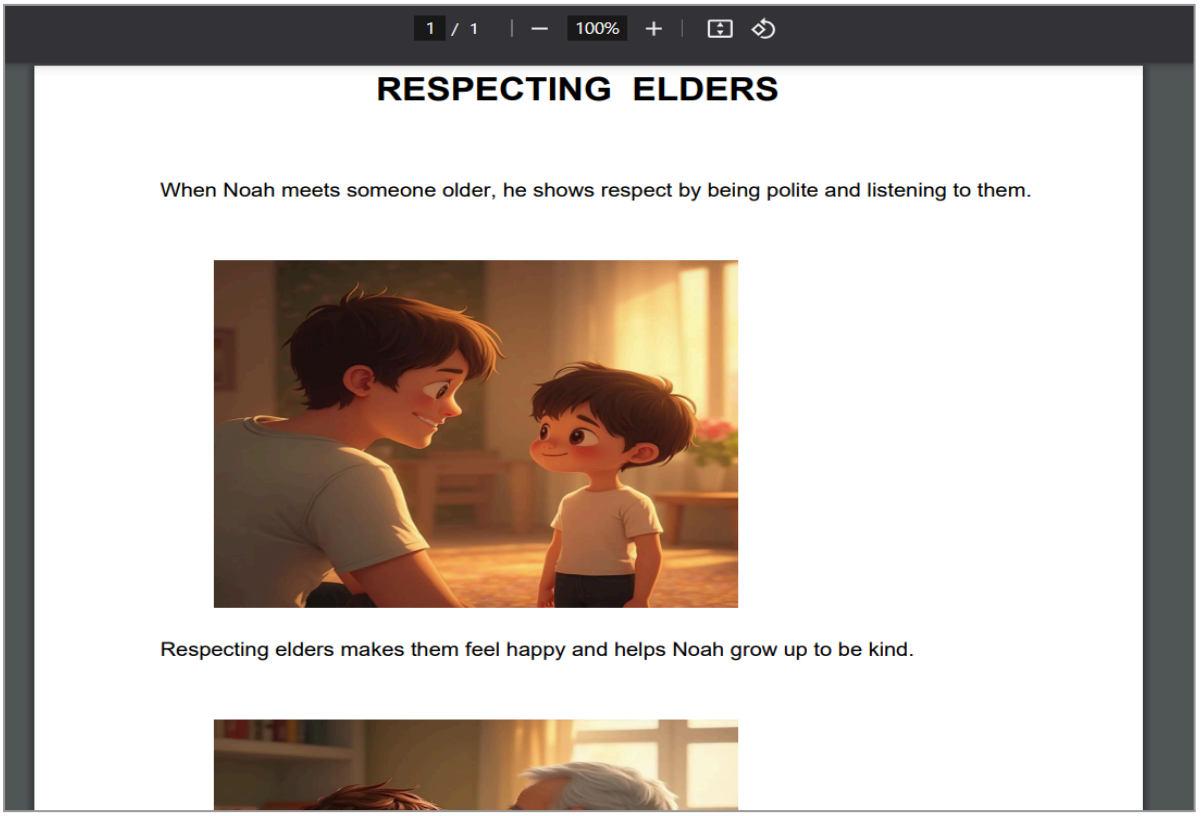


Fig no. 10 : Pdf format output

Implementation (Codes And Algorithm) :

Function to generate story

```
FUNCTION generate_story(topic)
    // Step 1: Initialize the generative model
    SET story_model = CreateGenerativeModel("tunedModels/socialstories-my8gxcmyb2vx")

    // Step 2: Start a new chat session
    SET story_session = story_model.StartChat(history=[])

    // Step 3: Generate a story based on the provided topic
    SET story = story_session.SendMessage("Generate a social story on topic " + topic).text

    // Step 4: Extract story components
    SET story_list = get_story_list(story)

    // Step 5: Return the results
    RETURN {
        'story': story,
        'story_list': story_list
    }
END FUNCTION
```

Fig no. 11 : generate_story function

Function to generate image prompt

```
FUNCTION generate_image_prompt(sentence)
    // Step 1: Initialize the generative model for image prompts
    SET image_prompt_model = CreateGenerativeModel("tunedModels/imagepromptsgeneration-n56rol

    // Step 2: Start a new chat session for generating image prompts
    SET image_prompt_session = image_prompt_model.StartChat(history=[])

    // Step 3: Generate an image prompt based on the provided sentence
    SET image_prompt = image_prompt_session.SendMessage(sentence).text

    // Step 4: Return the generated image prompt
    RETURN image_prompt
END FUNCTION
```

Fig no. 12 : genearte_image_prompt function

Function to generate story and image in proper format

```
FUNCTION generate_content(topic)
    // Step 1: Generate the story data using the given topic
    SET story_data = generate_story(topic)

    // Step 2: Extract the list of sentences from the generated story
    SET story_sentences = story_data['story_list']

    // Step 3: Initialize an empty list for image prompts
    SET image_prompts = []

    // Step 4: Generate image prompts for each sentence in the story
    FOR i FROM 0 TO LENGTH(story_sentences) - 1 DO
        SET prompt = generate_image_prompt(story_sentences[i])
        APPEND prompt TO image_prompts
    END FOR

    // Step 5: Print each generated image prompt
    FOR EACH prompt IN image_prompts DO
        PRINT prompt
        PRINT '\n'
    END FOR

    // Step 6: Return the results including story, story list, and image prompts
    RETURN {
        'story': story_data['story'],
        'story_list': story_data['story_list'],
        'image_prompts': image_prompts
    }
END FUNCTION
```

Fig no. 13 : generate_content function

```
FUNCTION get_story_list(story)
    // Step 1: Split the story into sentences based on the period character
    SET sentences = story.split('.')

    // Step 2: Remove any leading/trailing whitespace and filter out empty sentences
    SET sentences = [s.strip() FOR EACH s IN sentences IF s.strip() IS NOT EMPTY]

    // Step 3: Add a period to the end of each sentence
    SET sentences = [s + '.' FOR EACH s IN sentences]

    // Step 4: Print the list of sentences for debugging purposes
    PRINT sentences

    // Step 5: Return the list of sentences
    RETURN sentences
END FUNCTION
```

Fig no. 14 : split story into parts

Function for image generation

```
FUNCTION generate_image(prompt)

    // Step 1: Define the query function to send requests
    FUNCTION query(payload)
        TRY
            PRINT "Sending payload: " + payload  // Log the payload being sent
            SET response = POST(API_URL, headers=headers, json=payload)  // Send POST reques
            PRINT "Response status code: " + response.status_code  // Log the response statu
            response.raise_for_status()  // Raise an error for bad responses (4xx and 5xx)
            RETURN response.content  // Return the response content
        EXCEPT requests.exceptions.HTTPError AS http_err
            PRINT "HTTP error occurred: " + http_err  // Log HTTP errors
            PRINT "Response text: " + response.text  // Print response text for more details
            RETURN None  // Return None on error
        EXCEPT Exception AS err
            PRINT "Other error occurred: " + err  // Log other errors
            RETURN None  // Return None on error
        END TRY
    END FUNCTION

    // Step 2: Call the query function with the prompt
    SET image_bytes = query({"inputs": prompt})

    // Step 3: Check if image_bytes is valid and return the image
    IF image_bytes IS NOT None THEN
        RETURN Image.open(io.BytesIO(image_bytes))  // Return PIL Image object
    ELSE
        PRINT "Failed to generate image for prompt: " + prompt  // Log failure message
        RETURN None  // Return None on failure
    END IF
END FUNCTION
```

Fig no. 15 : generate images using image prompts

Function for audio generation

```
    SET tts_url = "https://api.elevenlabs.io/v1/text-to-speech/" + VOICE_ID + "/stream"

    // Step 2: Prepare the headers for the API request
    SET headers = {
        "Accept": "application/json",
        "xi-api-key": xi_api_key  // Include the API key for authorization
    }

    // Step 3: Define the payload data for the TTS request
    SET data = {
        "text": text,  // Text to be converted to speech
        "model_id": "eleven_multilingual_v2",  // Specify the model to use
        "voice_settings": {
            "stability": 0.8,  // High stability for clear speech
            "similarity_boost": 0.9,  // High resemblance to the base voice
            "style": 0.2,  // Low expressiveness for neutral tone
            "use_speaker_boost": False  // No extra intensity
```

Fig no. 16 : audio generation

```
    // Step 4: Make the POST request to generate audio
    SET response = POST(tts_url, headers=headers, json=data, stream=True)

    // Step 5: Check if the response is OK
    IF response.ok THEN
        // Step 5a: Create an in-memory binary stream to hold the audio data
        SET audio_stream = CreateBytesIO()  // Initialize an empty byte stream

        // Step 5b: Write the audio data in chunks to the stream
        FOR EACH chunk IN response.iter_content(chunk_size=1024) DO
            audio_stream.write(chunk)
        END FOR

        // Step 5c: Move the cursor to the beginning of the stream
        audio_stream.seek(0)
        RETURN audio_stream  // Return the audio stream
    ELSE
        PRINT "Error: " + response.text  // Log error details
        RETURN None  // Return None on failure
    END IF
END FUNCTION
```

Fig no. 17 : audio generation - 2

Folder Structure of the project

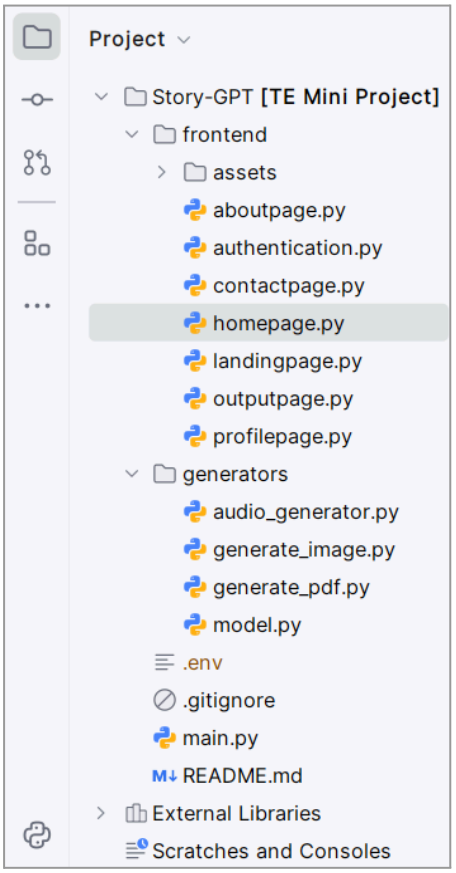


Fig no. 18 : folder structure

- All the frontend files are kept inside the frontend folder .
- All the files containing the models and apis to generate text, images, audio and pdfs are kept inside the generators folder .
- All the credentials of the various apis , api keys and secret ids are kept in the .env file .
- And the main.py file contains the starter code responsible for user authentication .

3.8 Conclusion and Future work

Our project aims to create a transformative tool that enhances social skill development for children with special needs through personalized, AI-driven educational content combined with immersive experiences. We will utilize advanced AI tools like Gemini and diffusers to create tailored social stories and images that meet each child's specific learning and sensory needs. This innovative approach addresses the challenges these children face in traditional learning environments and offers a flexible solution suitable for various educational and therapy settings. As we continue to improve and develop our prototype, our primary goal is to help children with special needs navigate social situations more effectively, ultimately enhancing their quality of life. This project represents a significant step toward making learning easier, more engaging, and more effective for those who need it most. Through ongoing development, testing, and collaboration, we believe our solution will have a positive impact on the lives of children and their families.

Future Scope

1. Multilingual Audio Generation: Expand audio capabilities to support multiple languages for broader accessibility.
2. NLP Integration: Implement natural language processing to highlight specific words in the text, enhancing engagement and user-friendliness.
3. Video Creation: Develop a process to generate videos from text and images, incorporating multimedia elements to enrich storytelling.
4. Testing and Optimization: Conduct thorough testing of the entire workflow to improve quality and user experience based on feedback.
5. Collaboration with Educators: Partner with educators and therapists to ensure the tool meets the needs of students and can be effectively integrated into educational settings.

4. References

[1] M. Alabdulrahman, R. Khayyat, K. Almowallad, and Z. Alharz, "Sarid: Arabic storyteller using a fine-tuned LLM and text-to-image generation," in *Proc. of the 2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, 2024.

[2] C. Gou, A. Felemban, F. F. Khan, D. Zhu, J. Cai, H. Rezatofighi, and M. Elhoseiny, "How Well Can Vision Language Models See Image Details?," *arXiv preprint* arXiv:2408.03940, 2024.

[3] M. Konenkov, A. Lykov, D. Trinitatova, and D. Tsetserukou, "VR-GPT: Visual Language Model for Intelligent Virtual Reality Applications," *arXiv preprint* arXiv:2405.11537, 2024.

[4] H. Chang, H. Zhang, J. Barber, A. J. Maschinot, J. Lezama, J. Lu, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, "Muse: Text-To-Image Generation via Masked Generative Transformers," *arXiv preprint* arXiv:2301.00704, 2023.

[5] J. Izquierdo-Domenech, J. Linares-Pellicer, and I. Ferri-Molla, "Virtual Reality and Language Models, a New Frontier in Learning," [Online]. Available: https://drive.google.com/file/d/1DXdgXvhx9zYKlvvZM4kYvRxIPykeRgJg/view

[6] J. Shen, J. Mire, H. W. Park, C. Breazeal, and M. Sap, "HEART-felt Narratives: Tracing Empathy and Narrative Style in Personal Stories with LLMs," *arXiv preprint* arXiv:2405.17633, 2024.

[7] "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv preprint* arXiv:2106.09685, 2021.

[8] "MuRIL: Multilingual Representations for Indian Languages," *arXiv preprint* arXiv:2103.10730, 2021.

[9] H. Chang, H. Zhang, J. Barber, A. J. Maschinot, J. Lezama, J. Lu, M.-H. Yang, K. Murphy, W. T. Freeman, M. Rubinstein, Y. Li, and D. Krishnan, "Muse: Text-To-Image Generation via Masked Generative Transformers," *arXiv preprint* arXiv:2301.00704, 2023.

[10] "Cross-Lingual Conversational Speech Summarization with Large Language Models," *arXiv preprint* arXiv:2408.06484, 2024.

[11] "CT-Eval: Benchmarking Chinese Text-to-Table Performance in Large Language Models," *arXiv preprint* arXiv:2405.12174, 2024.

[12] J. Shen, J. Mire, H. W. Park, C. Breazeal, and M. Sap, "HEART-felt Narratives: Tracing Empathy and Narrative Style in Personal Stories with LLMs," *arXiv preprint* arXiv:2405.17633, 2024.

[13] "LLM-DetectAIve: a Tool for Fine-Grained Machine-Generated Text Detection," *arXiv preprint* arXiv:2408.04284v1, 2024.

[14] "LLaVA-Surg: Towards Multimodal Surgical Assistant via Structured Surgical Video Learning," *arXiv preprint* arXiv:2408.07981, 2024.

[15] M. Konenkov, A. Lykov, D. Trinitatova, and D. Tsetserukou, "VR-GPT: Visual Language Model for Intelligent Virtual Reality Applications," *arXiv preprint* arXiv:2405.11537, 2024.