

ARI3333 - Project Report 2024/25

Storyboard-AI (*SB-AI*)



Name: **Andrea F. Lucas & Sean D. Muscat**
Course: ARI3333 - Artificial Intelligence (AI)
ID No: 0279704L & 0172004L

Project Report's Distribution of Work

Section	Responsible Person(s)
Project Overview & README.md	Andrea F. Lucas
Design Choice: Text Generation Module	Andrea F. Lucas
Design Choice: Image Generation Module	Sean D. Muscat
Implementation: Homepage	Andrea F. Lucas
Implementation: Login, Sign-in & Sign-out	Andrea F. Lucas
Implementation: My Profile	Andrea F. Lucas
Implementation: Story Creation	Both
Implementation: Archive	Andrea F. Lucas
Implementation: 404 Page	Andrea F. Lucas
Evaluation of Generated Story (<i>Examples</i>)	Both
Ethical Considerations	Sean D. Muscat
Challenges & Current Limitations	Andrea F. Lucas
Conclusion & Future Work	Andrea F. Lucas

Storyboard-AI GitHub Repository

The Storyboard-AI project's complete source code is available on GitHub via the following link: GitHub Repository (<https://github.com/AFLucas-UOM/Storyboard-AI>). The repository contains all the files and resources required to run the system, including the front-end interface, back-end Flask application, and the JSON datasets for storing user-generated stories. Additionally, the repository provides detailed documentation to assist with setup, usage, and customisation of the platform.

1 Project Overview

This document outlines the development, design, and implementation of Storyboard-AI (SB-AI), a local AI-driven storytelling platform that seamlessly integrates text and image generation to create immersive and interactive user experiences. The system prioritises user privacy and ethical AI practices by executing all processes locally on the user's device, safeguarding sensitive data and eliminating reliance on external servers. With its intuitive interface, SB-AI enables users to generate, refine, and adapt stories, accompanied by dynamically generated images, and offers features such as text-to-speech, PDF export, and a personalised archive for managing saved stories.

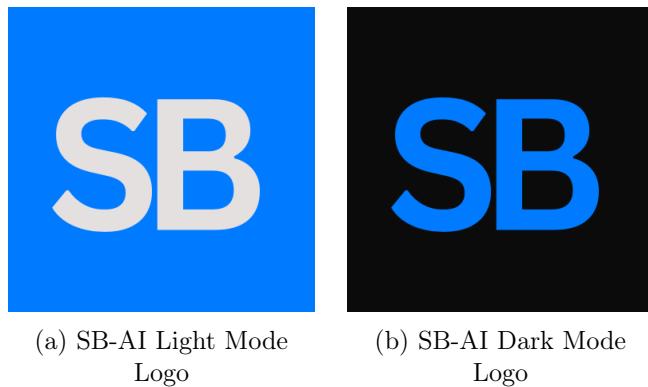


Figure 1: Website Logos for Light and Dark Modes

The platform's design leverages advanced AI models, including Orca-mini for text generation and Stable Diffusion for image creation, optimized for performance on consumer-grade hardware. By combining these technologies, the system delivers a cohesive storytelling experience while addressing challenges such as prompt engineering, computational efficiency, and hardware limitations. User feedback has played a pivotal role in shaping features like story re-generation, adaptive interactions, and multimodal capabilities, ensuring inclusivity and accessibility for a diverse audience.

2 Design choices for Each Generative Model

This section discusses the design and implementation of the Text and Image Generation Modules, which serve as the foundation of SB-AI. These modules have been developed with a focus on providing personalised, efficient, and engaging storytelling experiences while adhering to privacy-first principles. Advanced AI models, such as Orca-mini for text generation and Stable Diffusion for image creation, have been utilised to ensure seamless integration and high-quality outputs tailored to user inputs. The following subsections outline the technical decisions, challenges, and optimisations involved in their development.

2.1 Text Generation Module

The Text Generation Module in SB-AI was developed to deliver personalised storytelling through advanced language model capabilities, with the aim of running all models locally on the user's device for enhanced privacy and efficiency, as outlined in Section 5. This approach ensures that user data remains secure while reducing the reliance on external servers, thereby improving response times and overall system performance. To achieve this, the project utilised Ollama, a local model server that allows for efficient hosting and querying of language models on personal devices. Ollama provides an accessible interface for managing and running large language models locally, making it an ideal choice for integrating privacy-focused AI functionalities into the platform.

Initially, TinyLlama:1.1b-chat, a lightweight model based on Meta's Llama-2 architecture, was utilized for its speed and usability. However, its limitations in contextual understanding and prompt adherence led to a transition to Orca-mini:latest, a more advanced model trained on Orca-style datasets as detailed in [1]. This model, available in configurations with parameter sizes ranging from 3 to 70 billion, offered improved narrative coherence, contextual relevance, and adherence to user prompts, making it a better fit for the project's needs.

The switch from TinyLlama to Orca-mini marked a pivotal improvement, significantly enhancing the quality of generated stories. Orca-mini excelled at maintaining consistency in plot lines and crafting rich, engaging narratives tailored to user inputs. Additionally, its ability to generate structured content such as titles, summaries, and rephrased paragraphs streamlined various features of the platform. The language model also played a crucial role in generating effective prompts for other modules, including the Image Generation Module, ensuring that the entire system functioned cohesively.

Despite its advantages, integrating Orca-mini into the platform posed several challenges. Hardware limitations were a recurring obstacle, as running larger language models locally on personal computers led to increased latency in generating stories and responses. This occasionally disrupted the fluidity of the user experience, necessitating optimisations to balance performance and functionality. Prompt engineering presented another significant challenge, requiring iterative experimentation to craft prompts that yielded precise and desirable outputs. Each new feature demanded unique and well-defined prompts to maintain the overall quality and relevance of the generated content.

The reliance on Orca-mini for prompt generation and optimization became a cornerstone of the development process, enabling the system to handle tasks efficiently while improving user experience. By addressing the initial hurdles and leveraging Orca-mini's strengths, the Text Generation Module successfully delivers compelling and personalised storytelling, laying the groundwork for seamless integration with other AI-driven functionalities in the platform.

2.2 Image Generation Module

The image generation section was implemented using Stable Diffusion, which was selected as an ideal option for the storyboard project due to its versatility, efficiency, and accessibility. As an open-source model, Stable Diffusion not only allows free usage but also benefits from continuous community-driven improvements. This open nature provides a significant advantage, enabling the model to be adapted and customised to align seamlessly with the creative and technical requirements of the project via the prompts sent for image generation, primarily focused on supplementing the user's story with an image. Stable Diffusion incorporates built-in safeguards to prevent the generation of branded or copyrighted material, ensuring compliance with intellectual property laws and reducing the risk of inadvertent copyright infringement.

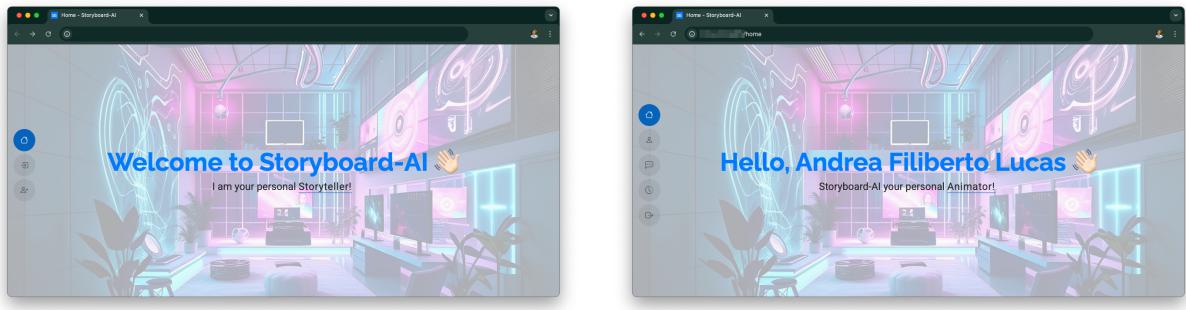
The model's training has been designed with restrictions to avoid outputs resembling specific brands, logos, or proprietary designs, thereby upholding the integrity of its creative outputs and maintaining ethical usage standards. It also cannot generate explicit content, making it ideal for use in this project. Unlike many generative models that operate directly in pixel space, Stable Diffusion utilises a compressed latent space, which, while significantly reducing the computational power required, introduces certain limitations. The visual outputs, being more akin to modern art, prioritise intricate and artistic interpretations over photorealistic accuracy. This means that while the images can enhance the storytelling experience with a unique visual style, they may not be suitable for projects requiring highly detailed, lifelike representations. This trade-off between computational efficiency and photorealism may limit its applicability in scenarios where realism is a critical requirement.

Moreover, Stable Diffusion is especially well-suited for systems with limited hardware, as it operates efficiently on consumer-grade GPUs, purely on CPUs, or using MPS (Metal Performance Shaders) acceleration for Apple M-series devices. To further optimise performance, caching mechanisms are employed so that the required files are downloaded and processed fully only during the initial run, enabling subsequent runs to execute more quickly. This ensures an efficient and user-friendly experience while maintaining the ability to produce detailed and expressive visuals, even on lower-end systems.

3 Implementation of Overall System Architecture

3.1 Homepage - (*index.html*) & (*index2.html*)

The splash screen, displayed on the first page (*index.html*), provides an engaging introduction to SB-AI for users who are not logged in. It features a dynamic hero section with a striking background image and an animated tagline that showcases the platform's key functionalities, such as storytelling, animating, and narrating. The design is fully responsive and enhanced with smooth animations, delivering a visually appealing and immersive experience across all devices. For logged-in users, the page automatically adapts to provide a more personalised experience, ensuring intuitive navigation and interaction.



(a) Homepage - User not logged in

(b) Homepage - User logged in

Figure 2: Comparison of Homepage States

After logging in, users are greeted with a customized version of the splash screen (*index2.html*), which personally welcomes them by their first name, creating a stronger connection with SB-AI. This version offers navigation options to access their profile, chat, archives, and the ability to sign out, enabling easy access to essential features. While maintaining the same visually appealing design, the page adjusts its content to reflect the user's logged-in status, ensuring a seamless and tailored user experience.

The footer, consistently present throughout the website, credits the project developers and acknowledges the use of the 'BootstrapMade' template for the website's layout. This provides proper attribution while maintaining a cohesive and professional aesthetic across the platform.

3.2 Login, Sign-up & Sign-out

The login and sign-up functionalities form an integral part of the authentication flow, linking directly to the system's splash screen and subsequent user experience.

3.2.1 Login Functionality

The `/login` route is designed to handle both GET and POST requests. During a POST request, it processes user credentials submitted via the login form (`login.html`) or an API call. User inputs are sanitized to prevent security vulnerabilities, such as injection attacks, and credentials are verified against stored user data. If the user enables the "Remember Me" option, cookies are set to allow a persistent session for up to 30 days. Upon successful authentication, the user is redirected to the personalised splash screen (`index2.html`). Additionally, if cookies from a previous session are valid, the route bypasses the login process and directly redirects the user to their personalised page.

3.2.2 Sign-up Functionality

The `/signup` route provides new users with the ability to register via a form (`signup.html`) or an API call. This route validates user inputs, including ensuring that passwords match, enforcing password complexity rules (minimum 8 characters with at least one number), and verifying email uniqueness using the `/check-email` endpoint. Upon successful validation, the system hashes the password for secure storage, saves the user's details, and redirects them to the personalised splash screen.

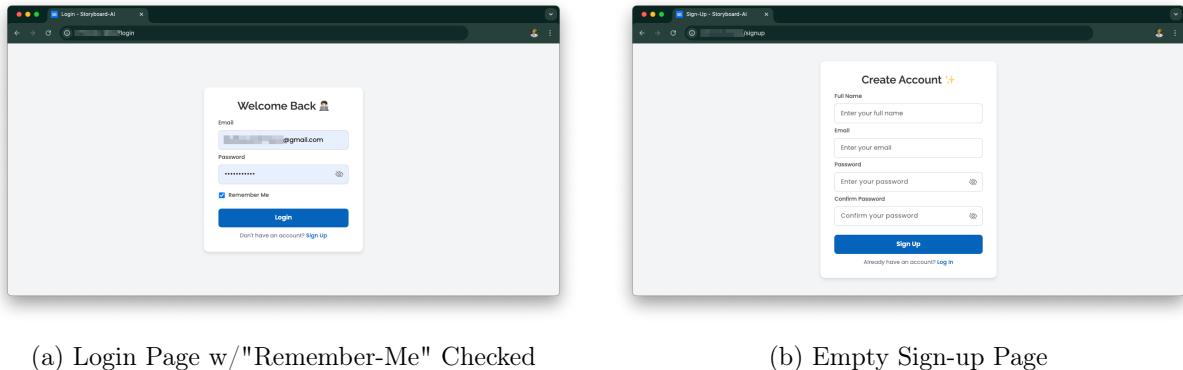


Figure 3: Login & Sign-up Pages

3.2.3 Sign-out Functionality

The `/signout` route facilitates the logout process, clearing session data and cookies. After signing out, the user is redirected to the default splash screen (`index.html`). Supporting JavaScript functions dynamically clear localStorage in the user's browser and update UI elements to ensure consistency across the entire system.

3.3 My Profile - (*profile.html*)

The profile page, (`profile.html`), serves as a interface that allows users to view and update their account information, including their name, email, password, and profile picture. The `/profile` route is responsible for rendering this page by retrieving the user's data from the `credentials.json` file. User authentication is handled by verifying the email stored in the session. Upon successful authentication, the system dynamically displays the user's details on the `profile.html` page, excluding sensitive information such as the password from direct display. The profile picture is retrieved from the `static/img/PFPs` directory, defaulting to `default.png` if no custom picture is available.

Users can update their profile details through a form on the profile page. The form supports changes to the user's name, email, password, and profile picture. Any input provided is sanitized using the `bleach` library to prevent XSS attacks, ensuring that user data remains secure. If a new profile picture is uploaded, it is validated against allowed file types (`png`, `jpg`, `jpeg`) and size limits (10 MB) before being saved securely using `secure_filename`. Users also have the option to remove their profile picture, which resets it to the default image.

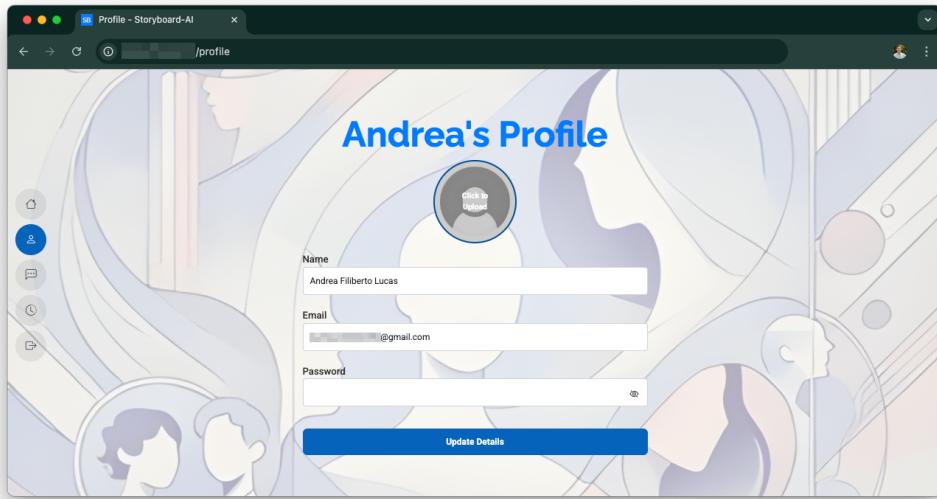


Figure 4: Profile Page w/Profile Upload Text Shown (*on hover*)

Password updates are handled with additional security checks. The updated password is hashed using the `generate_password_hash` function from `werkzeug.security` before being stored in `credentials.json`. To ensure usability, the form provides real-time feedback on password complexity, requiring at least eight characters and one numeric digit. The system prevents unnecessary password re-hashing if the user re-enters their existing password.

The `/update_profile` route processes the form data and updates the user's details in `credentials.json`. It also handles the removal of existing profile pictures from the server if requested by the user. Once the updates are saved, the user is redirected back to the profile page to view the changes. By integrating dynamic rendering and secure data handling, the profile page provides a robust and user-friendly experience for managing personal information.

3.4 Story Creation - (*chat.html*)

This section will cover the main functionality of `chat.html`, which serves as the interface for story generation, combining both text and image generation to create an engrossing and captivating story for users. All calls for text and image generation are made to the Python Flask back end, which are used to generate content, as well as write and read from json files.

3.4.1 Main Functionality

The chat system's core functionality is centred on creating an engaging and interactive storytelling experience. It starts by welcoming the user by name and explaining the process, initiating a series of dynamic queries to gather key details for the story. These questions cover elements like the main character's name, the story's theme, and the preferred ending, with optional prompts for unique traits or plot twists. To maintain a natural and conversational tone, each query is designed with five randomized variations, ensuring that no two interactions feel repetitive. This variation not only enhances the user experience but also plays a crucial role in sustaining user engagement and retention.

After gathering all user inputs, the system compiles a detailed prompt for the story creation process, ensuring the narrative aligns with the user's vision. At this stage, user input is disabled, preventing further messages from being sent to the system. This approach minimizes errors and serves as a clear indicator to the user that the story is being generated.

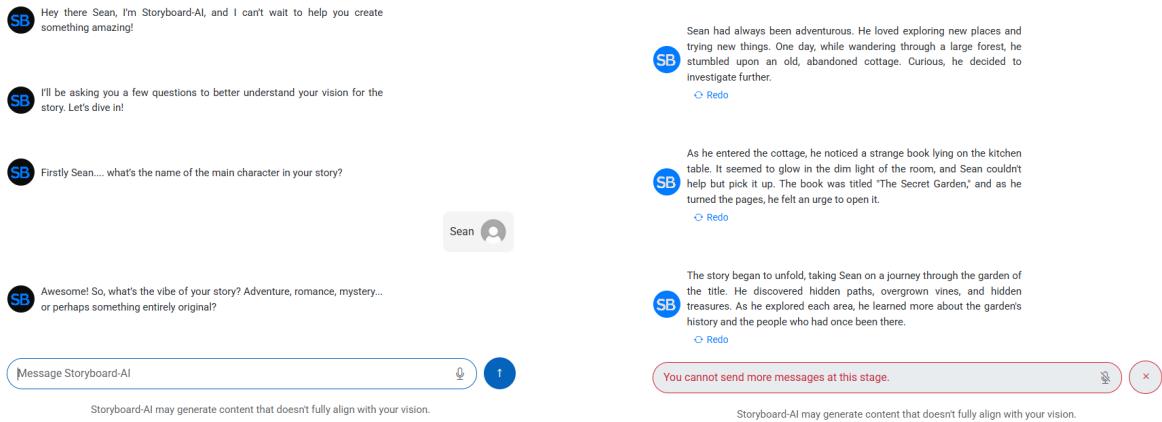


Figure 5: Story generation interaction: user prompt and response.

The system displays a loading GIF while the story is being generated by the LLM, maintaining user engagement during the process. Once the story is ready, the loading GIF is removed, and the narrative is segmented into smaller paragraphs. Each paragraph is presented to the user one at a time, accompanied by a realistic typing animation that mimics real-time interaction, similar to a user typing. Between paragraphs, a "typing dots" animation introduces brief pauses, further enhancing the sense of live conversation. The interface is designed to emulate popular chatbots like OpenAI's ChatGPT and Google's Gemini, delivering a familiar and intuitive experience. The entire story is combined into a cohesive narrative and stored with a unique story ID in the browser's localStorage, ensuring easy retrieval and continuity.

After the story is generated, a custom title is crafted using a specialized prompt designed to encapsulate the story's mood, theme, and key message. This title is dynamically displayed to the user and saved along with the story in conversations.json for record-keeping and future retrieval. The system also validates the title format to ensure clarity and readability. Additionally, a story summary is generated in the back-end by sending the full story to the LLM for concise summarisation. To enhance the storytelling experience, the system generates an image prompt based on the story's content and interacts with the Stable Diffusion model to create a visually compelling representation (see subsection 3.4.2).

Next, the story and accompanying image are presented, the system prompts the user with a "yes" or "no" button to decide whether they want to view the story summary. Once this is addressed, another prompt asks if the user wishes to view the entire story in a continuous format. Selecting "yes" displays the full story without breaks, while still allowing the text to be clicked and copied.

Users can download the story as a professionally styled PDF, which includes the generated title, story content, and any accompanying image. The PDF generation process ensures a polished appearance: the title is prominently displayed as the header, the story content is formatted with appropriate margins and line spacing for readability, and the image—if present—is converted to Base64 format and embedded seamlessly. Additionally, metadata such as the author and relevant keywords are included for improved categorisation and organisation.

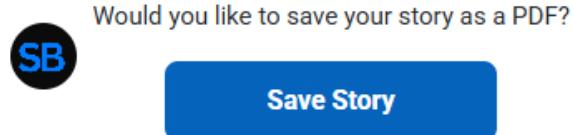


Figure 6: Button to save Story as PDF

Finally, the user is shown a message: *"Want to create a new story? Click Here."* Clicking this link refreshes the page, allowing the user to seamlessly start creating a new story.

3.4.2 Image Generation

The story is accompanied by an image generated using the Stable Diffusion model. This process begins with the complete story being sent to the LLM, which generates a descriptive prompt specifically tailored for Stable Diffusion (see subsection 3.4.4). The generated prompt is then used to create an image via the Stable Diffusion model, and the image is stored in the `GeneratedImages` folder. During this process, the user is notified that the image is being generated, and a loading GIF is displayed until the image is ready. Once the image is created, the loading GIF is replaced with the generated image.

To ensure the image meets high-quality standards, it undergoes an iterative upscaling process. This method increases the image resolution in gradual steps, preserving details and minimizing distortions. A LANCZOS resampling filter is used during each step of upscaling to maintain the sharpness and clarity of the image. The final result is a high-resolution version of the generated image, which is displayed to the user and embedded into the story's PDF (see subsection ??).

The upscaling process is essential to improve the visual quality of the image, making it suitable for storytelling and professional presentation. Users are also given the option to regenerate the image if they are unsatisfied with the initial result (see subsection 3.4.3). This ensures that the final image aligns closely with the user's expectations and enhances the overall storytelling experience.



Figure 7: Example: Image generated by Stable Diffusion

3.4.3 Redo Functionality

The chat system provides redo functionality for story paragraphs, title generation, and image generation, giving users the flexibility to refine their output. For titles, a "Redo" button allows users to request a new title, during which the system temporarily displays the feedback message "*Rewriting Title...*" while generating the updated title. This new title is dynamically updated in the user interface and synchronized with the back-end to ensure it is saved persistently in the stored data. This process allows users to fine-tune their storytelling elements without losing progress (see subsection 3.4.4).

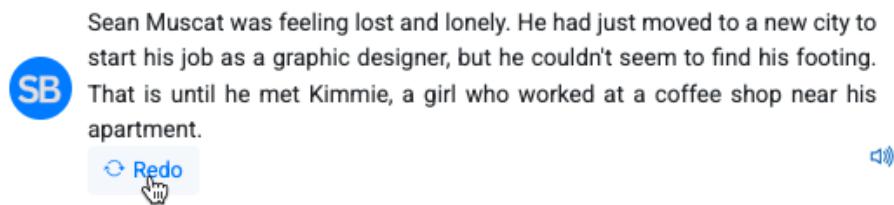


Figure 8: Redo Paragraph

For story paragraphs, users can request rewrites by selecting a specific paragraph. The LLM is tasked with generating a revised version of the paragraph while retaining the overall context of the story. The updated paragraph seamlessly replaces the original in both the user interface and the back-end storage.

For images, the redo functionality removes the previously generated image from the server and initiates the creation of a new image. During this process, a temporary loading indicator is displayed to inform the user that the system is processing the request. Once the updated image is ready, it is displayed with a timestamp appended to its URL to prevent caching issues. This ensures users always see the latest version.

Each redone component—whether a title, paragraph, or image—overwrites its corresponding entry in the `conversations.json` file. The updated version is then saved in the back-end, ensuring consistency across the system. When the user generates the final PDF, only the latest versions of the story elements are included. These redo features enhance the user experience by enabling iterative improvements to both textual and visual components, allowing users to make as many adjustments as needed before finalizing their story.

3.4.4 Prompt Engineering

Prompt engineering was a meticulous process, as Orca:Mini required precise and carefully crafted instructions to ensure it stayed within scope and avoided hallucinations. To achieve the best results, multiple variations of each prompt were tested to identify the most effective and versatile version. These prompts, covering tasks such as story generation, title creation, rewrites, image prompt generation, and summaries, were all developed in-house. Once finalized, the prompts were stored in the back-end and never exposed to users on the front end, ensuring seamless functionality and consistency across the system.

Generation the main Story: The prompt used to generate the story employs a detailed and structured approach to guide the storytelling model in producing immersive and engaging narratives. It begins by defining the model's role as a "*masterful storyteller*," ensuring alignment with the desired tone, expertise, and storytelling style. Key variables, such as the main character's name, the story's theme, and the desired ending tone—provided earlier by the user—are incorporated into the prompt to personalise the output. These inputs are gathered through the system's interactive query process, ensuring the story is tailored to the user's vision.

The prompt directs the model to craft vivid settings, natural dialogue, and compelling events, while maintaining a seamless narrative flow presented in simple paragraphs without titles or headings. The story length is limited to 400 words to maintain conciseness and user engagement. It also specifies a positive and uplifting tone suitable for all ages, ensuring that the output aligns with the system's inclusive and imaginative ethos. Additionally, user-provided notes are incorporated to further customize the story. If no notes are provided, the prompt gracefully defaults to a generic message to maintain consistency.

To uphold ethical standards, the prompt includes explicit guidelines prohibiting content that involves violence, harm, or offensive material. If the narrative begins to deviate, the model is instructed to gently redirect the story to a respectful and hopeful tone. This balance between specific instructions and creative flexibility ensures the outputs are not only high-quality and user-centric but also aligned with the system's values of inclusivity, creativity, and positivity.

This carefully engineered prompt structure empowers the model to produce stories that are engaging, ethically sound, and tailored to the user's preferences, while also maintaining a professional and enjoyable storytelling experience.

Rewriting Paragraphs: For rewriting paragraphs, a carefully designed prompt ensures the rewritten text retains its original meaning while enhancing readability and structure. The model is instructed, as a "*writing expert*," to preserve the paragraph's tone, context, and meaning, while rephrasing with varied wording and improved sentence flow. The rewritten version is concise, thorough, and similar in length, with no labels or commentary included. These clear guidelines ensure high-quality, faithful rewrites optimised for comprehension, while allowing creative rephrasing. The same approach is applied to rewriting titles.

Generating Summaries: For story summarization, a precise prompt guides the model to condense the story into a clear, concise overview, focusing on key events, central conflict, and resolution. This ensures summaries are coherent and informative. A fallback mechanism defaults to "*No summary generated*" if the model fails to produce a valid result, ensuring reliability and robustness in the process.

3.4.5 Multimodality within SB-AI

Text-to-Speak: The system's text-to-speech functionality leverages the browser's `SpeechSynthesisUtterance` API to provide auditory playback of messages. Through the `addSpeakerIcon` function, an interactive speaker icon is appended to messages once the typing animation is complete. Clicking the icon triggers the `speakText` function, which converts the paragraph text into clear, natural speech. This feature enhances accessibility, allowing users to listen to instructions or responses. Integrated with the `typeParagraphWithCursor` function, the text-to-speech capability is lightweight, efficient, and compatible with modern browsers.

Speech-to-Text: The microphone button serves as a dynamic interface element, adapting to the status of microphone permissions with visual feedback such as icon changes and placeholder text updates. By leveraging the `navigator.permissions.query` API to manage microphone access, the system ensures seamless and responsive voice recognition capabilities. Users can speak naturally, with their spoken words processed and directly entered into the input field. Importantly, the system automatically stops listening when it detects a pause or no new words being spoken, limiting microphone usage for ethical considerations. This natural language integration enhances accessibility and convenience while ensuring user privacy and minimizing unnecessary resource use.

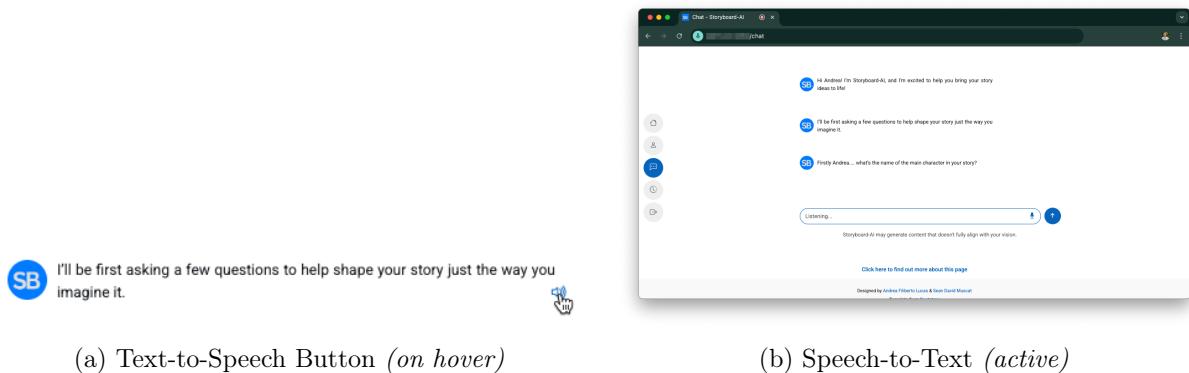


Figure 9: TTS & STT within SB-AI

3.4.6 Disclaimer and Footer

The page footer contains a clickable link labeled "*Click here to find out more about this page,*" which reveals an informational section about the SB-AI Chatbot. This section includes a critical disclaimer stating that the stories are fictional and intended for entertainment purposes, with a note about the potential for occasional errors or deviations in content. The disclaimer is essential for transparency, ensuring users understand the creative nature of the chatbot and its limitations. The section also credits the Ollama platform and its Orca-Mini model, providing links for further exploration. This transparency fosters trust while offering users insight into the technologies powering the experience.

3.5 Archive - (*archive.html*)

The Archive Page in SB-AI serves as a centralised interface for users to efficiently manage their saved stories. Designed with a user-centric approach, the page dynamically personalises the experience by incorporating the user's first name in the title, "*Your Story Archive*". A visually appealing hero section with a background image enhances the aesthetics while maintaining consistency with the platform's design.

At the core of the Archive Page is the ability to manage and interact with saved stories through the *Story Archive Manager*. Stories are presented as scrollable, interactive story cards, each equipped with actionable features. Users can view the full story text in a modal with justified formatting for improved readability. The modal also includes a *Copy to Clipboard* feature, allowing users to easily duplicate the story content. For enhanced usability, hover effects on interactive elements, such as colour transitions, improve visual feedback and engagement.

A prominent feature is the real-time search functionality, enabling users to filter stories by keywords or phrases. As users type into the search bar, the list of stories dynamically updates, and a counter displays the number of matching results. This feature enhances efficiency by allowing users to quickly locate specific stories. Additionally, the platform enables users to download stories as PDFs, leveraging the `pdfmake` library to generate formatted documents that include styled titles and content. This capability ensures that users can easily store and share their stories outside the platform.

To maintain data integrity and privacy, the system fetches user-specific stories from the back-end using the `/assets/json/conversations.json` endpoint. Only stories belonging to the logged-in user are displayed, ensuring a secure and personalised user experience. The page adapts gracefully to different scenarios; for instance, if no stories are available, a message is displayed, and non-relevant elements such as the search bar are hidden to maintain a clean and focused interface.

Users also benefit from advanced content management features, such as the ability to delete stories securely. A confirmation modal is displayed before any deletion to prevent accidental loss of content. This feature is complemented by the system's intuitive and responsive design, which simplifies interaction and minimizes errors.

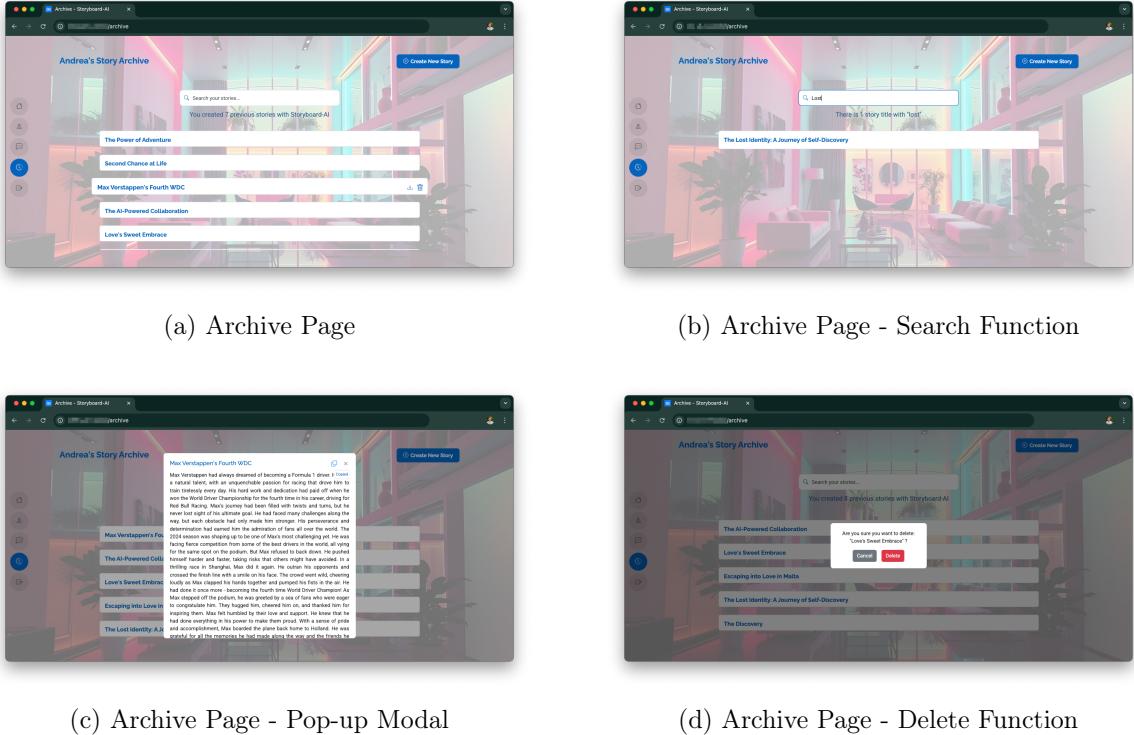


Figure 10: Overview of Archive Page Features in SB-AI.

3.6 404 Page (*404.html*)

The 404 Page in SB-AI provides a user-friendly experience for handling missing or invalid URLs. Featuring a creative and humorous message, the page dynamically displays rotating phrases such as “lost in the Spice Mines of Kessel” or “exploring the Jedi Temple,” ensuring engagement even in error scenarios. A prominent call-to-action button redirects users either to the home page or their personalised dashboard, based on their login status.

This page maintains the platform’s cohesive visual style with a responsive layout, smooth animations, and an immersive background image. By combining aesthetic design with practical functionality, the 404 Page minimizes user frustration and encourages re-engagement with the platform.

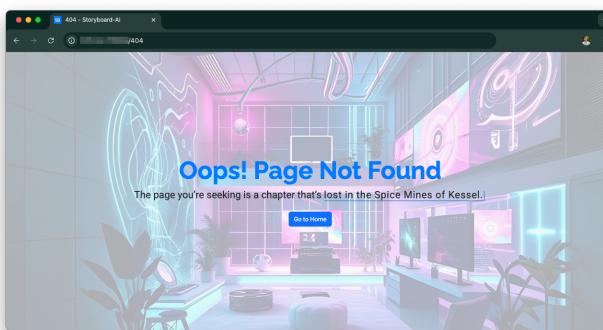


Figure 11: SB-AI 404 Page

4 Evaluation of Generated Story (*Examples*)

4.1 Story 1: Adventure in The Jungle

SB-AI Generated Text: Sean was always looking for an adventure. He had grown up in a small town, and while he loved the comforts of home, he couldn't help but feel restless. One day, he decided to explore the world. He packed his bags with clothes and supplies, and set off on his journey.

His first stop was a jungle - a place that sounded like a forbidden adventure waiting to happen. He had heard stories about the dense rainforests, the hidden waterfalls, and the mysterious animals that lived there. As he entered the jungle, Sean felt a sense of wonder and excitement. The air was thick with humidity, but he welcomed the challenge. He walked through the dense greenery, crossed the treacherous rivers, and climbed the steep hills. He saw colourful birds and monkeys swinging from tree to tree, and felt the spray of waterfalls as they crashed into him. But his journey wasn't without danger. Sean had heard stories of dangerous animals that lurked in the jungle - wild cats, snakes, and spiders. He kept a close eye on his surroundings and watched his step, but he never lost his sense of wonder.

After a few days of wandering through the jungle, Sean stumbled upon an ancient temple. The walls were covered in intricate carvings, and the floor was made of polished stone. Inside, he found a group of people who had been living there for years, unchanged by time. They welcomed him with open arms, sharing their knowledge of the jungle and their stories of adventure.

As the sun began to set, Sean realized that he had finally found what he was looking for. The adventure had come at a cost - he had lost his backpack and had to spend the night in the temple, but he didn't care. He had found something even more valuable - a sense of wonder, a sense of excitement, and a new appreciation for life. And as he slept that night, he dreamed of further adventures. But now, he knew that he would never forget the jungle and the people who had shown him the true meaning of adventure.



Figure 12: Image Generated for Story 1: Adventure in The Jungle

Reflection: This story was created with main character "Sean", "Adventure" theme, "Happy ending" and an additional feature of "The story is set in a jungle". The title "Adventure in The Jungle" described the story correctly and one of the paragraphs was redone and it still appeared correct and coherent in the story. Overall this was a very good test story that could satisfy users, with each feature working properly and intertwined together. It took slightly long to get the full story, but this is because of the specifications of my computer, which are a low end GPU (NVIDIA GeForce GTX 1050 TI) and CPU (Intel Core i5-8400K), which effects runtimes significantly.

4.2 Story 2: The Discovery

SB-AI Generated Text: Clara had always been fascinated by ancient civilizations and their artifacts. She dreamed of finding something truly special - a treasure that would change her life forever. So, she decided to travel to Malta, a small island in the Mediterranean, where many believe there is a vast treasure hidden somewhere. Clara packed her bags and set off on her journey.

As she explored the island, she stumbled upon an old castle that was said to be associated with the treasure. She asked the locals about it, but they didn't know much. They warned her to avoid the workers' quarters, as something strange had happened there. Clara ignored their advice and continued her search. After hours of searching, she finally found an old map that showed the location of the treasure. She followed the map and found a hidden chamber deep in the mountains. Inside, she saw a chest filled with gold coins, precious gems, and ancient artifacts.

Her heart leaped with joy as she realized she had found it - the treasure she had been searching for. Overwhelmed by her discovery, Clara realized that the true treasure wasn't the gold itself, but the experience of discovering something truly special. She returned home a wealthy woman, but she never forgot the lesson she learned on her journey - to always trust your instincts and follow your heart.



Figure 13: Image Generated for Story 2: The Discovery

Reflection: The generated story demonstrates the effectiveness of the SB-AI system in aligning with user-provided prompts. The main character, "Clara" was specified in the prompt alongside a "Mystery" theme and a "Happy Ending." Additional details, such as the main character exploring for treasure in Malta, were accurately incorporated into the narrative. The language model successfully crafted a cohesive and engaging story, delivering an enjoyable and imaginative result. While the English was slightly off in certain areas, the system allowed for regeneration, ensuring the user could refine the story further if needed.

The accompanying image generated by Stable Diffusion effectively captured the mood and setting of the story, further enhancing the overall experience. Impressively, this was achieved efficiently and quickly on a MacBook Air M2, highlighting the system's ability to deliver high-quality outputs with minimal computational resources. The seamless integration of text and image generation showcases the platform's potential for creating immersive storytelling experiences while maintaining a user-friendly and resource-efficient design.

5 Ethical Considerations

5.1 User Privacy

A key privacy feature of the SB-AI project is the "Remember Me" option, which allows users to maintain a persistent session for up to 30 days via cookies. This feature enhances user convenience by eliminating the need to repeatedly log in during this period. However, after 30 days, clearing the cache helps protect user privacy by removing cookies and associated session data, ensuring that sensitive information is not left vulnerable to unauthorised access on shared or public devices. This approach balances user convenience with robust privacy safeguards, limiting the potential risks of long-term cookie storage.

In line with the project's commitment to data protection, no prompts or user responses are stored on the system. Only the story generated by the AI is saved in the back end, ensuring that user input remains transient and cannot be used for unintended purposes. This design choice significantly reduces the risk of exposing any information, as no sensitive details provided by the user are retained once the story is generated. By focusing on the end product rather than the interaction process, the system upholds strict standards for user confidentiality. As well as this, all stories are tied to their individual accounts, so no cross-access between user's stories can occur.

User passwords in the SB-AI system are securely protected using industry-standard hashing techniques provided by the 'werkzeug.security' library. When a user creates an account or updates their password, it is processed through the 'generate_password_hash' method, which applies a robust hashing algorithm. This ensures that the stored password is transformed into a cryptographically secure format, making it computationally infeasible to reverse-engineer the original password. By storing only the hashed version of the password in the 'credentials.json' file, the system mitigates the risk of data breaches or unauthorised access. Even if the hashed passwords were exposed, the lack of plaintext information provides a strong layer of defence, adhering to best practices for user authentication and data security.

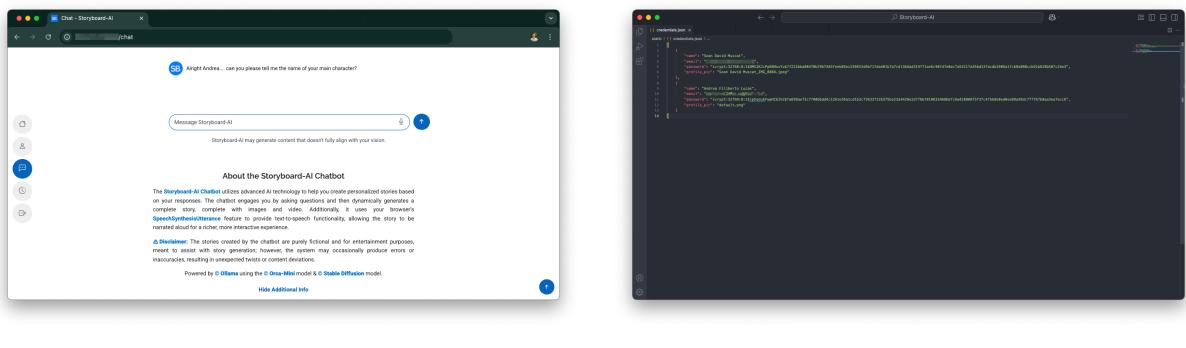


Figure 14: Ethical Mitigations

The microphone feature in the SB-AI system is designed to enhance user privacy by limiting the collection and processing of voice data. The feature automatically stops listening when pauses in speech are detected or when the user ceases talking, ensuring that no unnecessary or unintended audio is captured. This approach minimises the risk of over-collection of sensitive data, as only the spoken input necessary for the current interaction is processed. By implementing this real-time monitoring and control mechanism, the system adheres to privacy-first principles, offering users confidence that their voice data is handled responsibly and transiently, without being retained or processed beyond its immediate purpose.

Running the AI locally on the user's device enhances privacy by ensuring all data and interactions remain on-device. Unlike cloud-based solutions, this approach eliminates risks associated with data transmission or external storage, aligning with privacy-by-design principles. It gives users full control over their data while reducing reliance on third-party services.

The chat page includes a disclaimer that highlights SB-AI's reliance on advanced models like Ollama, Orca-Mini, and Stable Diffusion, along with their potential limitations, such as occasional inaccuracies. By providing links to the models' sources and setting realistic expectations, the system fosters transparency, trust, and informed user engagement, demonstrating an ethical approach to AI deployment. This decentralized method not only improves user privacy but also supports sustainability by lessening dependence on extensive data centres, since these extensive data centres could lead to data breaches or other ethical concerns.

5.2 Content Generation and Filtering

A proactive approach to ethical content generation is integral to the system's design. Specific filters and guidelines are implemented to screen out offensive, abusive, or explicit content during both the text generation and image creation processes. These filters ensure that the system adheres to a set of moral principles while maintaining a space for creative expression. Furthermore, user-generated prompts are monitored to prevent the creation of content that could be considered harmful, ensuring compliance with community guidelines. As a further line to protect users from explicit content, Stable Diffusion also cannot create images deemed explicit, as can be seen in Figure 15, it would generate a blank image.

Moreover, Stable Diffusion has built-in restrictions to avoid generating branded or copyrighted material. This ensures that the system adheres to intellectual property laws and respects copyright regulations, mitigating the risk of unintentional violations. The model has been trained from its creators with appropriate safeguards to ensure that generated content does not resemble specific brands, logos, or proprietary designs, maintaining the integrity of the creative outputs.

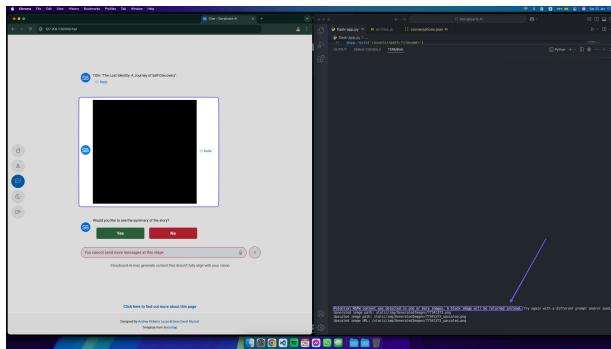


Figure 15: Blank image generated for NSFW (not-safe-for-work) content request

5.3 Accessibility and Inclusivity

Accessibility is prioritized by optimizing the system for performance without compromising quality, ensuring it works on lower-end devices. Models like Orca-Mini and Stable Diffusion have been tested to generate good results on such hardware, enabling inclusivity. While processing may take longer, this trade-off ensures all data remains stored locally, enhancing privacy.

6 Challenges & Current Limitations

The development of SB-AI has been accompanied by several technical and creative challenges. Initially, reliance on TinyLlama posed significant hurdles, as the model frequently deviated from intended prompts, resulting in incoherent narratives. This limitation necessitated a transition to Orca-mini, which significantly improved narrative coherence and consistency. However, this process highlighted the importance of staying adaptable and continuously evaluating tools to ensure alignment with project goals.

One of the most pressing challenges has been hardware limitations. Running larger AI models locally proved impractical on less powerful systems, leading to increased latency for generating stories, images, and other AI-driven functionalities. To mitigate these issues, caching mechanisms were implemented to reduce loading times during subsequent runs, enhancing efficiency without sacrificing privacy or local execution. Nonetheless, the system's initial startup remains lengthy due to the background loading of image generation and local LLM components, which will require further optimization in future iterations.

Integrating visual elements into text-based storytelling also presented unique obstacles. Stable Diffusion, used for image generation, excels at producing artistic and abstract visuals but struggles with creating realistic imagery. This limitation has impacted the platform's ability to fully meet user expectations in scenarios requiring photorealistic outputs. Additionally, generating effective prompts for Stable Diffusion required extensive experimentation, with much of the process relying on Orca-mini to optimize and refine these prompts. Tasks such as title generation, story summarization, and paragraph rephrasing similarly required iterative prompt engineering to ensure quality and accuracy.

Another key challenge lies in providing a more personalised storytelling experience. While the current implementation allows users to customize story elements, there remains a need for more granular control, such as the ability to pause and modify the AI's progress during story generation. User feedback, collected from testers outside the development team, has emphasized the importance of these interactive features and informed ongoing refinements to the platform.

Despite these challenges, the project has achieved significant milestones. The integration of image generation and speech-to-text functionalities has enhanced the platform's interactivity and immersion. The addition of a PDF export option has further streamlined the user experience, enabling easy saving and sharing of stories. The ability to directly read, write, and override JSON files from the frontend has simplified data management, marking another critical improvement.

7 Conclusion & Future Work

This project successfully combines text and image generation to create an innovative and interactive storytelling system. By seamlessly integrating a language model to generate compelling narratives and leveraging it to craft prompts for image generation, the platform ensures that both processes work harmoniously to deliver a cohesive and immersive user experience. The system's easy-to-use interface empowers users to create, refine, and adapt their stories with features such as regenerating specific parts of the narrative or visuals, reviewing and editing generated content, and downloading their creations as PDFs. Additionally, the ability to access, search, and manage previously saved stories enhances long-term engagement and provides a structured approach to storytelling.

A cornerstone of the project's design philosophy has been the emphasis on privacy and ethical AI usage. By running all models locally on the user's device, the system eliminates the need for external servers, ensuring that sensitive data remains entirely under the user's control. This approach aligns with privacy-by-design principles, fostering trust and safeguarding confidentiality. Furthermore, the reliance on local AI models not only protects user data but also enhances accessibility by allowing the platform to function without continuous internet connectivity. These design decisions underscore the project's commitment to ethical practices in AI deployment.

Looking ahead, the platform offers numerous opportunities for expansion and enhancement. A key area for future development involves the addition of video generation capabilities, allowing users to create fully animated video versions of their stories. This would take the storytelling experience to the next level by providing rich multimedia outputs. Furthermore, incorporating support for multiple languages would significantly broaden the platform's accessibility and usability, enabling a more diverse user base to create stories in their native languages.

Additional enhancements to the web platform could include more intelligent and adaptive interaction features, such as personalised suggestions for story elements and dynamic prompts that respond to user preferences in real time. Expanding accessibility through a mobile companion app would also offer users greater flexibility, allowing them to create and manage their stories on the go while maintaining the seamless integration of features from the web platform.

Overall, this project has demonstrated the feasibility and impact of combining AI-driven storytelling with a user-focused design. By prioritizing privacy, user engagement, and adaptability, the system has laid the groundwork for a versatile platform that aligns technical innovation with ethical considerations. Future iterations will aim to optimize performance, expand personalization options, further enhancing the storytelling experience and accessibility for all users.

8 References & List of Resources Used

- [1] S. Mukherjee, A. Mitra, G. Jawahar, S. Agarwal, H. Palangi, and A. Awadallah, "Orca: Progressive Learning from Complex Explanation Traces of GPT-4," arXiv preprint arXiv:2306.02707, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2306.02707>. [Accessed: Dec. 15, 2024].
 - [2] Ollama, "Orca-mini Library," [Online]. Available: <https://ollama.com/library/orca-mini>. [Accessed: Jan. 26th, 2025].
 - [3] Ollama, "TinyLlama Library," [Online]. Available: <https://ollama.com/library/tinyllama>. [Accessed: Jan. 26th, 2025].
 - [4] BootstrapMade, "Free Bootstrap Themes and Website Templates," [Online]. Available: <https://bootstrapmade.com/>. [Accessed: Jan. 26th, 2025].
 - [5] Hugging Face, "CompVis Stable Diffusion v1.4," [Online]. Available: <https://huggingface.co/CompVis/stable-diffusion-v1-4>. [Accessed: Jan. 26th, 2025].
 - [6] Pixabay, "Load, Loading, Process, Wait, Delay," Animated GIF. [Online]. Available: <https://pixabay.com/gifs/load-loading-process-wait-delay-35/>. [Accessed: Jan. 26th, 2025].
-