StorySprout: On-Demand AI-Generated Children's Stories Leveraging AI to Create Customized Children’s Stories in a Web Application

CS4991 Capstone Report, 2025

Ryan Chung
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
msc2pd@virginia.edu

ABSTRACT

Early in their development, elementary school-aged children often have trouble finding motivation to take on a daunting task like learning to read. To solve this issue, I created an application, called StorySprout using NextJS, Firebase, and OpenAI’s ChatGPT that allows children to enter many different parameters to generate stories specific to them. The application required the ChatGPT 3.5-turbo API as well as the typescript Firebase package. I used ChatGPT’s 3.5-turbo API to feed in parameters and receive a generated story, and used Firebase to save user data such as stories, views, likes and total story generations. The application assists children in their reading development by creating customized stories that can include their family, friends, or teachers. To bridge the gap between an online story and a physical book, the application will need to have an interface that is more similar to a book. This would include a page flipping animation, images, and even the ability to buy a physical copy.

1. INTRODUCTION

Early in their development, elementary school-aged children often find it challenging to stay motivated when learning to read. Traditional books and school lessons fail to cater to their interests or connect with them on a more personal level, making reading feel like a chore rather than an adventure. Parents and educators alike acknowledge the need more for more engaging and personalized reading materials. Stories that not only entertain but reflect the realities and interests on a per child basis.

StorySprout allows educators and parents to ditch one-size-fits-all content, giving children the ability to include family members, friends, favorite activities and other meaningful details. Young learners can see themselves and their world reflected in the stories they read. This makes them more likely to remain curious, enjoy the reading process, and develop stronger literacy skills over time.

2. RELATED WORKS

A vital early influence on the creation of StorySprout came from utilizing OpenAI’s ChatGPT platform. ChatGPT illustrates how large language models can product highly engaging, contextually relevant text at the click of a button (OpenAI, 2023). Watching ChatGPT provided insight into the power of dynamic storytelling. Having a system that can generate different narratives based on user inputs sparked an idea about how we can customize content for younger readers. By interacting with ChatGPT, I shifted focus toward creating an application where children feel involved on a personal level, reinforcing their enthusiasm for literacy.

In addition, learning from the official Next.js documentation helped shape strategies for building a web-based platform that could efficiently render content for diverse user inputs (Next.js, 2023). This guidance informed decisions about how best to structure the project, making it more intuitive for children, parents and educators alike. Meanwhile, the Firebase Codelabs documentation provided examples of real-world implementation workflows, showing the importance of robust data handling and user-centric design (Firebase, 2023). These resources collectively influenced the foundational approach to StorySprout's overall architecture, user experience and ability to scale in a way that accommodates interactive storytelling.

3. PROJECT DESIGN

The core of StorySprout is built around Next.js for frontend interfaces, Firebase for database management, authentication, and serverless functionalities, and OpenAI's ChatGPT 3.5-turbo API for story generation.

The smooth experience created for StorySprout revolves around Next.Js. This framework was selected for its ability to provide server-side rendering and improved performance with a level of simplicity that led to easy development. The pages created in Next.Js handle all user interaction including inputting parameters into the stories. Not only does the frontend created with Next.js allow the user to input parameters into the story, it allows the user to see other generated stories, like other stories created by users, and view and manage their own story creation.

Firebase acts as the API for all StorySprout app functionalities. For context, an API is an application programming interface. This allows our Next.js frontend, to communicate with backend processes such as our story generation workflow. The cloud functionality of Firebase Cloud Functions allows the Next.js application to send the parameters chosen by the child into a function that Firebase runs in the cloud. The function will then take in the parameters set by the child and format a specific prompt already set in the function code to send to OpenAI. The function then parses the response returned by OpenAI's API and sends it back to the application for display and use. Once it arrives to the Next.js frontend, we parse and display the response for the child to read. However, Firebase cloud functions are only used for specific custom functionality: in this case the story-generation workflow.

Firebase also handles the database management and authentication portion of the application. It allows users to save their data to their profile while also giving the development team insights into usage statistics. Access to the database storage and authentication management is managed through the Firebase Typescript package provided by NPM, a package distribution library. Through that package, the Next.js frontend can query the database and confirm authentication with the user before they access any data. This allows for safe and consistent data across all users on

StorySprout also currently supports a purchase model through the system of tokens. The conversion is a token per singular story generation. The infrastructure for this system is handled through a platform called RevenueCat, a platform that allows us to manage subscriptions through a payment platform like Stripe. It holds a separate database and assigns a specific identification number for each payment tied to the user. From there, we use our Firebase database to manage how many tokens each user has and use our Next.js application to update the token values on the account as they generate tokens.

Finally, this all comes together in an ecosystem where the Next.js app is the center where users will view, generate and read stories. Firebase handles all of the background tasks and database management, allowing the user to have a low latency, streamlined, efficient application to utilize for generating stories.

4. RESULTS

The most compelling aspect of StorySprout is that in mere seconds a child has a captivating story at their disposal. The integration can take a child's input of characters, setting, and actions and seconds later provide them with coherent, compelling stories, all thanks to ChatGPT 3.5-turbo. In addition, during the prototype evaluation and feedback session, evaluators revealed that this was a great motivator for continued engagement with the reading experience; for instance, children were far more likely to want to read about something that had to do with them or at a location they recognized. Parents and teachers claimed that they found their children returning to the integration more frequently; the customized stories appeared to create more opportunities for literacy engagement.

Also, beyond creating a love for reading, the app interface provides teachers with an assessment beyond reading levels of a child's development. Because StorySprout uses Firebase to assess analytics (how many stories are created, how often children revisit the same stories), it can provide averages of interests and developmental changes over time. Such data would be beneficial to teachers and parents because it will help them to understand what interests children, potential mastery of comprehension for future stories, and adjustments necessary for at-home reading facilitation. Although the project will need formal usability and comprehension testing in the future, the anticipated results show that StorySprout will keep the love for reading alive with future possibilities like print on demand or AI-created images being more relevant to stories.

5. CONCLUSION

StorySprout fills an essential void within the need to read by rendering reading more personal and captivating for young, elementary-aged minds. The ability to take characters, settings and plots developed from a child's day and input them into the app transforms reading from a simple, passive "I have to do this" task into more of an active, personal endeavor. Powered by Next.js, Firebase and ChatGPT, it links school-taught concepts with real-life applications during one of the most impressionable times of their lives. For parent and teacher use, StorySprout offers simple logistical solutions for any stories generated while also providing an analytics history of use and activity, allowing the adults in these children's lives to have a better understanding of their growing interests and emerging reading habits.

Aside from fostering child engagement with reading, StorySprout sparks the potential for on-demand, automated and AI-generated content in the future. Incorporating something like this personalized storytelling option into a web application could spur other innovations in the areas of comprehension and literacy. This project exemplifies how innovative technology can transform a simple reading process into something engaging, personalized and, in the end, successful for children, their caregivers and educational institutions.

6. FUTURE WORK

There are multiple opportunities for StorySprout to expand in the future. For example, the interface could be upgraded to enhance a reading experience akin to a real book (pages turning, illustrations inserted) would capture children's attention and get educators on board as well as publishers interested in a digital and physical reading hybrid. A user study for usability testing would assess potential improvements to the interface as well, providing direction on what a child-friendly interface should include to best operate for the purpose of teaching children reading and writing.

Additions such as customizing AI image generation and an ability to print completed stories on demand would only enhance StorySprout both within the classroom and at home. Furthermore, additional analytics for teachers and parents to understand a child's comprehension of the story, exposure and adjusted interests would allow for more personalized response and support when they, too, know what the child is going through. Even translating StorySprout into different languages for story creation, for instance, would help learners in more multicultural environments. These all suggest a connected world where technology, creativity, and the essentials of reading intersect, fostering a lifelong love of stories.

REFERENCES

Firebase. (2025). Integrate Firebase with a Next.js app. https://firebase.google.com/codelabs/firebase-nextjs#0

OpenAI. (2022, November 30). ChatGPT. Openai.com. https://openai.com/index/chatgpt/

Next.js. (n.d.) *Next.js by vercel - The REACT framework*. Next.js by Vercel - The React Framework. https://nextjs.org/