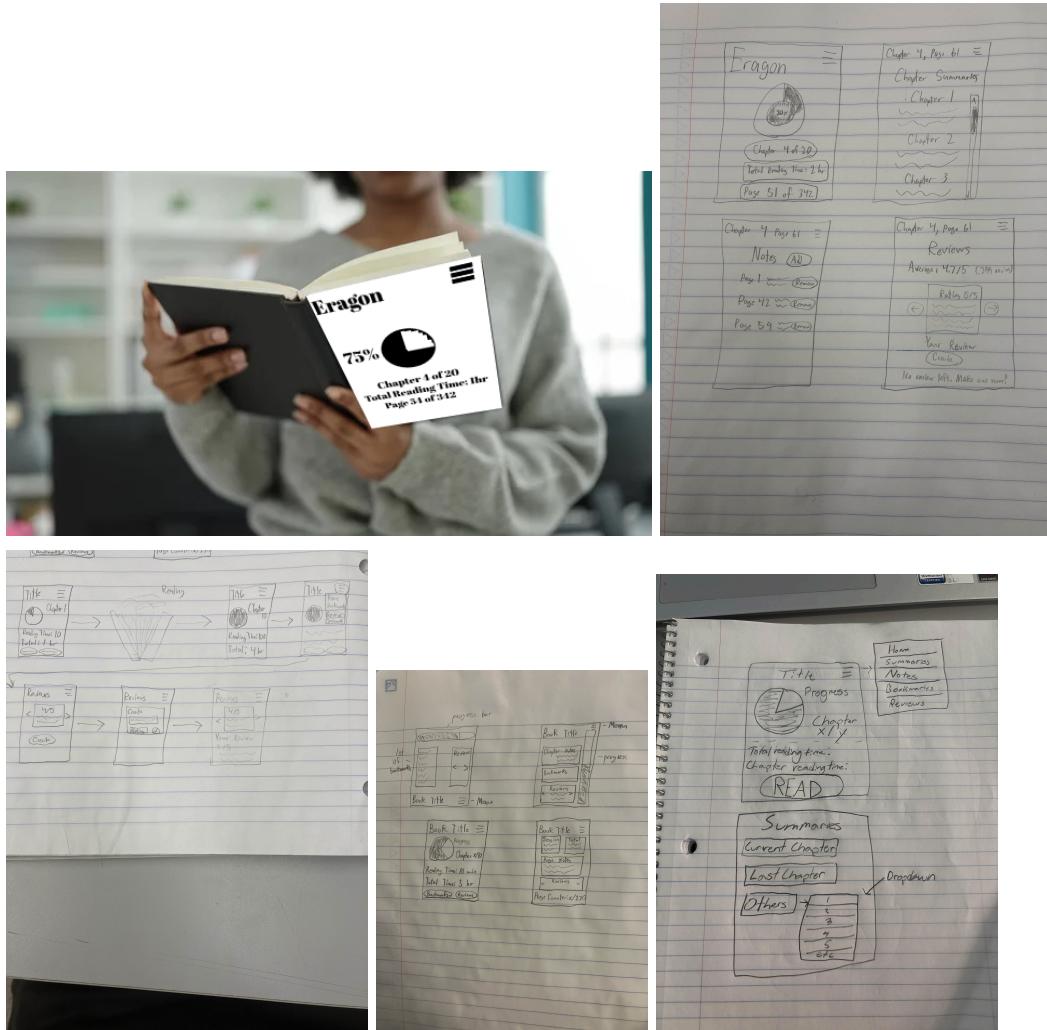


Smart Book

Our smart book project is an interactive upgrade to a book designed to give a user a better reading experience through the addition of chapter summaries, notes, bookmarks, and reviews. Additionally, it will allow the user to see their current status in the book with real time updating reading progress and time spent on each chapter.

Design Work



We wanted the design to be sleek, and easy for the user to use. We learned in our interviews that we needed the design to be very easy to use, as a book is very primitive and nearly everybody on Earth knows how to use a book. In order to do this, we started with sketching, by making a handful of normal sketches, storybook sketches, and hybrid sketches. From these, we were able to start our first prototypes.

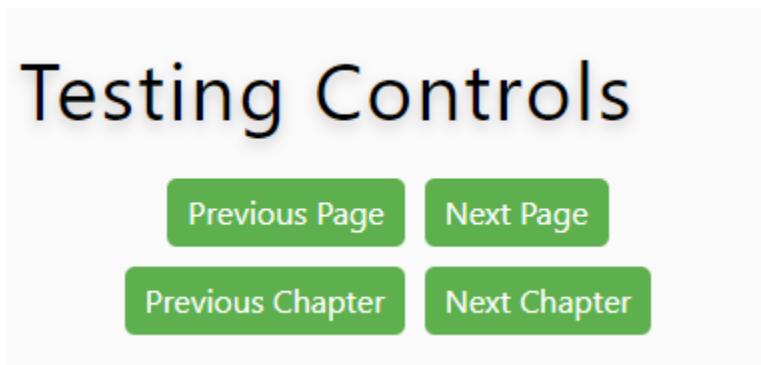
Key Goals:

- **Simplicity:** The layout we created is clean and uncluttered, letting the user to truly focus on the content vs the design.
- **Consistency:** The UI elements used throughout the project such as buttons, fonts, colors, etc., are used throughout the whole project giving the project a consistent look.
- **Responsiveness:** The design adapts to different screen sizes, not limiting the trial user to any one device. The final product will be able to fit on any book, regardless of the shape or size.
- **Accessibility:** The color contrast, fonts, and interactive components are all designed to be easily used for all users, regardless of educational background.

Interface Details

Main Components:

1. Hamburger Bar
 - a. While generic, the hamburger bar allows the user to easily access the other features the “smart” book has to offer.
A blue square icon with a yellow 'e' on the left and three horizontal lines on the right, representing a hamburger menu.
2. Testing UI
 - a. The testing UI is simply built for the demo, allowing the user to simulate reading pages/chapters easily.



- b.
3. Progress
 - a. The progress on the home page keeps track of the last page read, along with what chapter you are on. It also has time read for both the current chapter and whole book (it is simulated at the moment, but in a real book it would be true).



b.

4. Chapter Summaries

- a. Shows a summary of the chapter the user is currently reading, along with all previous chapters. Allows the user to recall the general story of the book.

The screenshot shows a dark-themed mobile application interface. At the top, the title 'Chapter Summaries' is displayed in yellow. Below it, 'Chapter 1' is shown in white. The summary text for Chapter 1 is: 'Eragon, a 15-year-old farm boy, finds a mysterious blue stone in the Spine, a mountainous forest near his home in Carvahall. He hopes it will bring money to help his family through the winter.' Below Chapter 1 is 'Chapter 2' in white, with the summary text: 'Eragon returns home with the stone but is unable to sell it at the local butcher, Sloan, due to its unknown origin. Disappointed, he keeps the stone, unsure of its value or purpose.' Below Chapter 2 is 'Chapter 3' in white, with the summary text: 'Eragon overhears the village storyteller, Brom, telling stories about the legendary Dragon Riders, long extinct but once protectors of the land. The stories fascinate Eragon, though he has no idea how relevant they will soon become to his life.'

b.

5. Notes

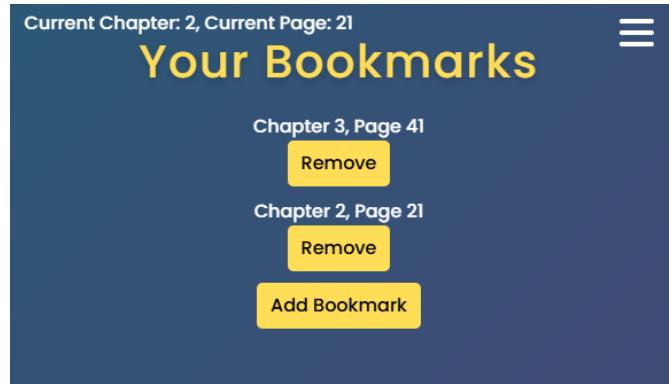
- a. Users can add, view, or remove notes in order to have an overall better reading experience.

The screenshot shows a dark-themed mobile application interface. At the top left, it says 'Current Chapter: 1, Current Page: 1'. On the right is a menu icon (three horizontal lines). The main title 'Your Notes' is centered in yellow. Below it is a button labeled 'Add Note'. At the bottom, there is a note card with the text 'Page 1: Test' and a yellow 'Remove' button next to it.

b.

6. Bookmarks

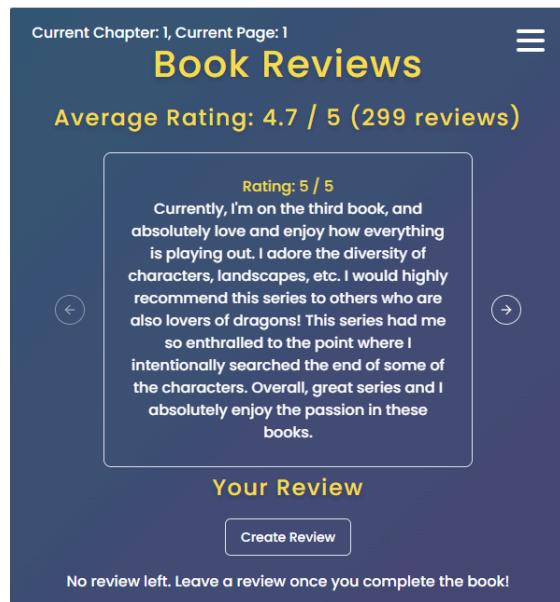
- a. Users can add bookmarks for easy referencing at another time.



b.

7. Reviews

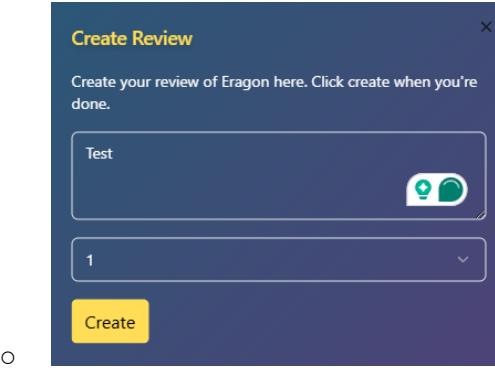
- a. Users can read and leave reviews for the book.



b.

Features and Controls:

- **Book Navigation:** Users can navigate to both different pages/chapters through the testing UI on the right hand side of the screen
- **Reading Simulation:** While the final product would ideally track time *actually* reading, the current implementation simulates reading time for each page/chapter.
- **Notes and Bookmarks:** Users can add both notes and bookmarks, allowing the user to reference either at any time. The bookmark will be more for keeping a spot in the book you may want to visit later, while the notes will be for any specific quotes/thoughts you had about a specific chapter/page.
- **Review System:** The review system is displayed in a carousel format, allowing the user to easily scroll through the different reviews, as well as leave their own for other readers!



Implementation

Libraries and Technologies

We used only a few different libraries across the project. Firstly, we obviously used **Svelte** as per the project requirements. Beyond this, we used the **Svelte Store** for the constant state management, so we could actively share the data. **Shadcn-Svelte**, which includes **TailwindCSS**, was also used for a few components, like the pop-up dialogs, the reviews carousel, and the scrollable summaries area.

Code Structure

We organized the project into a few different components, each having their own central job.

- **App.svelte:** This was the home of the project. Houses the navigation and routing logic
- **TestingUI.svelte:** Implements the controls for the user to simulate reading.
- **Components:** Includes **Home.svelte**, **Summaries.svelte**, **Notes.svelte**, **Bookmarks.svelte**, and **Reviews.svelte**. These files contain all the individual components for each important feature that the book has to offer.
- **store.js:** Handles the application's state, including the current chapter and page, along with different user interactions such as setting a bookmark or changing the page.

AI Use:

AI was used in the project to speed up the overall development. While the project was mostly done by us, Jacob used AI here and there to fix some general errors and increase production. He used Cursor, an up and coming AI code generation tool in order to push his ideas into production faster, along with finding different libraries/tools he wasn't familiar with.

Future Work

Implement live time-tracking on a user's statistics when reading, as right now there is a set time correlated with reading a page and finishing a whole chapter. We'd like to use this to give a reader more insight into which chapters they spent the most time on and to better understand their overall reading speed.

Add customization options for a user by letting them set a custom color scheme or organize the components on the home page to be in whatever design they want.

VIDEO:

<https://drive.google.com/file/d/14xM18HZr4kaqZs8KvTDkHK47A7Dt8TGE/view?usp=sharing>