Book Wishlist App: Portfolio Part 3

Objectives

Develop a web application for managing personal reading lists. The app allows users to track books they want to read, are currently reading, and have completed. The application should be intuitive, fast and easy to use.

Concept

The application addresses the challenge of maintaining a structured reading list by providing a digital platform where users can:

- Manage personal reading lists
- Create an account

Implementation

The development of this application followed a structured approach, moving from initial inspiration and design to front-end and back-end implementation, testing, and documentation.

Design Process

To create an intuitive and visually appealing interface, I began by researching existing book-tracking applications on Dribbble. After gathering inspiration, I sketched a wireframe on paper.

To refine the user experience, I used ChatGPT to evaluate the wireframe, comparing it to similar designs from Dribbble. This provided useful feedback on potential UX improvements, such as optimizing the layout for mobile responsiveness and making key interactions more intuitive.

Once the design was ready, I moved on to implementing the front-end using Next.js, React, Tailwind CSS, and shadcn. Tailwind CSS made styling highly efficient, while shadcn provided well-designed, customizable components that ensured a polished UI without requiring excessive manual styling. I focused on building reusable components, such as the footer, header and private route guard.

During this phase, I prioritized a smooth user experience by ensuring quick interactions and minimal page reloads. The state management was handled with React's built-in hooks, such as useState and useEffect, to dynamically update the UI as users added or modified books.

Back-end Development

After reaching a stable checkpoint with the front-end, I set up the back-end using Next.js Server Actions. Before implementing the back-end logic, I spent time reading documentation and watching videos about the latest Next.js features, ensuring I fully understood Server Actions and how they could replace traditional API routes for handling database operations securely.

For authentication, I explored different approaches by consulting ChatGPT, Claude, Reddit, and Stack Overflow. This research helped me determine the best method for integrating a secure authentication system with minimal complexity.

To store user data, I chose SQLite as the database due to its simplicity and efficiency for a small-scale project.

Testing & Debugging

After implementing the core features, I conducted testing and bug fixing. This involved checking for UI inconsistencies, fixing layout issues on different screen sizes and testing the back-end logic. I also performed manual user testing, simulating different workflows to catch edge cases such as invalid form inputs and unexpected interactions.

Documentation & Finalization

To complete the project, I wrote a README, showcasing the features of the application, explaining the installation process, how to run the application, and providing test cases.

Lessons Learned

One of the most valuable lessons from this project was the importance of planning. While previous projects I've worked on often involved jumping directly into coding, taking time to define the technology stack, structure the database, and plan functionalities significantly streamlined the development process. Another key takeaway was the opportunity to work with Next.js Server Actions for the first time. Although initially unfamiliar, they proved to be an efficient way to handle back-end logic within the same codebase, making development faster and more manageable.

Additionally, working with component-based UI frameworks like shadcn highlighted the advantages of using pre-built, customizable elements to speed up development without sacrificing design quality. This approach not only reduced development time but also ensured consistency in the user interface.

Reflections & Future Improvements

The final product successfully met the goal of providing a simple yet functional book-tracking solution. However, there are several potential improvements that could enhance the application. Introducing book recommendations based on a user's reading history could make the experience more engaging. Finally, adding advanced filtering and sorting options would allow users to organize their collection more effectively.