

1 Abstract

1.1 Phase 1

The project began with an extensive planning phase where I tried to find the most fitting technological stack. Prior to this project I had minimal practical experience with web development, making the first few days a period of intensive research and experimentation with various technologies. After evaluating multiple options, I decided to use Node.js for the server backend and MongoDB as the database system. For the frontend, I opted to work with HTML, CSS, and JavaScript, which in hindsight, made the GUI design particularly challenging. But by the time I realized the complexities involved, I was already too invested in my chosen technologies to consider adding a different framework.

1.2 Phase 2

At the beginning of the second stage, I focused on developing the backend. I used NPM to set up the server environment. Before starting the implementation, I needed to deepen my understanding of Node.js and JavaScript. This learning process took several days. It was essential for completing the backend.

I studied Node.js and JavaScript extensively. I read documentation, watched tutorials, and practiced coding exercises. Gradually, I became more confident in my skills. Once I acquired the necessary knowledge, I started implementing the backend.

The implementation of the backend went smoothly. I applied what I had learned efficiently. Setting up routes and handling database operations became straightforward. My preparation paid off, allowing me to complete the backend swiftly and effectively.

After completing the backend development, I redirected my focus to the front end. I began by establishing the HTML structure and implementing a simple CSS template. After that, I concentrated on the web application's logic, encompassing item rendering, retrieval, and updating. While the logical components posed no significant hurdles, I faced substantial challenges during the JavaScript implementation.

Despite knowing what I wanted to achieve, the most challenging part was developing functions efficiently. Adding new features often required reworking portions of the program. These issues mainly stemmed from a lack of initial practice with the technologies. Debugging and optimizing performance was also difficult. However, I persevered and continued refining my code, gradually overcoming these difficulties and improving the front end's functionality. This experience was valuable, enhancing my skills and knowledge in web development.

The project's most challenging aspect was designing the graphical user interface (GUI). I underestimated the time needed to create and style the GUI elements effectively. The increasing complexity made the task even more difficult. Despite these challenges, I remained committed to my chosen technologies and hesitated to consider adding another GUI framework.

If I were to start the project again, I would allocate significantly more time to the GUI development. This experience taught me the importance of planning and time management, especially when dealing with complex UI and GUI elements. It also highlighted the need for flexibility in technology choices to adapt to project demands.

After finishing the web app, I put the frontend and backend in Docker containers. Then, I created a Docker Compose YAML file to manage them together. To facilitate easier deployment and collaboration, I set up a GitHub repository with a detailed README. This README included instructions on how to use Docker Compose to get the application running.

1.3 Phase 3

Upon completing the deliverables for the second phase, a review revealed several missing elements. Notably, I lacked test cases and sample data, which I subsequently added. Additionally, I discovered several bugs related to the responsiveness of my web application.

Initially, the application displayed a table format on desktop screens, which became cumbersome on smaller devices. My first solution was to make the table movable, but this was not effective for mobile phones. I then decided to switch to a card-like display for smaller screens, ensuring a smoother user experience. This adjustment improved the application's usability across different devices.

2 Lessons Learned

This project was a profound learning experience, teaching me several valuable lessons:

1. The importance of thorough research and initial practice with chosen technologies must be considered. Investing more time in understanding the tools at my disposal would have saved considerable time and effort later.
2. Underestimating GUI design's complexity and time requirements can lead to significant delays. Future projects will benefit from a more balanced time allocation between backend, frontend, and UI/UX design.

In conclusion, despite the ups and downs, this project provided a comprehensive understanding of web application development and highlighted areas for future improvement. The challenges significantly enhanced my skills and knowledge in Webdeveloping.