

## Node

1. It went smoothly because the endpoints relied on the same validation rules. I used the Zod schema for both creating and editing books, which kept the validation consistent. The main change I made was switching from an insert query to an update query.
2. The deleted tests were straightforward because they mainly checked for a 204 response. The edit case, however, was more involved because I had to make many boxes. Such as successful updates, invalid authors, and invalid inputs. Using Vitest and Axios made it easier to test the API from the client's POV.

## React

1. I integrated editing and deletion into the main books table so users have an easy way to edit or delete anything they want. I additionally added a confirmation when users were going to delete something, so they know anything that they delete will not be recoverable.
2. For me, it was managing the component states across dialogs and keeping the UI in sync with backend changes.  
Note: I am not the best with design, so I did use an LLM to help me make the site look more presentable.

## Material UI

1. I did struggle with using it in the beginning, just because I was so overwhelmed by all the things it could do. So, I wanted a few YouTube videos, and used an LLM to help me understand the UI a little better. After that, using it was pretty easy, and it provided many useful components such as tables, dialogs, and alerts.
2. Refactoring my existing UI to use Material UI was easier than expected, because I took a little time to learn a little about Material UI. Most changes involved replacing HTML elements with Material UI components and adjusting props.