

Core

Deadline: Sunday of Week 6

Difficulty Level: Advanced

Est. Time: 03:00-04:00



MERNing with Books II (Core)

Build a front end for the MERning with Books back end featuring the ability to create, display and delete collection documents.



Learning Objectives

- Analyze the provided wireframe to identify components as views or common
- Implement CRUD operations for managing documents in the collection

Welcome to another **Core assignment!** Some students like to explore the assignments before they're finished reading through the lessons, and that's okay! It can be good for your brain to have a preview of what your future challenges might be. However, before you begin this assignment, it's important that you've first:

- Completed the preceding lesson modules
- Taken the knowledge checks to confirm your understanding
- Viewed lecture material related to the assignment topics
- Completed and submitted your practice assignments

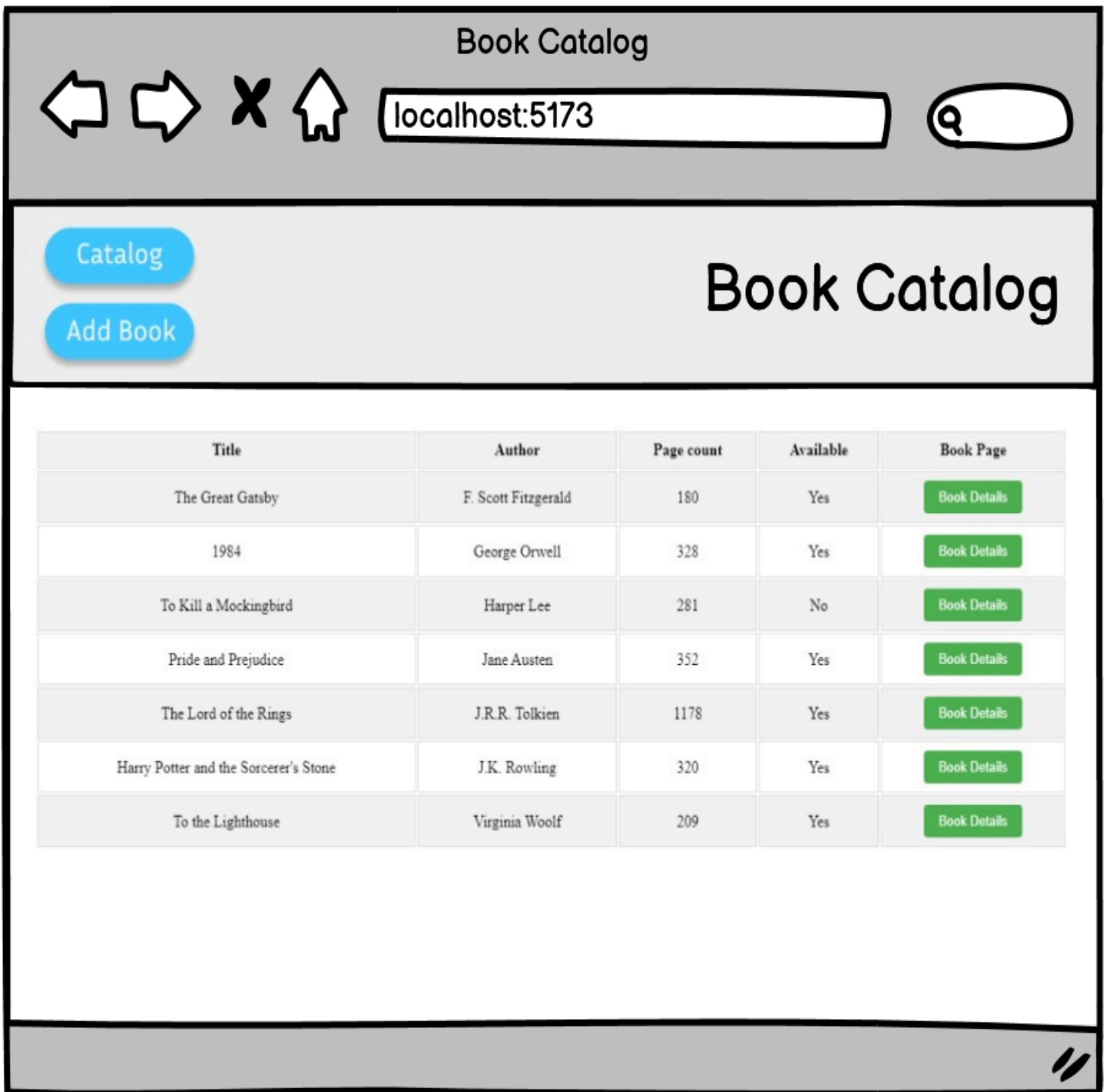
MERNing with Books II:

Using the back-end built from MERning with Books, it is time to add a front-end to this application. Below is a wireframe detailing how the React components should look to your users. Think carefully through the wireframe, planning ahead to solve design circumstances such as:

- which components are views or common components
- If components need to share a view
- If lifted state is necessary
- How to include front-end routing params in requests made to API routes requiring their own path parameters

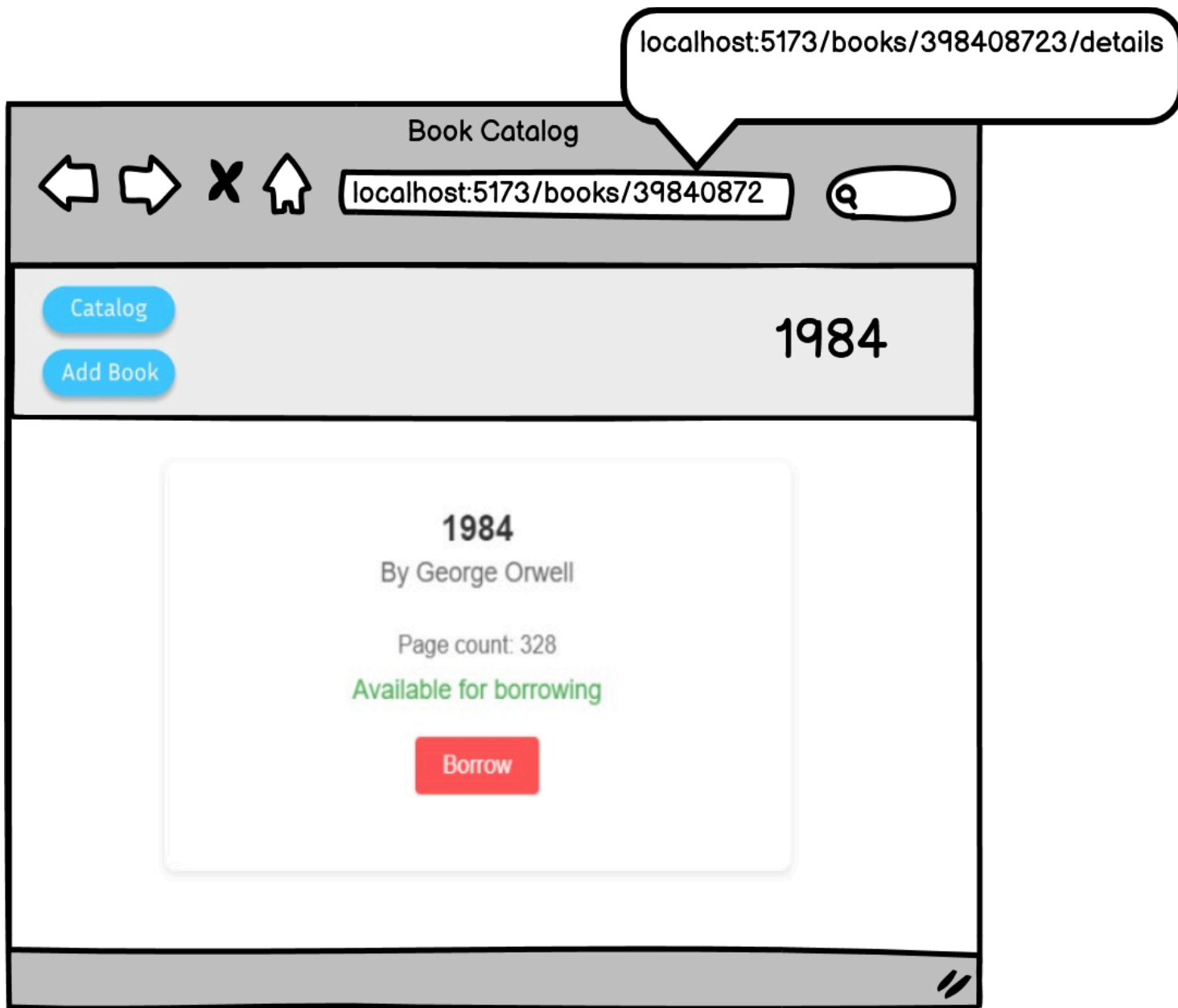
This assignment is part two of three assignments. In this middle assignment, begin by implementing the functionality to get one of the books or all the books in a collection, and add a book or delete a book from the collection. Stick to the visual identity and functionality as portrayed in the wireframe.

This should be the first thing users see upon loading your application:



A header section should show across all components, with buttons that link users to the component to display all books or the component to add a book to the collection. The book details link in each row should link to a component displaying details about the specific book whose Book Details link was clicked.

Details should be displayed according to the wireframe:



The "Borrow" button should delete the book document from the database and redirect the user to the component displaying all books.

A component to add a new book to the collection should resemble the wireframe:

The wireframe illustrates a web application titled "Book Catalog". At the top, there is a navigation bar with a back arrow, a forward arrow, a close button (X), and a home button (house icon). To the right of these icons is a search bar containing the text "localhost:5173/create" and a magnifying glass icon. Below the navigation bar, the main content area is divided into two sections. On the left, there are two blue buttons: "Catalog" and "Add Book". On the right, the heading "Add a Book" is displayed. The "Add a Book" section contains a form with the following fields: "Title" (text input), "Author Name" (text input), "Page Count" (number input with a spinner), and "Is it Available?" (checkbox). A blue "Add Book!" button is positioned below the form. The entire application is enclosed in a gray border with a small double-slash icon in the bottom right corner.

Once the book is added, users should be directed back to the component that displays all book documents.

After familiarizing yourself with the wireframe, begin to plan and develop your front end, ensuring the tasks in the checklist are complete before removing the `node_modules` folder from both the client and the server before submitting the assignment in a zipped folder:

- ☐ Configure front-end routing using the paths shown in the wireframe.
- ☐ Create view components for each of the components specified in the wireframe.
- ☐ Design a header that allow users to view all books or create a new book from any component in the application.

- ☐ Request all book documents, and display them in a table conditionally rendering text where appropriate.
- ☐ Include a column that allows users to view details about an individual book document.
- ☐ Request and display data about an individual book document, conditionally displaying data where appropriate.
- ☐ Allow users to "borrow" (delete from the db) a book from the details page and redirect to the display all component.
- ☐ Create a form component with two strings, a number and a boolean that sends a POST request that adds a new book document when submitted.