Project Overview

In the MyReads project, you'll create a bookshelf app that allows you to select and categorize books you have read, are currently reading, or want to read. The project emphasizes using React to build the application and provides an API server and client library that you will use to persist information as you interact with the application.

Get the Project

You have a few options to start developing this project:

- using your preconfigured Workspace
- forking and cloning the **start repository**
- starting from scratch with **Create React App**

Starter Code

Your Workspace is a development environment integrated into the Udacity Classroom and comes preconfigured with the MyReads project starter repository. You can both develop your entire in your Workspace as well as submit your project to be reviewed. For more information about your Workspace, check out the Your Project Workspace page.

If you'd like to work locally on your own computer, fork and clone the **starter repository**.

The code in both your Workspace and the starter repo contains all the CSS and HTML markup that may be used but omits the React code that is required to complete the project. This can save you some time if you don't wish to write all the CSS and HTML from scratch. The provided code will demonstrate a static HTML page of the finished application, but with no interactive functionality.

Starting from Scratch

If you'd like to start completely from scratch, you can use **Create React App** to create a new React application. However, even if you create your own application from scratch you'll still need to use the `booksAPI.js` **file** that comes with the starter template to interact with the backend API. Instructions for using the API methods are provided in the `README` in the starter template repository.

App Functionality

In this application, the main page displays a list of "shelves" (i.e. categories), each of which contains a number of books. The three shelves are:

- Currently Reading
- Want to Read
- Read
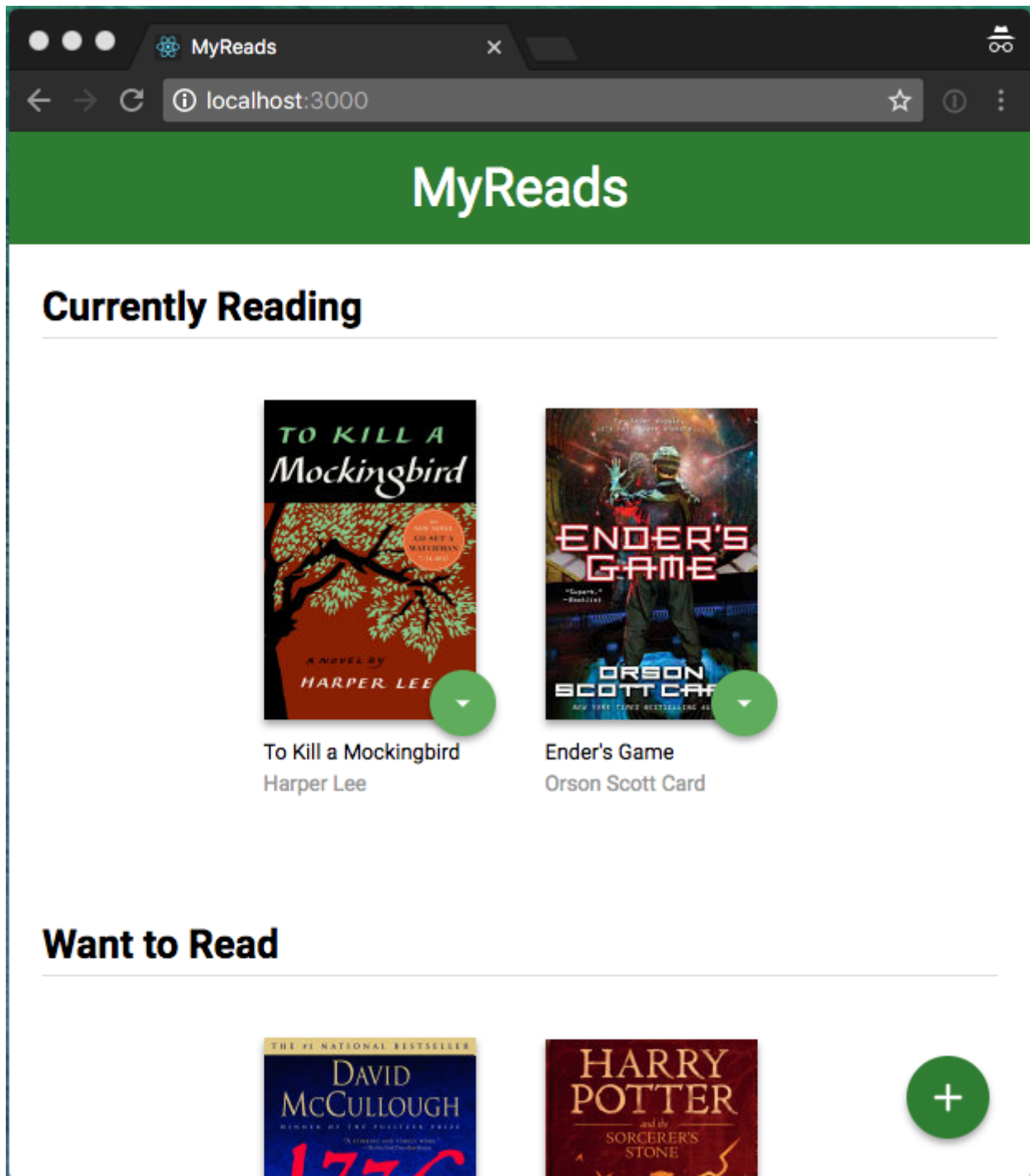
The static page (and your completed app) should look something like this.

Each book has a control that lets you select the shelf for that book. When you select a different shelf, the book moves there. Note that the default value for the control should always be the current shelf the book is in.
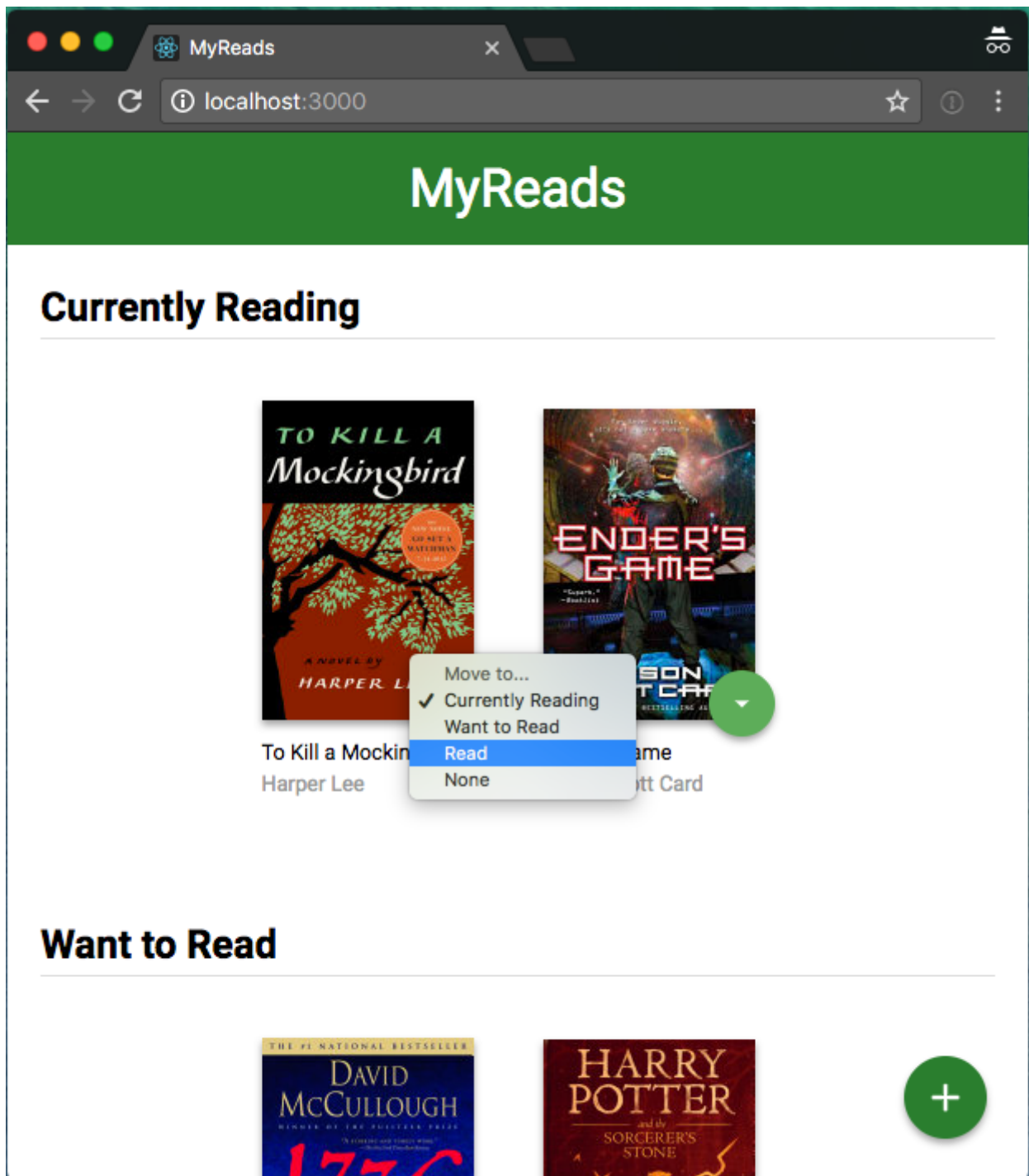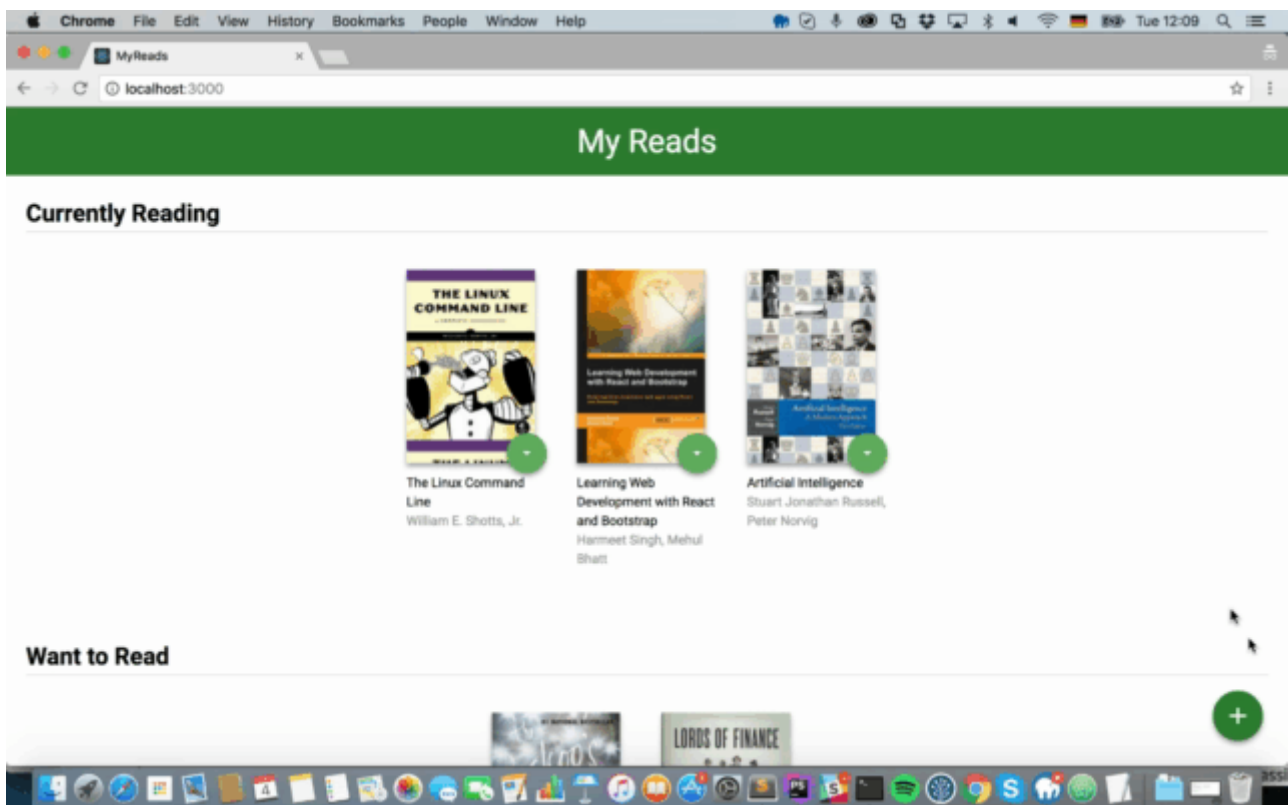
The main page also has a link to `/search`, a search page that allows you to find books to add to your library.

The search page has a text input that may be used to find books. As the value of the text input changes, the books that match that query are displayed on the page, along with a control that lets you add the book to your library. To keep the interface consistent, you may consider re-using some of the code you used to display the books on the main page.
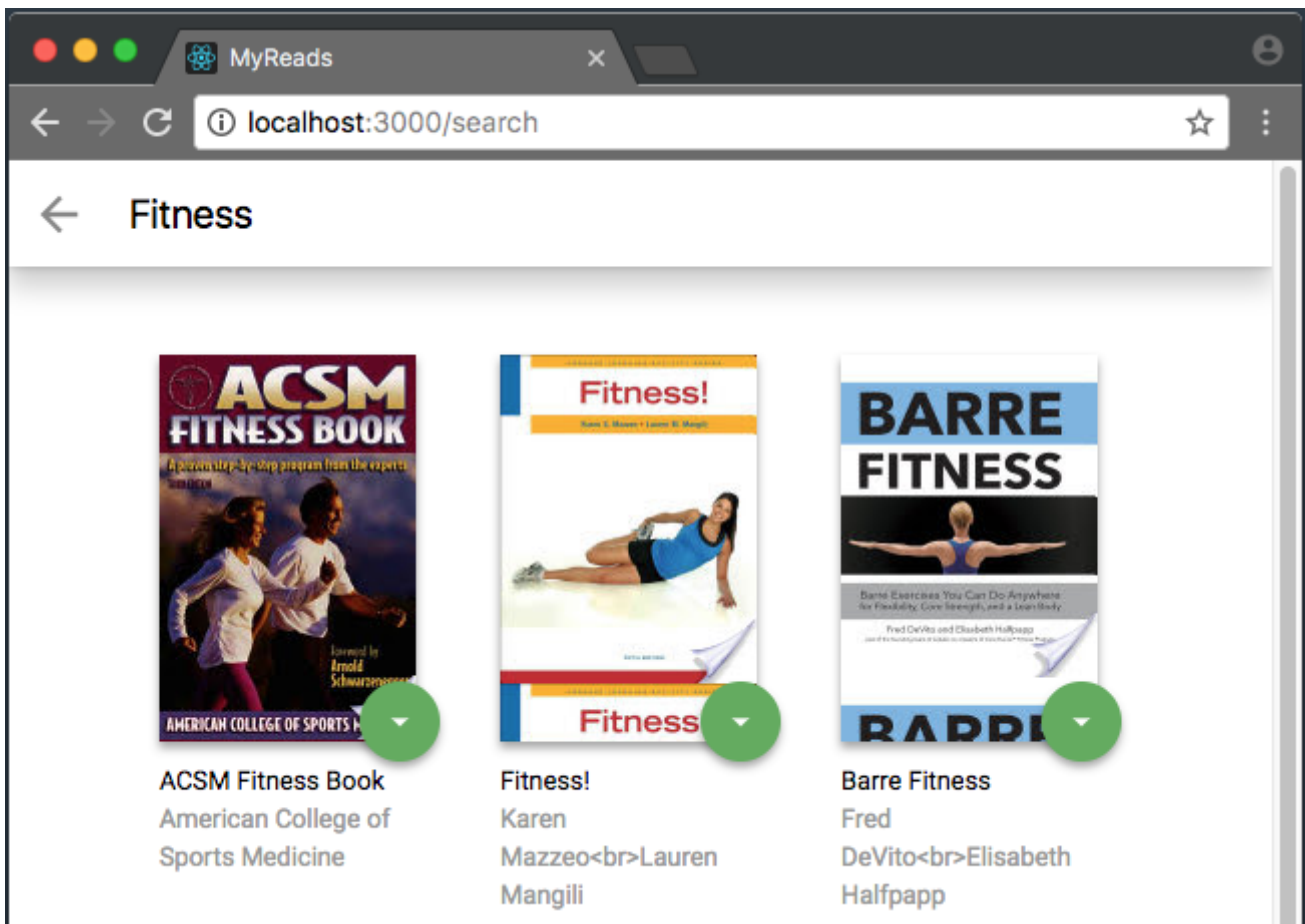
When a book is on a bookshelf, it should have the same state on both the main application page and the search page.

The search page also has a link to / (the root URL), which leads back to the main page.

When you navigate back to the main page from the search page, you should instantly see all of the selections you made on the search page in your library.

Submission Requirements

Your submission should include all of the files necessary to install and launch your web application on a local web server. For files that include JSX, please refrain from using the `.jsx` extension (you can prefer `.js`). You can assume that your reviewer will have npm installed on their machine.

Even if you don't end up using the **starter code**, you should probably still use **Create React App** to generate your submission since it's the easiest way to ensure you have everything the project reviewer will need to install and run your app.

Considerations

The focus of this project is on writing functional React code, not on making the page beautiful. Feel free to spend some time working on your layout and CSS if you want to, but the goal for this project is correct functionality.