Bookstore CRUD Application

Backend:

- Node.js
- Express.js

Frontend:

- React
- Vite

Testing:

- Mocha
- Chai

API

- https://6578511ef08799dc8044e5b2.mockapi.io/EBookStore
- Fetch this api in your backend and save it to a local file Books.json using `fs` module
- use and do CRUD from Books.json

`**1. Project Structure**`
Organize your project with directories like src, routes, controllers, and models.

`**1. Create Book DB**`
Create a models/Book.js file as your Database.

`**1. Express.js Routes**`
Create routes in the routes directory to handle CRUD operations for books.

`**1. Middleware**`
Implement middleware functions for error handling, and any other necessary features.

`**1. Write Tests**`
Create test scripts in the tests directory for your backend routes and middleware.

`**1. Run Tests**`
Execute tests using the Mocha .

Project Features

`### 1. Create a Book`
Implement a route and controller to add a new book to the bookstore.

`### 1. Get All Books`
Create a route and controller to retrieve a list of all books in the bookstore.

`### 1. Get a Single Book`
Implement a route and controller to retrieve information about a single book based on its ID.

`### 1. Delete a Book`
Create a route and controller to delete a book from the bookstore.

`### 1. Update a Book`
Implement a route and controller to update the information of an existing book

Frontend (REACT)

`### 1. Create React Components`
Develop frontend components for displaying and interacting with books.

`### 2. Integrate with Backend`
Connect frontend components to the Express.js backend using API calls.
`### 3. style your Frontend`
use everything we learned about CSS and make a good looking UI

EXTRA

- Implement Search / Filter and run Tests
- Implement Sort features by year and run Tests
- Filter by Price and run Tests