

Create a simple book store app using ReactJs or VanillaJS.

Using included “db.json” file and “json-server” package (<https://github.com/typicode/json-server>) run the server to get the data from mocked API.

Here are the endpoints (assumed its run on port 3000):

GET <https://upflex-book-store-api.herokuapp.com/books> - returns a full list of books

GET <https://upflex-book-store-api.herokuapp.com/books/{id}> - returns single book

GET <http://localhost:3000/availableBooks> - returns a short list of books (without details)

GET <https://upflex-book-store-api.herokuapp.com//homepage> - returns homepage content

Here is the list of required features in this app

HOMEPAGE (page 1):

- ● Display the homepage and show static and dynamic content (copy and images from API or from included assets or mockups)
- ● Show the list of available books as links or buttons
- ● User can click on a single book a be redirected to the detailed view
- ● At the top, show the current amount of items in the basket, when the basket is empty display “empty” otherwise display “X item(s)”
- ● Display the “VIEW BASKET” button which redirects the user to the basket page
- BOOK DETAILED VIEW (page 2):
 - ● Display the book page and show static and dynamic content (copy and images from API or from included assets or mockups)

- • Display the “Go Back” button which goes back to the homepage once clicked
 - • At the top, show the current amount of items in the basket, when the basket is empty display “empty” otherwise display “X item(s)”
 - • Show the “Add to basket button” which when clicked adds the currently displayed book to the basket and dynamically changes the basket information at the top
 - • Display “VIEW BASKET” button which redirects the user to the basket page
 - BASKET VIEW (page 3)
 - • Display the basket page and show static and dynamic content (copy and images from API or from included assets or mockups)
 - • Display the “Go to Homepage” button which goes to the homepage once clicked
 - • Show the list of items in the basket with a counter next to each item title; eg: x1
 - • Every item should have a remove button which decreases the amount of current item in the basket by 1
 - • Display a “PAY” button which upon clicking console logs current items in the basket in the following structure: [{ id: 1, quantity: 2 }, { id: 2, quantity: 1 }] etc.
- Some general rules:**



- - Follow the live review design but styling is entirely up to you. Please keep it clean and structured
- - When the app is refreshed the basket should be cleared
- Happy coding!