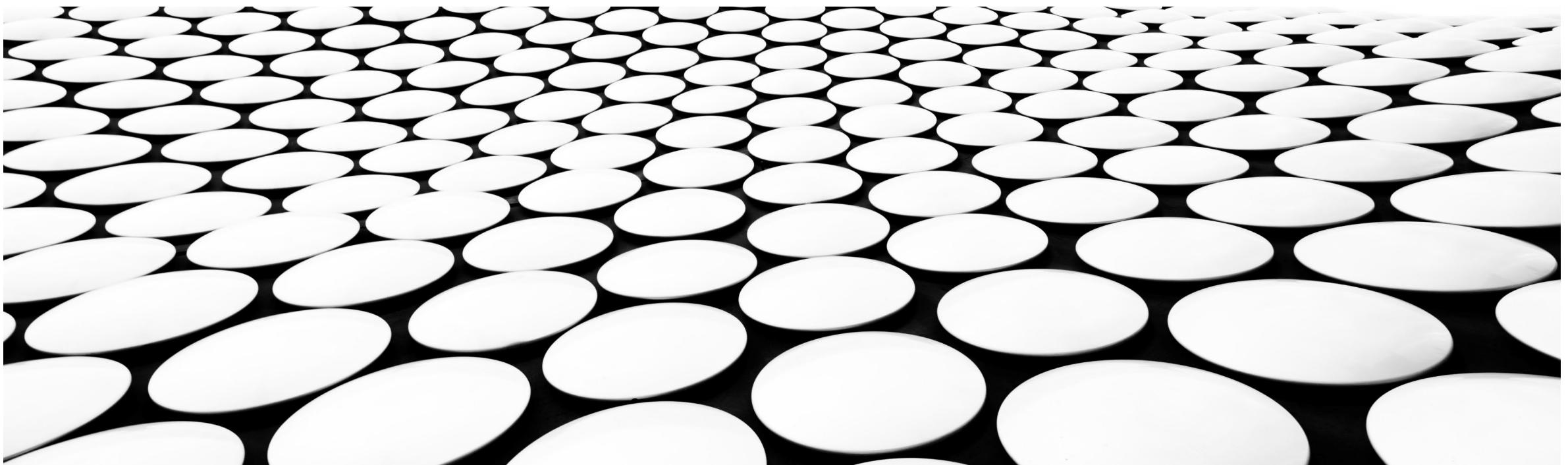


---

---

# S21 CHANGE QUANTITY AND DELETE BOOKS



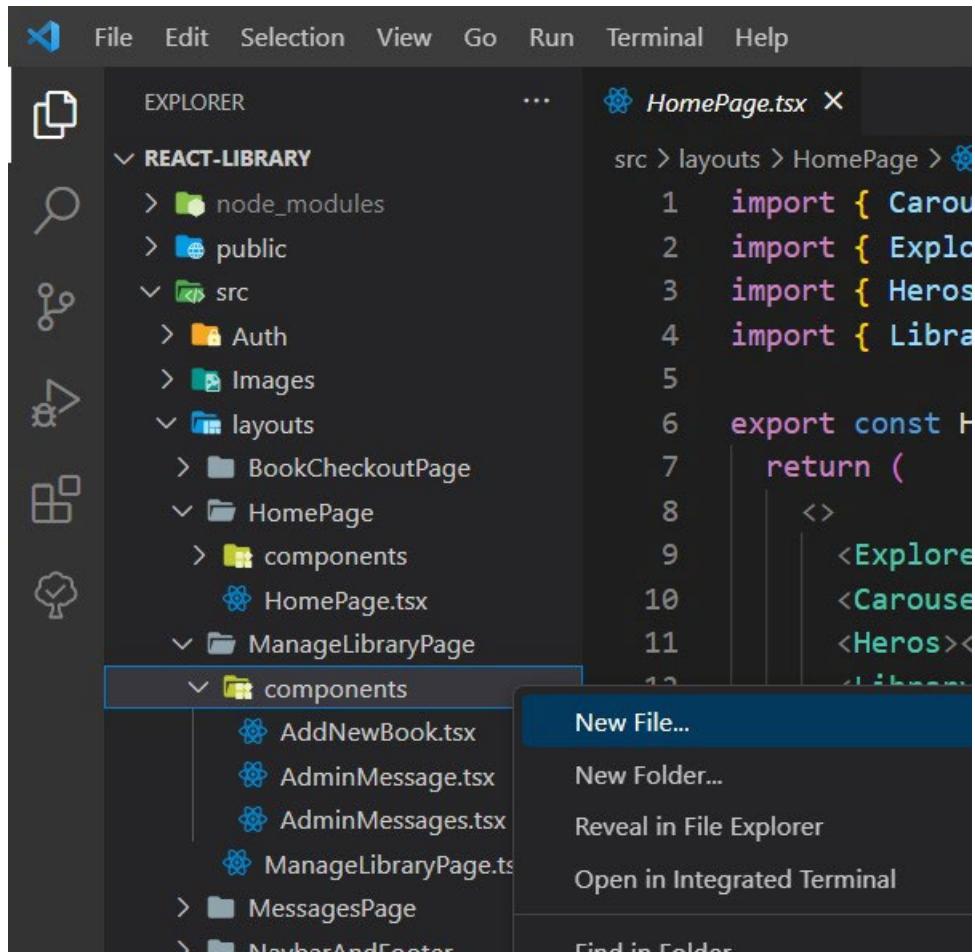
---

---

---

# 1. REACT-CHANGE QUANTITY OF BOOKS



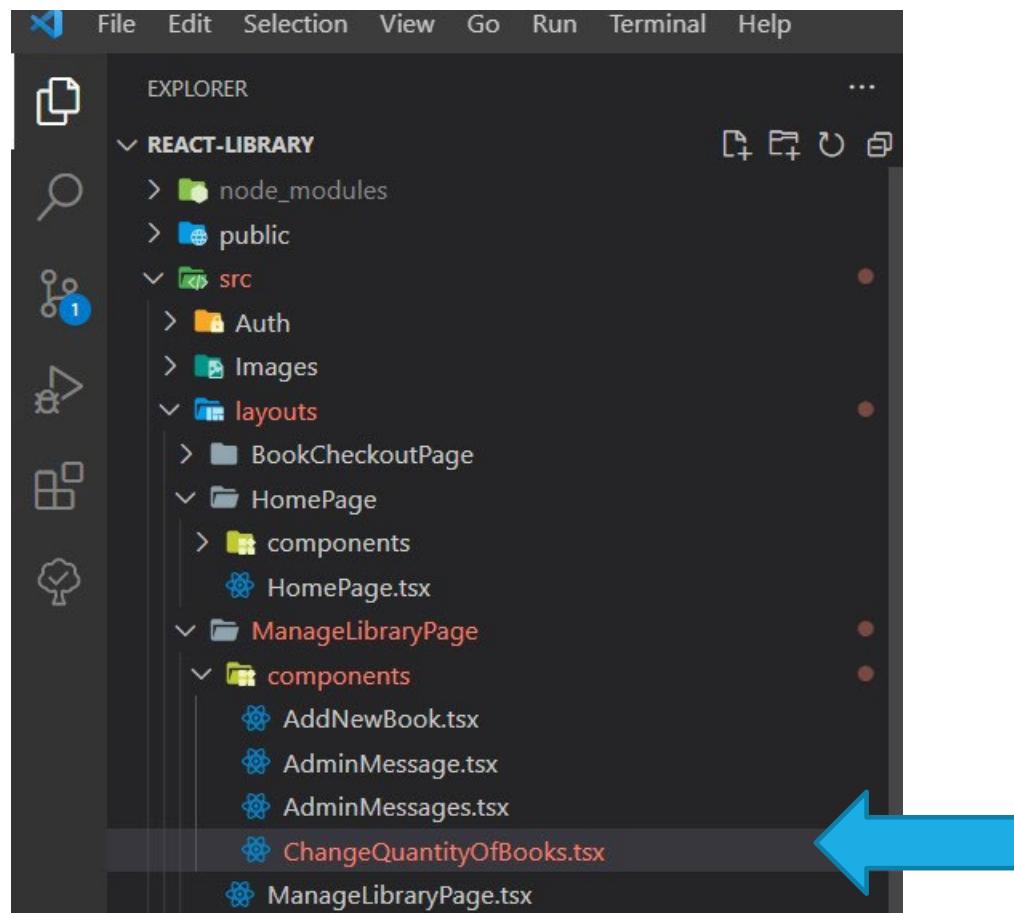


The screenshot shows the VS Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Explorer Sidebar:** Shows the project structure under "REACT-LIBRARY".
  - node\_modules
  - public
  - src
    - Auth
    - Images
    - layouts
      - BookCheckoutPage
      - HomePage
        - components
      - ManageLibraryPage
        - components
- Editor Area:** Displays the content of `HomePage.tsx`.

```
src > layouts > HomePage > HomePage.tsx
1 import { Carousel, Explore, HeroSection, LibraryList } from 'react-components';
2 import { Box, Flex, Grid, Text } from 'ui';
3
4 export const HomePage: React.FC = () =>
5   <>
6     <Box>
7       <Grid>
8         <GridItem>
9           <Explore>
10          <HeroSection>
11            <Carousel>
```
- Context Menu:** Opened over the "components" folder in the `ManageLibraryPage` section of the Explorer. The "New File..." option is highlighted.

Step 1, Go to our `ManageLibraryPage` folder, go inside our `components` folder, right click “new file”.



name it ChangeQuantityOfBooks.tsx.

Step 2, as always create export const ChangeQuantityOfBooks stucture.

The screenshot shows a Visual Studio Code interface with a dark theme. The left sidebar is the Explorer view, showing the project structure:

- REACT-LIBRARY**
  - > node\_modules
  - > public
  - < src
    - > Auth
    - > Images
    - < layouts
      - > BookCheckoutPage
      - < HomePage
        - > components
        - HomePage.tsx
    - < ManageLibraryPage
      - < components
        - AddNewBook.tsx
        - AdminMessage.tsx
        - AdminMessages.tsx
        - ChangeQuantityOfBooks.tsx

ChangeQuantityOfBooks.tsx - react-library - Visual Studio Code

HomePage.tsx ChangeQuantityOfBooks.tsx 2, U X

src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > [e] ChangeQuantityOfBooks

```
1  export const ChangeQuantityOfBooks = () => {
2    return()
3  }
4  }
5 }
```

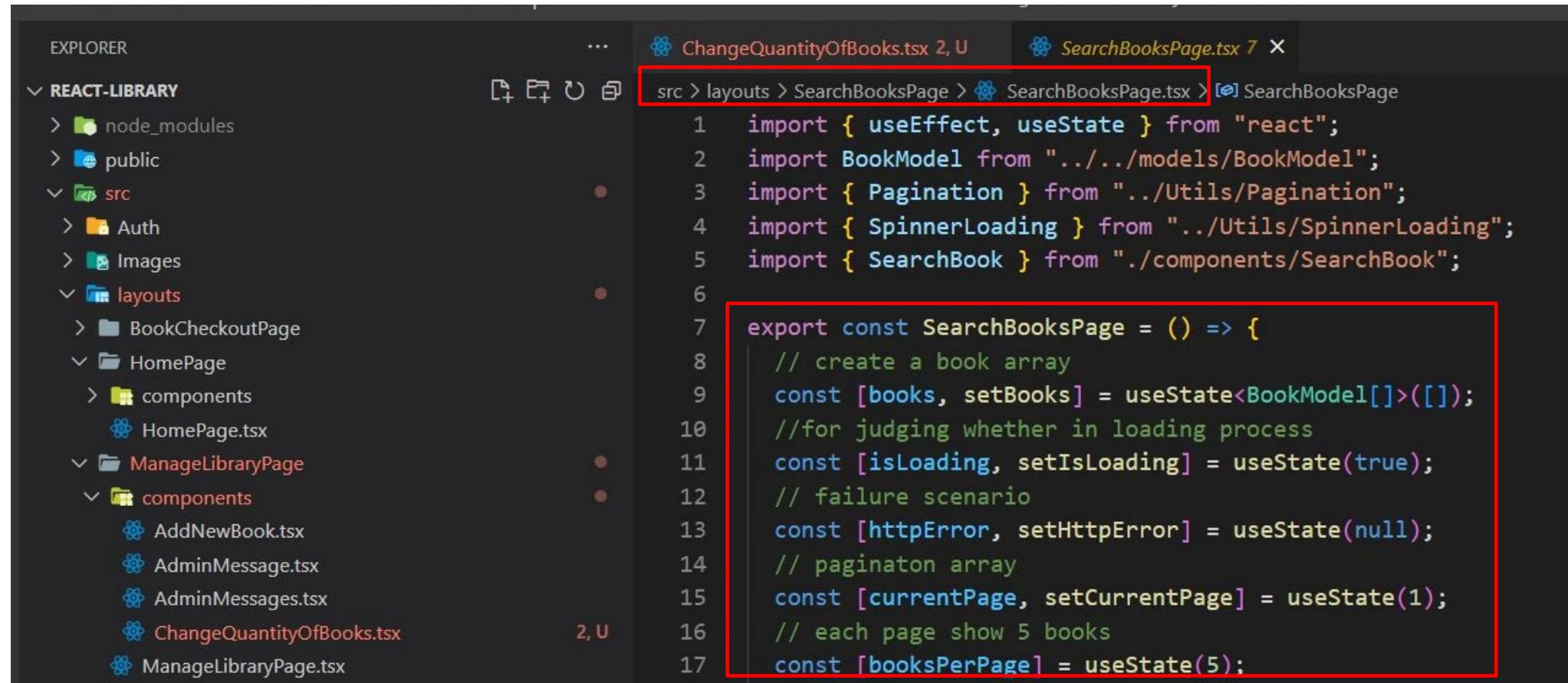
2, U

## Step 3 Create useState.

Open up our SearchBooksPage folder, go to our SearchBooksPage.tsx .

### 3.1 Grab following useState:

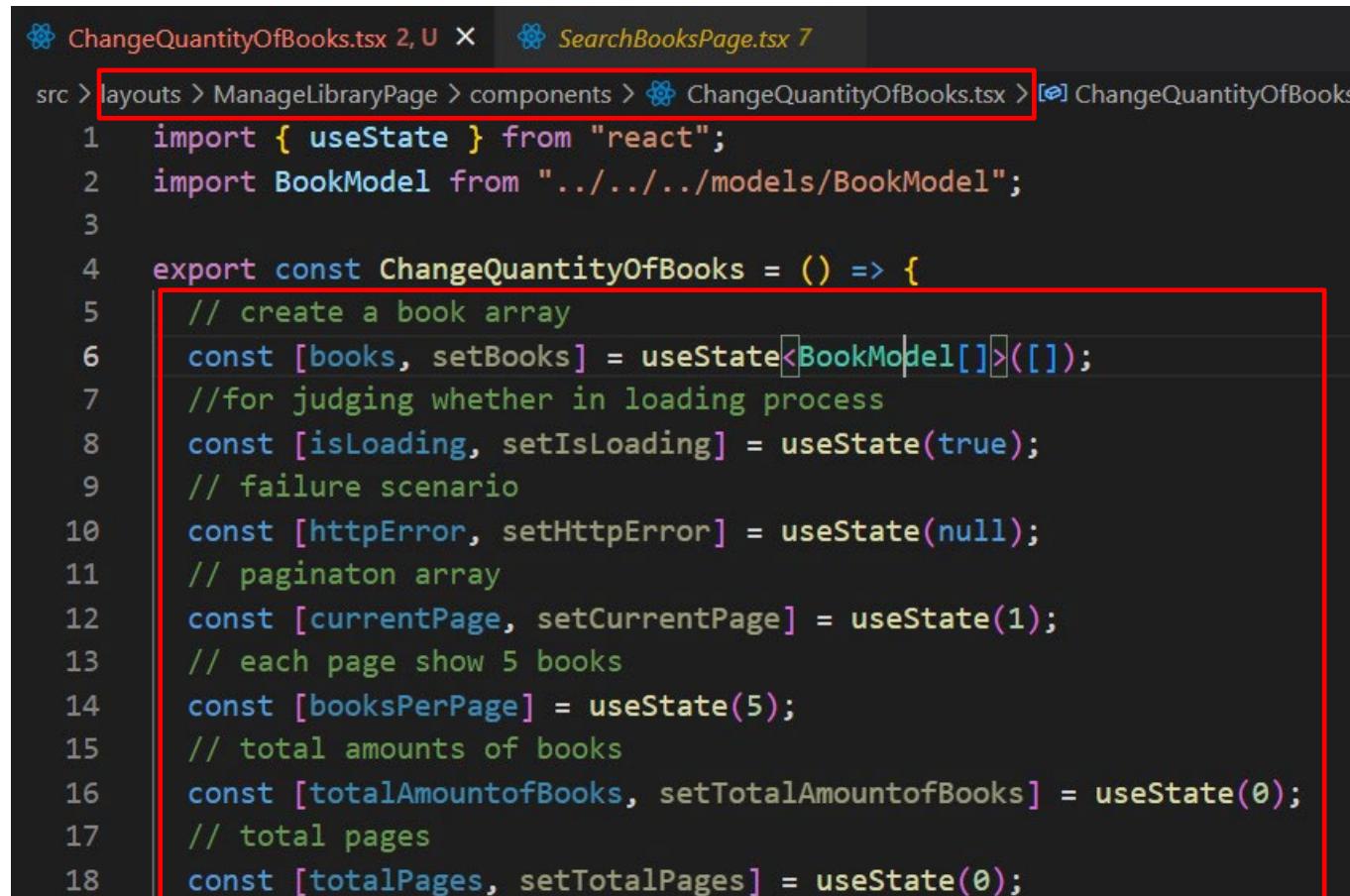
books, isLoading, httpError, currentPage, BooksPerPage, TotalAmountOfBooks and totalPages.



The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: REACT-LIBRARY, showing node\_modules, public, src (Auth, Images, layouts/BookCheckoutPage, layouts/HomePage/components/HomePage.tsx, ManageLibraryPage/components/AddNewBook.tsx, AdminMessage.tsx, AdminMessages.tsx, ChangeQuantityOfBooks.tsx, ManageLibraryPage.tsx).
- CODE EDITOR**: File: SearchBooksPage.tsx (7 lines)
- File Path**: src > layouts > SearchBooksPage > SearchBooksPage.tsx
- Code Content (highlighted by a red box):**

```
1 import { useEffect, useState } from "react";
2 import BookModel from "../../models/BookModel";
3 import { Pagination } from "../../Utils/Pagination";
4 import { SpinnerLoading } from "../../Utils/SpinnerLoading";
5 import { SearchBook } from "./components/SearchBook";
6
7 export const SearchBooksPage = () => {
8     // create a book array
9     const [books, setBooks] = useState<BookModel[]>([]);
10    //for judging whether in loading process
11    const [isLoading, setIsLoading] = useState(true);
12    // failure scenario
13    const [httpError, setHttpError] = useState(null);
14    // paginaton array
15    const [currentPage, setCurrentPage] = useState(1);
16    // each page show 5 books
17    const [booksPerPage] = useState(5);
```



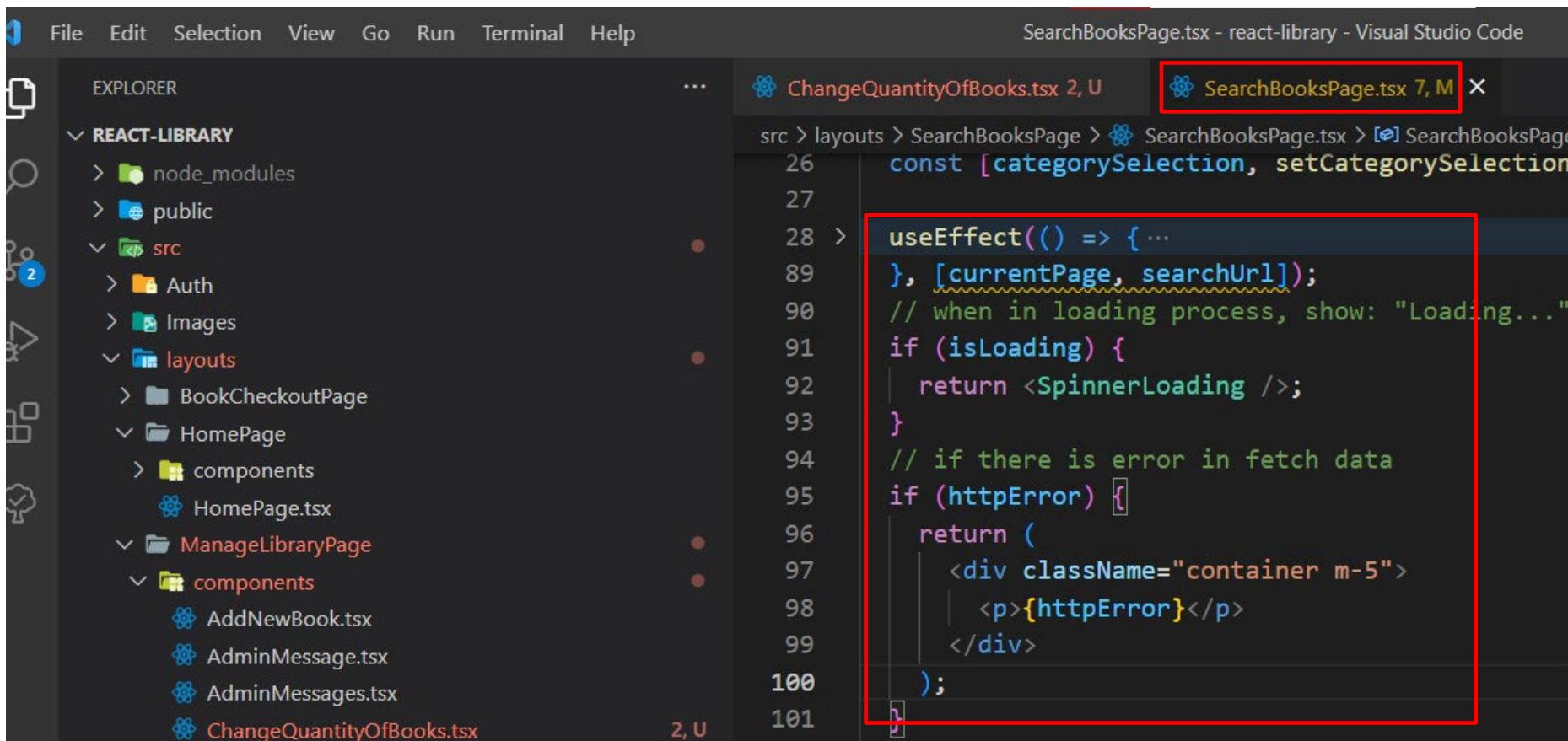
```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > [e] ChangeQuantityOfBooks
1 import { useState } from "react";
2 import BookModel from "../../../../../models/BookModel";
3
4 export const ChangeQuantityOfBooks = () => {
5   // create a book array
6   const [books, setBooks] = useState<BookModel[]>([[]]);
7   //for judging whether in loading process
8   const [isLoading, setIsLoading] = useState(true);
9   // failure scenario
10  const [httpError, setHttpError] = useState(null);
11  // paginaton array
12  const [currentPage, setCurrentPage] = useState(1);
13  // each page show 5 books
14  const [booksPerPage] = useState(5);
15  // total amounts of books
16  const [totalAmountofBooks, setTotalAmountofBooks] = useState(0);
17  // total pages
18  const [totalPages, setTotalPages] = useState(0);
```

### 3.2

Go back to our ChangeQuantityOfBooks and paste this in.  
Then don't forget to import useState from react.

## Step 4 Create useEffect.

4.1 Go back to SearchBooksPage.tsx, grab this entire useEffect so the fetchBooks() to build a call and get all of our books, also the httpError, and isLoading part.



The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help
- Explorer:** Shows the project structure under "REACT-LIBRARY".
- Search Books Page:** The active file is "SearchBooksPage.tsx" located at "src > layouts > SearchBooksPage".
- Code Editor:** The code for "useEffect" is highlighted with a red box:

```
const [categorySelection, setCategorySelection] = useState("All");

useEffect(() => {
  const [currentPage, searchUrl] = useState([1, ""]);
  // when in loading process, show: "Loading..."
  if (isLoading) {
    return <SpinnerLoading />;
  }
  // if there is error in fetch data
  if (httpError) {
    return (
      <div className="container m-5">
        <p>{httpError}</p>
      </div>
    );
  }
  // Fetch books
  fetchBooks();
}, [categorySelection, currentPage, searchUrl]);
```

4.2 Go back to our ChangeQuantityOfBooks component and right under our pieces of state, paste them.

Also, don't forget to import useEffect from react.

The screenshot shows a code editor with a sidebar on the left displaying a file tree. The file tree includes:

- BookCheckoutPage
- HomePage
- ManageLibraryPage
  - components
    - HomePage.tsx
  - ChangeQuantityOfBooks.tsx (highlighted with a red box)
  - ManageLibraryPage.tsx
- MessagesPage
- NavbarAndFooter
- SearchBooksPage
  - components
    - SearchBook.tsx
  - SearchBooksPage.tsx

The main editor area shows the content of ChangeQuantityOfBooks.tsx:

```
// total pages
const [totalPages, setTotalPages] = useState(0);

useEffect(() => {
}, [currentPage, searchUrl]);
// when in loading process, show: "Loading..."
if (isLoading) {
  return <SpinnerLoading />;
}
// if there is error in fetch data
if (httpError) {
  return (
    <div className="container m-5">
      <p>{httpError}</p>
    </div>
  );
}
```

## 4.3 fix url.

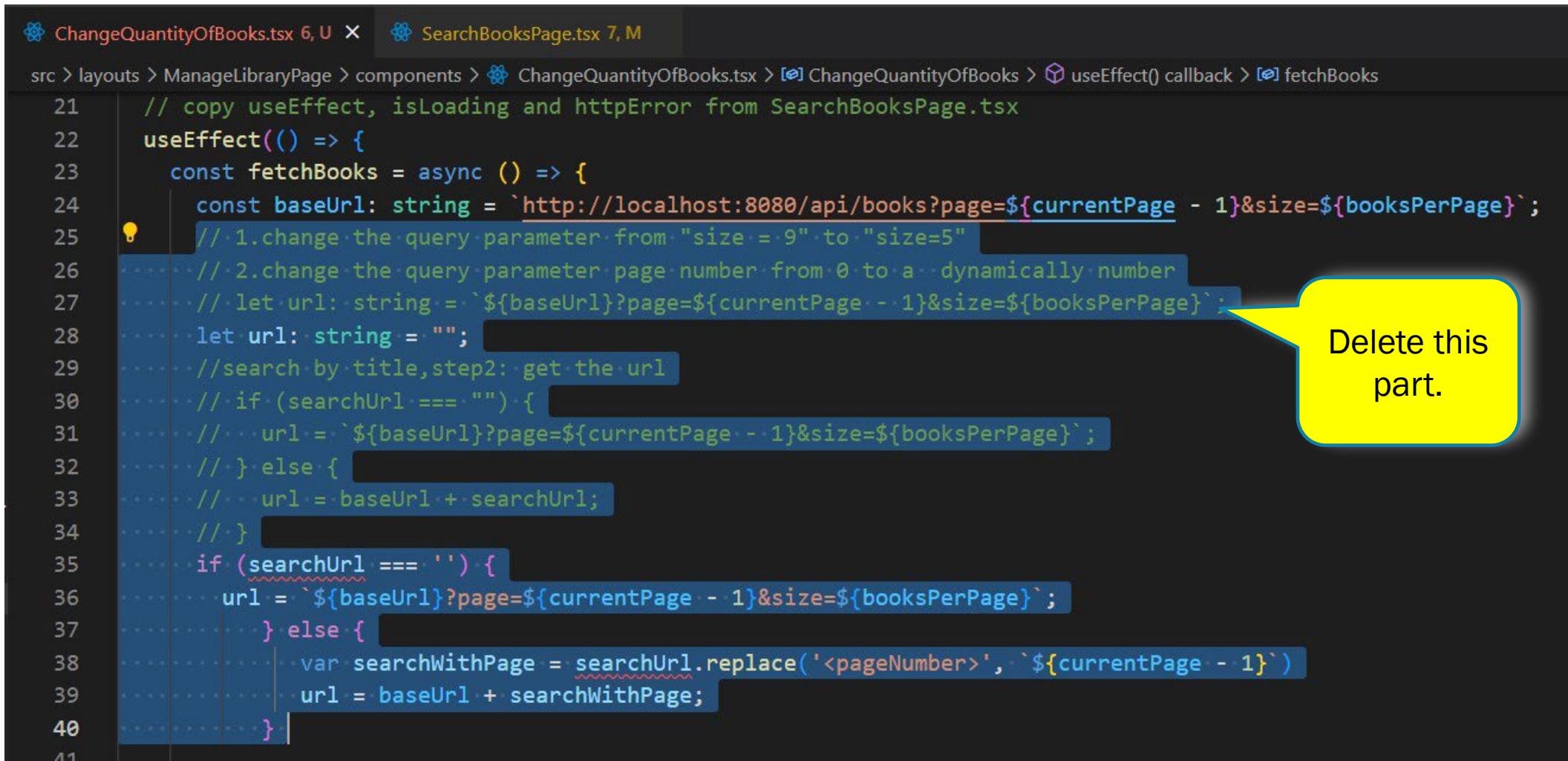
Change double quotation into backticks and add “`page=${currentPage - 1}&size=${booksPerPage}`” to the end of url string.

```
useEffect(() => {
  const fetchBooks = async () => {
    const baseUrl: string = "http://localhost:8080/api/books";
    // 1.change the query parameter from "size = 9" to "size=5"
    // ...
  }
  // ...
}

const [totalPages, setTotalPages] = useState(0);

// copy useEffect, isLoading and httpError from SearchBooksPage.tsx
useEffect(() => {
  const fetchBooks = async () => {
    const baseUrl: string = `http://localhost:8080/api/books?page=${currentPage - 1}&size=${booksPerPage}`;
```

## 4.4 Delete “let url” and “if” statement.



ChangeQuantityOfBooks.tsx 6, U X SearchBooksPage.tsx 7, M

src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > useEffect() callback > fetchBooks

```
21 // copy useEffect, isLoading and httpError from SearchBooksPage.tsx
22 useEffect(() => {
23   const fetchBooks = async () => {
24     const baseUrl: string = `http://localhost:8080/api/books?page=${currentPage - 1}&size=${booksPerPage}`;
25     // 1.change the query parameter from "size = 9" to "size=5"
26     // 2.change the query parameter page number from 0 to a dynamically number
27     // let url: string = `${baseUrl}?page=${currentPage - 1}&size=${booksPerPage}`;
28     let url: string = "";
29     //search by title,step2: get the url
30     // if (searchUrl === "") {
31     //   url = `${baseUrl}?page=${currentPage - 1}&size=${booksPerPage}`;
32     // } else {
33     //   url = baseUrl + searchUrl;
34     // }
35     if (searchUrl === '') {
36       url = `${baseUrl}?page=${currentPage - 1}&size=${booksPerPage}`;
37     } else {
38       var searchWithPage = searchUrl.replace('<pageNumber>', `${currentPage - 1}`);
39       url = baseUrl + searchWithPage;
40     }
41 }
```

Delete this part.

#### 4.5 change the parameter of await fetch from “url” into “baseURL”.

```
ChangeQuantityOfBooks.tsx 5, U X SearchBooksPage.tsx 7, M  
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx >  
21 // copy useEffect, isLoading and httpError from Se  
22 useEffect(() => {  
23   const fetchBooks = async () => {  
24     const baseUrl: string = `http://localhost:8086`  
25     // fetching the url data  
26     const response = await fetch(url);  
27     //failure scenario
```



```
ChangeQuantityOfBooks.tsx 4, U X SearchBooksPage.tsx 7, M  
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks  
21 // copy useEffect, isLoading and httpError fr  
22 useEffect(() => {  
23   const fetchBooks = async () => {  
24     const baseUrl: string = `http://localhost  
25     // fetching the url data  
26     const response = await fetch(baseUrl);
```

4.6 delete window.scrollTo(0, 0);

```
};  
// if async has an error  
fetchBooks().catch(error: any) => {  
    setIsLoading(false);  
    setHttpError(error.message);  
});  
//each time turn to a new page, it will go to the top of the page  
window.scrollTo(0, 0);  
// pass current page so each time we click on different page number
```

Delete this part.

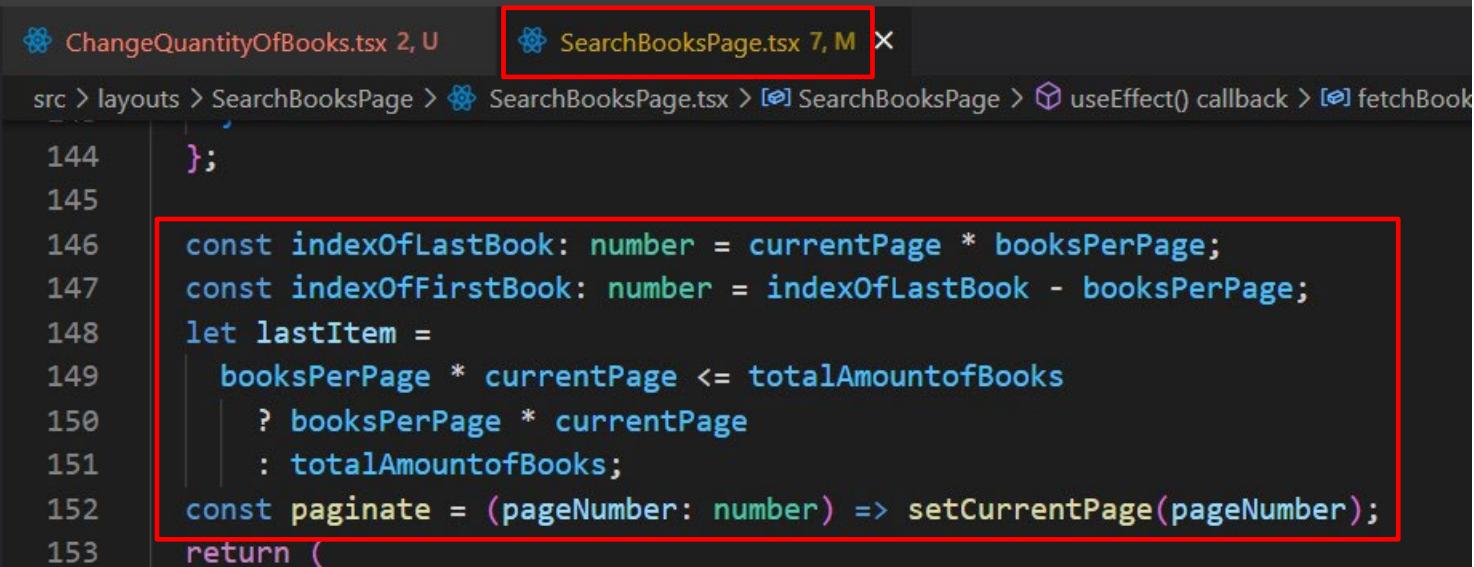
4.7 delete “searchUrl” as a state change factor in the array at the end of the useEffect.

```
// pass current page , so each time we click on different page number  
//(S13-95,search by title)step3: add searchUrl parameter  
, [currentPage, searchUrl]);  
// when in loading process. show: "Loading..."
```

Delete this  
part.

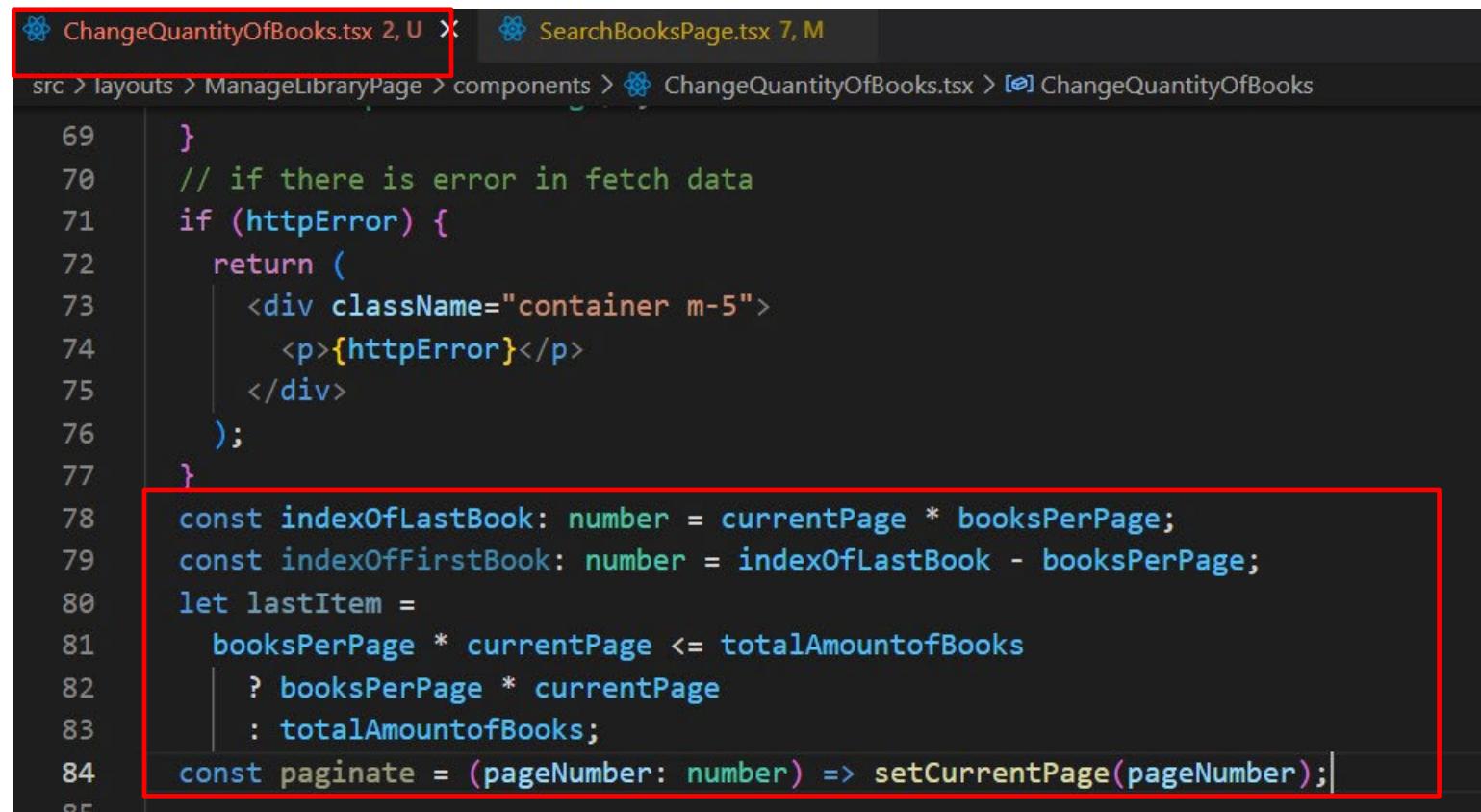
## Step 5 create pagination functionality.

5.1 Let's go back to our SearchBooksPage.tsx, scroll all the way to the bottom until we can see this “const indexOfLastBook,const indexOfFirstBook, const paginate”, grab all of those functions.



```
ChangeQuantityOfBooks.tsx 2, U SearchBooksPage.tsx 7, M
src > layouts > SearchBooksPage > SearchBooksPage.tsx > [ ] SearchBooksPage > [ ] useEffect() callback > [ ] fetchBook
144     };
145
146     const indexOfLastBook: number = currentPage * booksPerPage;
147     const indexOfFirstBook: number = indexOfLastBook - booksPerPage;
148     let lastItem =
149       booksPerPage * currentPage <= totalAmountofBooks
150         ? booksPerPage * currentPage
151         : totalAmountofBooks;
152     const paginate = (pageNumber: number) => setCurrentPage(pageNumber);
153   return (
```

5.2 Go back up to our ChangeQuantityOfBooks.tsx , and paste the code in, under the httpError part.



```
ChangeQuantityOfBooks.tsx 2, U > SearchBooksPage.tsx 7, M
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > [o] ChangeQuantityOfBooks
69 }
70 // if there is error in fetch data
71 if (httpError) {
72     return (
73         <div className="container m-5">
74             <p>{httpError}</p>
75         </div>
76     );
77 }
78 const indexOfLastBook: number = currentPage * booksPerPage;
79 const indexOfFirstBook: number = indexOfLastBook - booksPerPage;
80 let lastItem =
81     booksPerPage * currentPage <= totalAmountofBooks
82     ? booksPerPage * currentPage
83     : totalAmountofBooks;
84 const paginate = (pageNumber: number) => setCurrentPage(pageNumber);
```

Step 6 Fill in return part with HTML/CSS code.

```
return(
  <div className='container mt-5'>
    /* judge if the total amount of books is positive, which means the book exists */
    {totalAmountofBooks > 0 ?
      <>
        <div className='mt-3'>
          <h3>Number of results: ({totalAmountofBooks})</h3>
        </div>
        <p>
          {indexOfFirstBook + 1} to {lastItem} of {totalAmountofBooks} items:
        </p>
        {books.map(book => (
          <p>display different quantity of books </p>
        )));
      </>
      :
      <h5>Add a book before changing quantity</h5>
    }
    {totalPages > 1 && <Pagination currentPage={currentPage} totalPages={totalPages} paginate={paginate}>}
  </div>
)
```

Number of results: (23)

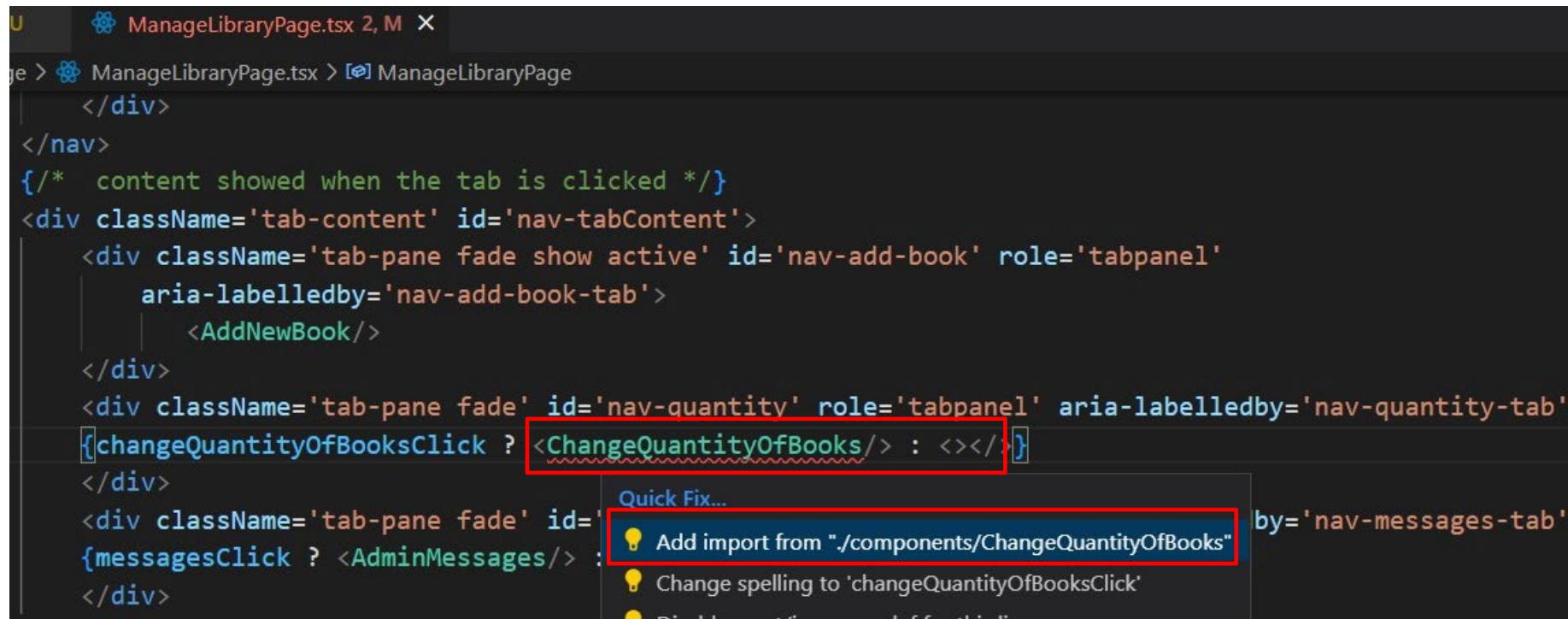
1 to 5 of 23 items:

display different quantity of books

First Page 1 2 3 Last Page

Step 7. Go to ManageLibraryPage.tsx, change static words “change quantity” into our new ChangeQuantityOfBooks component. And import this component to the file.

```
|     |     <AddNewBook/>
|     |   </div>
|     <div className='tab-pane fade' id='nav-quantity' role='tabpanel' aria-labelledby='nav-quantity-tab'
|       {changeQuantityOfBooksClick ? <>Change Quantity Of Books</> : <></>}
|     </div>
```



The screenshot shows the ManageLibraryPage.tsx file in a code editor. A red box highlights the line of code where 'Change Quantity Of Books' is written. A second red box highlights the 'ChangeQuantityOfBooks' component name in the 'Quick Fix...' dropdown menu.

```
U ManageLibraryPage.tsx 2, M X
je > ManageLibraryPage.tsx > ManageLibraryPage
    </div>
</nav>
/* content showed when the tab is clicked */
<div className='tab-content' id='nav-tabContent'>
  <div className='tab-pane fade show active' id='nav-add-book' role='tabpanel'
    aria-labelledby='nav-add-book-tab'>
    <AddNewBook/>
  </div>
  <div className='tab-pane fade' id='nav-quantity' role='tabpanel' aria-labelledby='nav-quantity-tab'>
    {changeQuantityOfBooksClick ? <ChangeQuantityOfBooks/> : <></>}
  </div>
  <div className='tab-pane fade' id='nav-messages' role='tabpanel' aria-labelledby='nav-messages-tab'>
    {messagesClick ? <AdminMessages/> : <></>}
  </div>
```



## Manage Library

Add new book

Change quantity

Messages

Number of results: (23)

1 to 5 of 23 items:

display different quantity of books

First Page 1 2 3 Last Page



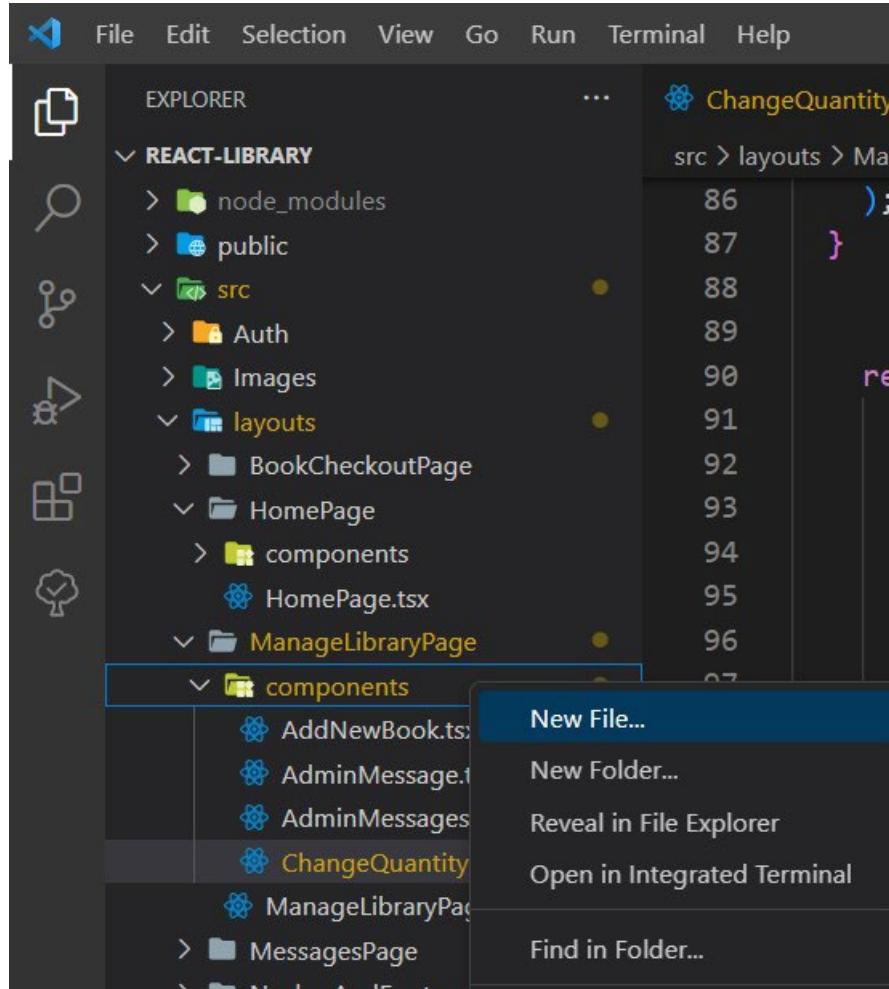
Now, with admin user logged in status, visit <http://localhost:3000/admin>, Click on the “change quantity” tab, we can get this result:

---

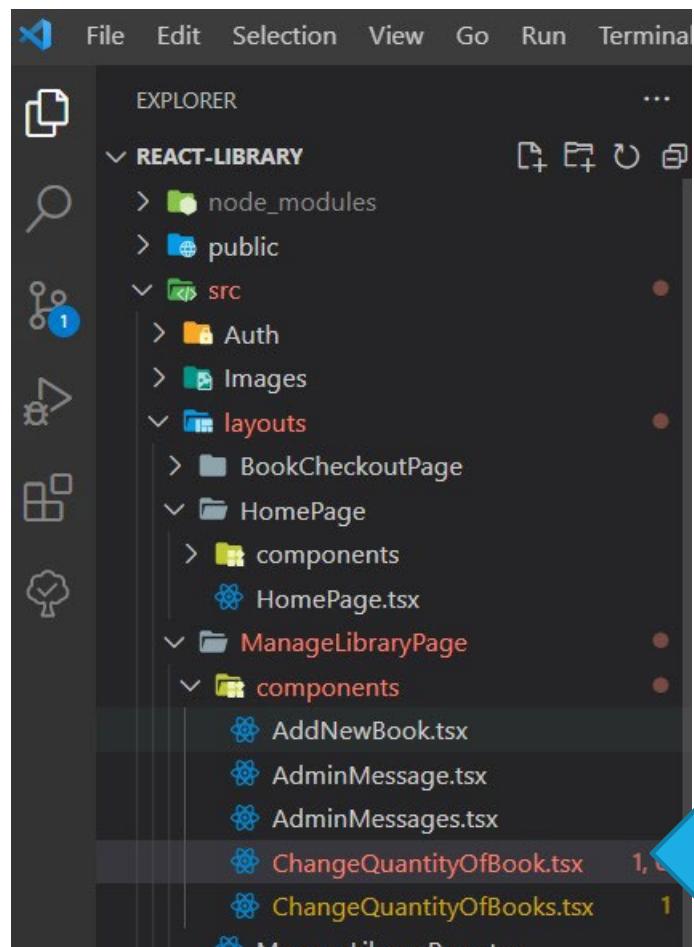
---

---

## **2. REACT-CHANGE QUANTITY OF BOOK**

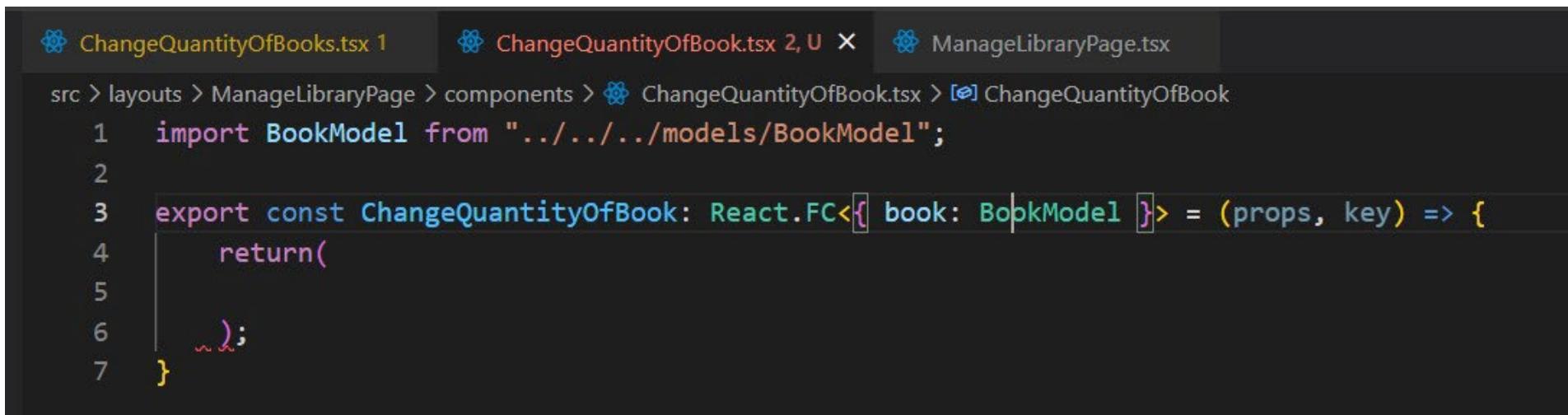


Step1 ,  
Go to our ManageLibraryPage folder, go inside our  
components folder, right click new file.



Name it ChangeQuantityOfBook.tsx.

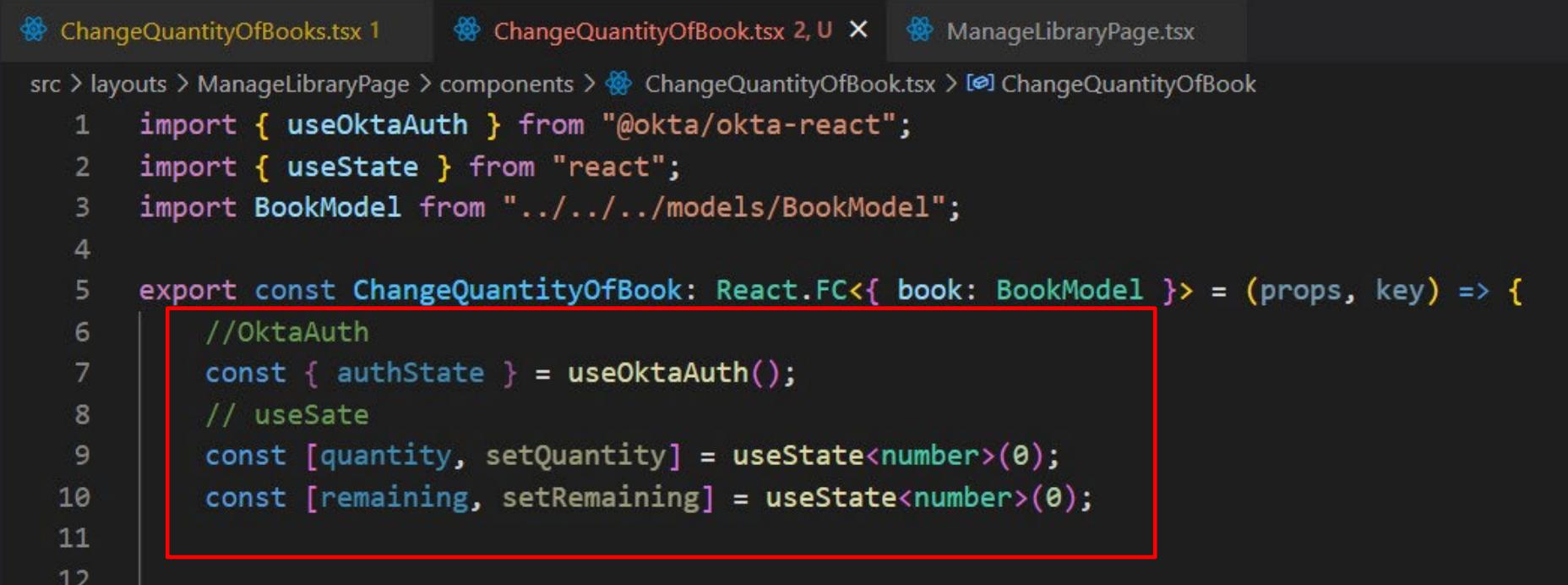
Step 2, create export const ChangeQuantityOfBook stucture.



The screenshot shows a code editor with three tabs at the top: 'ChangeQuantityOfBooks.tsx 1', 'ChangeQuantityOfBook.tsx 2, U X', and 'ManageLibraryPage.tsx'. The current tab is 'ChangeQuantityOfBook.tsx 1'. The code editor displays the following TypeScript code:

```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx > [?] ChangeQuantityOfBook
1 import BookModel from "../../models/BookModel";
2
3 export const ChangeQuantityOfBook: React.FC<{ book: BookModel }> = (props, key) => {
4     return(
5
6     );
7 }
```

Step 3 create useState, useOktaAuth , and import them.



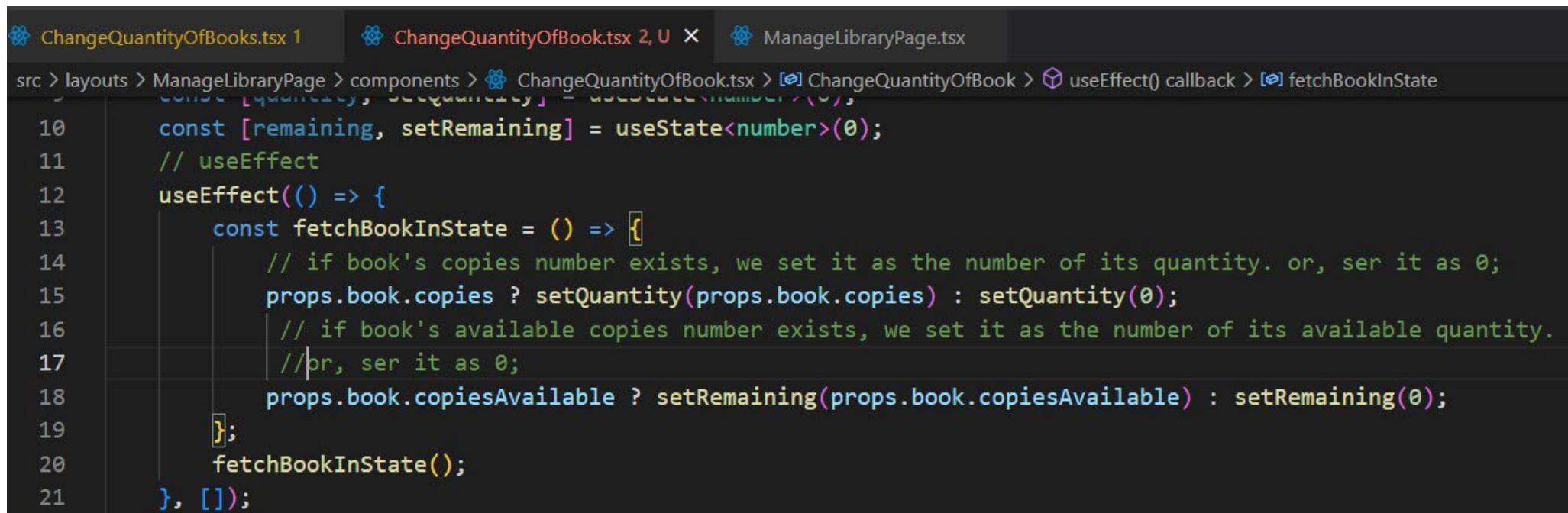
The screenshot shows a code editor with three tabs at the top: "ChangeQuantityOfBooks.tsx 1", "ChangeQuantityOfBook.tsx 2, U X", and "ManageLibraryPage.tsx". The "ChangeQuantityOfBook.tsx 2, U X" tab is active. The code editor displays the following TypeScript code:

```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx > ChangeQuantityOfBook
1 import { useOktaAuth } from "@okta(okta-react";
2 import { useState } from "react";
3 import BookModel from "../../../../../models/BookModel";
4
5 export const ChangeQuantityOfBook: React.FC<{ book: BookModel }> = (props, key) => {
6   //OktaAuth
7   const { authState } = useOktaAuth();
8   // useState
9   const [quantity, setQuantity] = useState<number>(0);
10  const [remaining, setRemaining] = useState<number>(0);
11
12
```

A red rectangular box highlights the imports for `useOktaAuth` and `useState` in lines 1 and 2 respectively, as instructed in the step.

## Step 4 create useEffect.

Inside useEffect, we need an async function fetchBookInState() to set the number of total copies and available copies.



```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx > [e] ChangeQuantityOfBook > [i] useEffect() callback > [e] fetchBookInState
  const [remaining, setRemaining] = useState<number>(0);
  // useEffect
  useEffect(() => {
    const fetchBookInState = () => {
      // if book's copies number exists, we set it as the number of its quantity. or, ser it as 0;
      props.book.copies ? setQuantity(props.book.copies) : setQuantity(0);
      // if book's available copies number exists, we set it as the number of its available quantity.
      //pr, ser it as 0;
      props.book.copiesAvailable ? setRemaining(props.book.copiesAvailable) : setRemaining(0);
    };
    fetchBookInState();
  }, []);
```

---

---

### **3. REACT-CHANGE QUANTITY OF BOOK HTML/CSS**



```

<div className='card mt-3 shadow p-3 mb-3 bg-body rounded'>
<div className='row g-0'>
  <div className='col-md-2'>
    /* image part, if exists, show its picture, if not, show default image */
    <div className='d-none d-lg-block'>
      {props.book.img ?
        <img src={props.book.img} width='123' height='196' alt='Book' />
        :
        <img src={require('../../../../../Images/BooksImages/book-luv2code-1000.png')}>
          width='123' height='196' alt='Book' />
      }
    </div>
    /* mobile version */
    <div className='d-lg-none d-flex justify-content-center align-items-center'>
      {props.book.img ?
        <img src={props.book.img} width='123' height='196' alt='Book' />
        :
        <img src={require('../../../../../Images/BooksImages/book-luv2code-1000.png')}>
          width='123' height='196' alt='Book' />
      }
    </div>
  </div>
</div>

```



Step 1,  
image part.  
(desktop version and mobile  
version).  
The two parts of codes are the  
same except the className of the  
div.  
For the mobile version, we set it “  
d-flex justify-content-center align-  
items-center”.

Luv, Zofia

**JavaScript Cookbook**

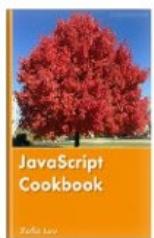
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin  
 risus tortor, condimentum eget sapien ac, dapibus varius ligula.  
 Maecenas justo erat, semper sed nunc vel, vulputate eleifend  
 dui. Integer id incus vitae nisi malesuada feugiat. Praesent sit amet

Total Quantity: **10**

Books Remaining: **10**

## Step 2, book info part.

```
❶ ChangeQuantityOfBooks.tsx 1   ❷ ChangeQuantityOfBook.tsx 2, U X ❸ ManageLibraryPage.tsx
src > layouts > ManageLibraryPage > components > ❷ ChangeQuantityOfBook > ❸ ChangeQuantityOfBook
45           </div>
46           </div>
47           <div className='col-md-6'>
48             <div className='card-body'>
49               <h5 className='card-title'>{props.book.author}</h5>
50               <h4>{props.book.title}</h4>
51               <p className='card-text'> {props.book.description} </p>
52           </div>
53       </div>
```



Luv, Zofia

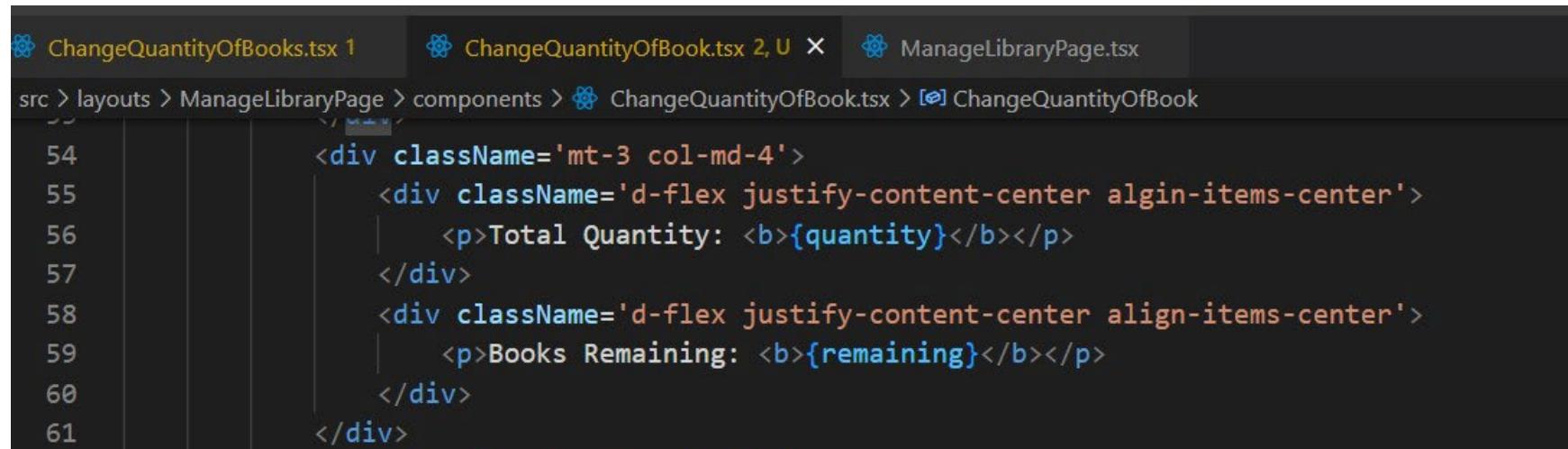
**JavaScript Cookbook**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin  
risus tortor, condimentum eget sapien ac, dapibus varius ligula.  
Maecenas justo erat, semper sed nunc vel, vulputate eleifend  
dui. Integer id ipsum vitae nisi malesuada feugiat. Proin sit amet

Total Quantity: **10**

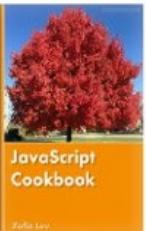
Books Remaining: **10**

## Step 3, quantity part.



```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx > ChangeQuantityOfBook
54     <div className='mt-3 col-md-4'>
55         <div className='d-flex justify-content-center align-items-center'>
56             <p>Total Quantity: <b>{quantity}</b></p>
57         </div>
58         <div className='d-flex justify-content-center align-items-center'>
59             <p>Books Remaining: <b>{remaining}</b></p>
60         </div>
61     </div>
```





Luv, Zofia

**JavaScript Cookbook**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin  
risus tortor, condimentum eget sapien ac, dapibus varius ligula.  
Maecenas justo erat, semper sed nunc vel, vulputate eleifend  
dui. Integer id insum vitae nisi malesuada feugiat. Proin sit amet

Total Quantity: **10**

Books Remaining: **10**

```
    ...
    <div className='mt-3 col-md-1'>
      <div className='d-flex justify-content-start'>
        <button className='m-1 btn btn-md btn-danger' >Delete</button>
      </div>
      <button className='m1 btn btn-md main-color text-white' >Add Quantity</button>
      <button className='m1 btn btn-md btn-warning' >Decrease Quantity</button>
    </div>
  </div>
```

The screenshot shows a card for a book titled "JavaScript Cookbook". The card includes a thumbnail image, the book title, a short description, and a "Books Remaining: 10" indicator. At the bottom of the card, there are three buttons: a red "Delete" button, a blue "Add Quantity" button, and a yellow "Decrease Quantity" button. A large blue arrow points from the explanatory text above down to the "Add Quantity" and "Decrease Quantity" buttons.

**JavaScript Cookbook**

Books Remaining: **10**

Delete

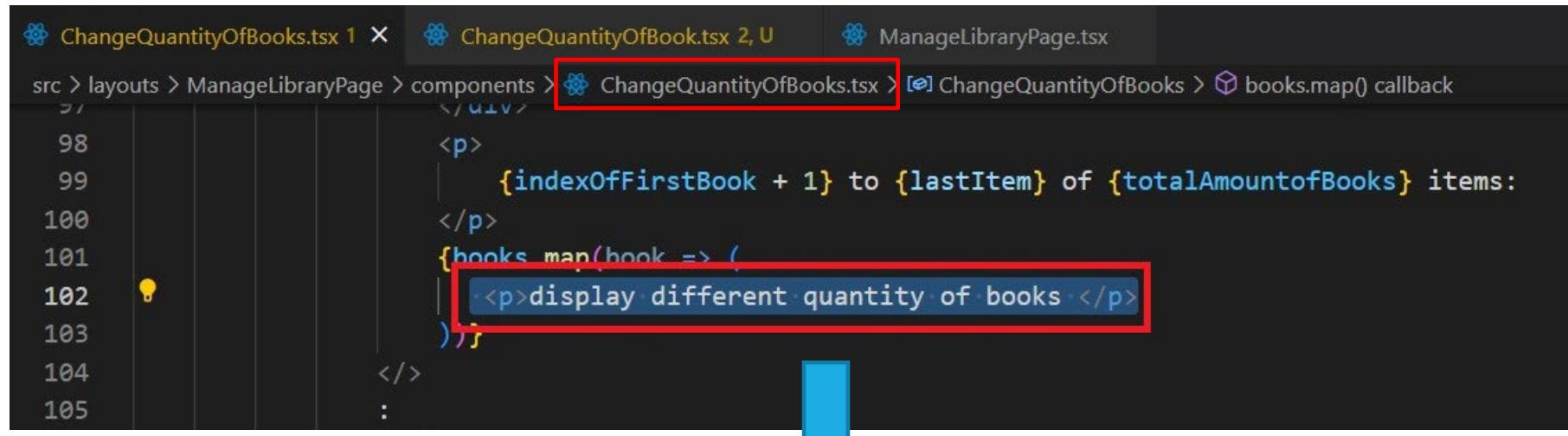
Add Quantity

Decrease Quantity

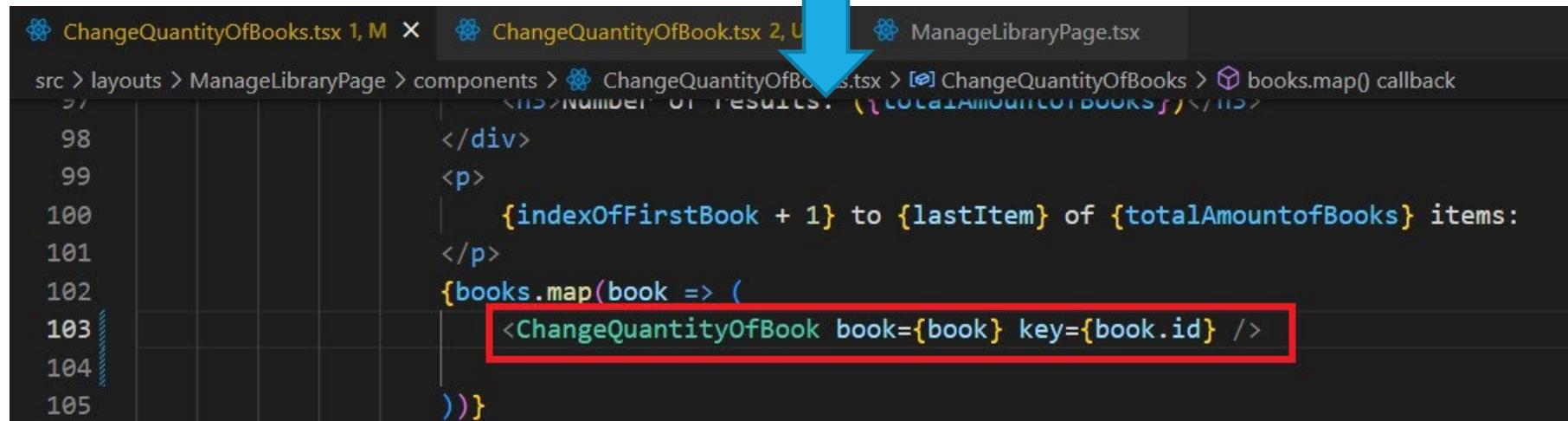
**Step 4 , “Add Quantity” and “Decrease Quantity” button.**

Step 5. Let's jump back into our ChangeQuantityOfBooks.tsx. Scroll down until we can see display different quantity of books.

Replace them with our ChangeQuantityOfBook component.

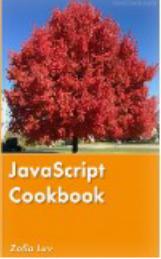


```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > ChangeQuantityOfBooks > books.map() callback
  ...
  98
  99      {indexOfFirstBook + 1} to {lastItem} of {totalAmountofBooks} items:
100
101    {books.map(book => (
102      <p>display different quantity of books</p>
103    )));
104
105  :
```



```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > ChangeQuantityOfBooks > books.map() callback
  ...
  98
  99      {indexOfFirstBook + 1} to {lastItem} of {totalAmountofBooks} items:
100
101    {books.map(book => (
102      <ChangeQuantityOfBook book={book} key={book.id} />
103    )));
104
105  :
```

## Our ChangeQuantityOfBook component:



Luv, Zofia

**JavaScript Cookbook**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin risus tortor, condimentum eget sapien ac, dapibus varius ligula. Maecenas justo erat, semper sed nunc vel, vulputate eleifend dui. Integer id ipsum vitae nisi malesuada feugiat. Proin sit amet quam laoreet, feugiat mi vitae, vestibulum dui. Aliquam erat volutpat. Etiam hendrerit erat nec mi auctor elementum. Curabitur vestibulum lectus a ante tempor tincidunt et sed orci. Proin maximus tortor in risus auctor efficitur. Phasellus quam mauris, laoreet et feugiat ac, imperdiet at quam. Nullam sollicitudin nec diam vel finibus.

Total Quantity: **10**

Books Remaining: **10**

**Delete**

Add Quantity

Decrease Quantity

React App    ×    +

localhost:3000/admin

Radio-Canada.ca |... Canada's federal... Sign in to your IR... Chenelière Éducat... Montreal zdm Nos incontournab... TED TED: Ideas worth... 在线课程 - 时

JAC Read Home Search Books Shelf

## Manage Library

Add new book Change quantity Messages

Number of results: (23)

1 to 5 of 23 items:

Luv, Zofia  
**JavaScript Cookbook**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin risus tortor, condimentum eget sapien ac, dapibus varius ligula. Maecenas justo erat, semper sed nunc vel, vulputate eleifend  
dui. Integer id ipsum vitae nisi malesuada feugiat. Proin sit amet quam laoreet, feugiat mi vitae, vestibulum  
dui. Aliquam erat volutpat. Etiam hendrerit erat nec mi auctor elementum. Curabitur vestibulum lectus a ante tempor tincidunt et sed orci. Proin maximus tortor in  
risus auctor efficitur. Phasellus quam mauris, laoreet et feugiat ac, imperdiet at quam. Nullam sollicitudin nec diam vel finibus.

Total Quantity: **10**  
Books Remaining: **10**

Delete

Add Quantity

Decrease Quantity

Now, with admin user logged in status, visit <http://localhost:3000/admin>, Click on the “change quantity” tab, we can get this result:

React App × + localhost:3000/admin

Radio-Canada.ca |... Canada's federal... Sign in to your IR... Chenelière Éducat... Montreal zdm Nos incontournab... TED TED: Ideas worth... 在线课

libero elementum non. Sed velit lacus, laoreet sed augue fermentum, sagittis convallis metus. Sed nec est at massa venenatis aliquet. Donec pretium interdum fringilla. Sed ornare tellus enim, a tincidunt libero dictum vitae. Proin bibendum posuere dui. Donec sagittis neque massa, sed semper nulla vehicula at.

Delete

Add Quantity

Decrease Quantity

New Author1

Advanced Tech in C#

great C# book.

Total Quantity: 6

Books Remaining: 5

Delete

Add Quantity

Decrease Quantity

First Page 3 4 5 Last Page

In the bottom of the page, there is a pagination.

---

---

## **4. SPRINGBOOT INCREASE BOOK QUANTITY SERVICE AND CONTROLLER LAYER**



The screenshot shows a Java IDE interface with the following details:

- Menu Bar:** File Refactor Build Run Tools Git Window Help
- Project Bar:** SpringBootLibraryApplication
- File List:** AdminService.java (highlighted with a red box), AdminController.java, SecurityConfiguration.java
- Code Area:** The code for AdminService.java is displayed. A specific method, `increaseBookQuantity(Long bookId)`, is highlighted with a red box.

```
    this.bookRepository = bookRepository;
}
//increase the number of the book
new *
public void increaseBookQuantity(Long bookId) throws Exception {
    //save the book which we found in our database by passing bookId.
    Optional<Book> book = bookRepository.findById(bookId);
    //if the book does not exist
    if (!book.isPresent()) {
        throw new Exception("Book not found");
    }
    // if exists, increase the number of copies, available copies by 1
    book.get().setCopiesAvailable(book.get().getCopiesAvailable() + 1);
    book.get().setCopies(book.get().getCopies() + 1);
    // save the book to the database.
    bookRepository.save(book.get());
}
```

we are going to add the API endpoint that will allow an admin user to be able to increase how many copies are available for a specific book.

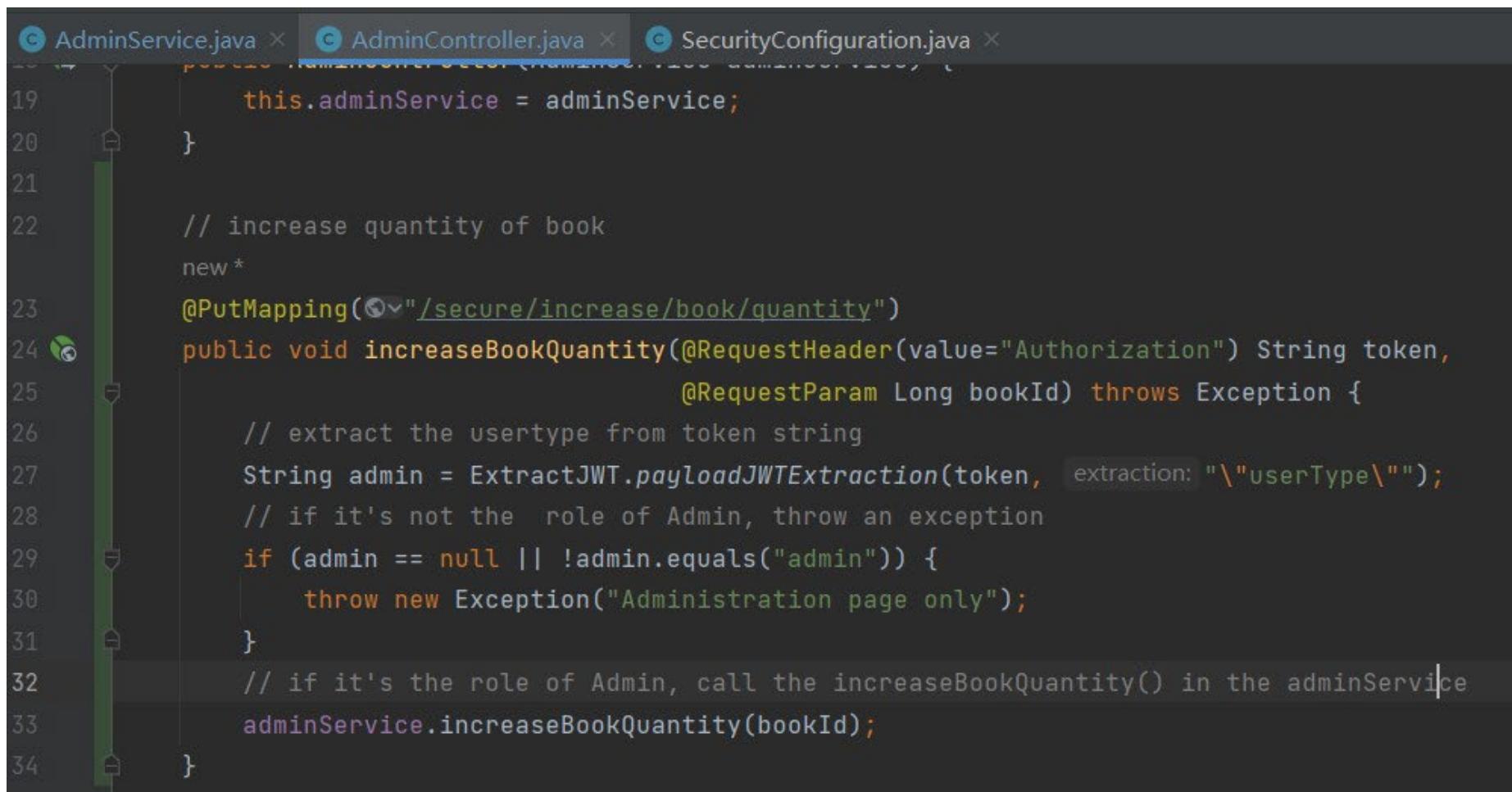
## Step 1

So let's open “service” directory and jump into our AdminService file.

Now right above our `postbook()`, create a new function `increaseBookQuantity()`.

Step 2 Go to controller package, open AdminController file, use “PUT” method to create API for increasing book's quantity.

```
@PutMapping("/secure/increase/book/quantity")
```



```
AdminController.java
19     this.adminService = adminService;
20 }
21
22 // increase quantity of book
23 new *
24 @PutMapping("/secure/increase/book/quantity")
25 public void increaseBookQuantity(@RequestHeader(value="Authorization") String token,
26                                     @RequestParam Long bookId) throws Exception {
27     // extract the usertype from token string
28     String admin = ExtractJWT.payloadJWTExtraction(token, extraction: "\"userType\"");
29     // if it's not the role of Admin, throw an exception
30     if (admin == null || !admin.equals("admin")) {
31         throw new Exception("Administration page only");
32     }
33     // if it's the role of Admin, call the increaseBookQuantity() in the adminService
34     adminService.increaseBookQuantity(bookId);
35 }
```

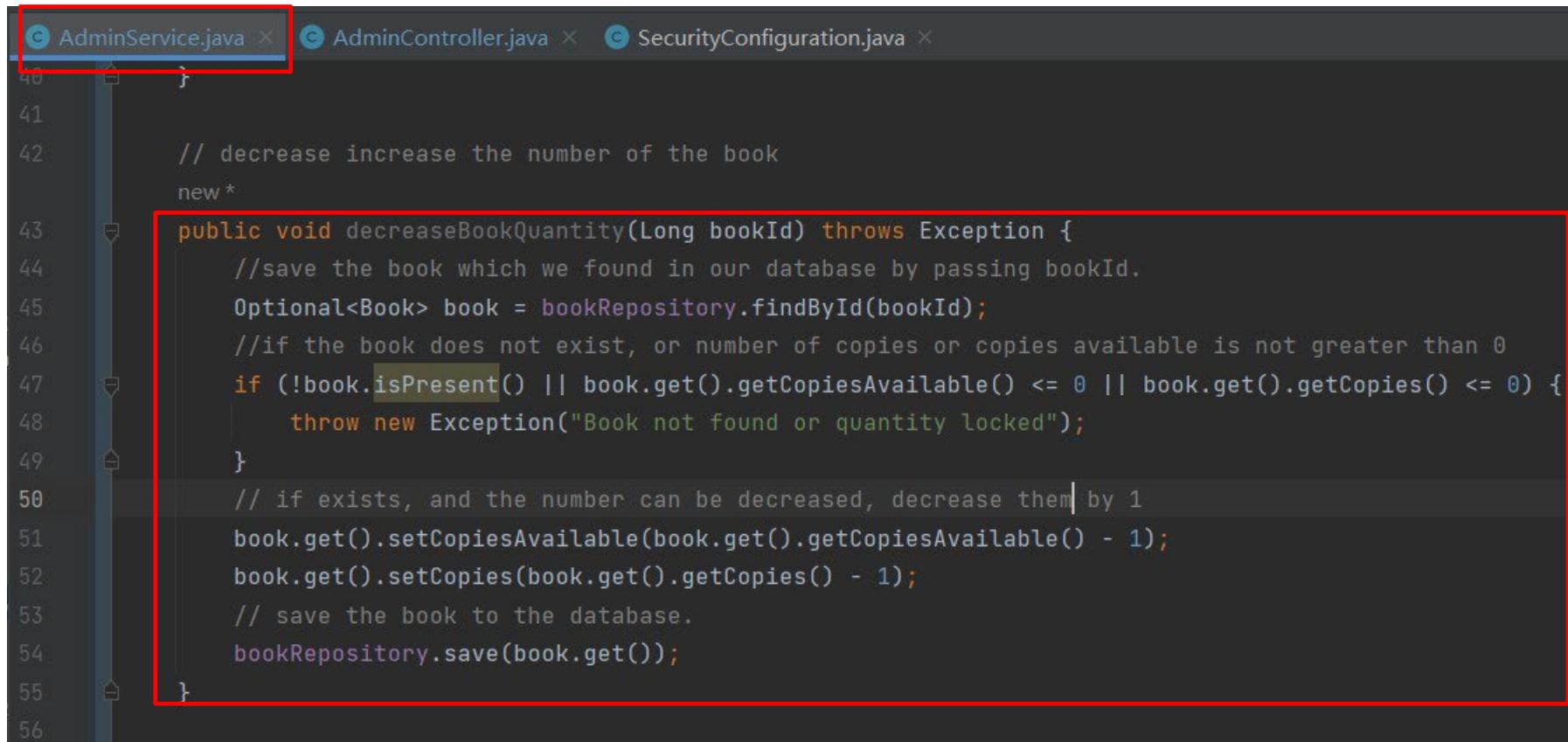
---

---

## **5. SPRINGBOOT DECREASE BOOK QUANTITY SERVICE AND CONTROLLER LAYER**



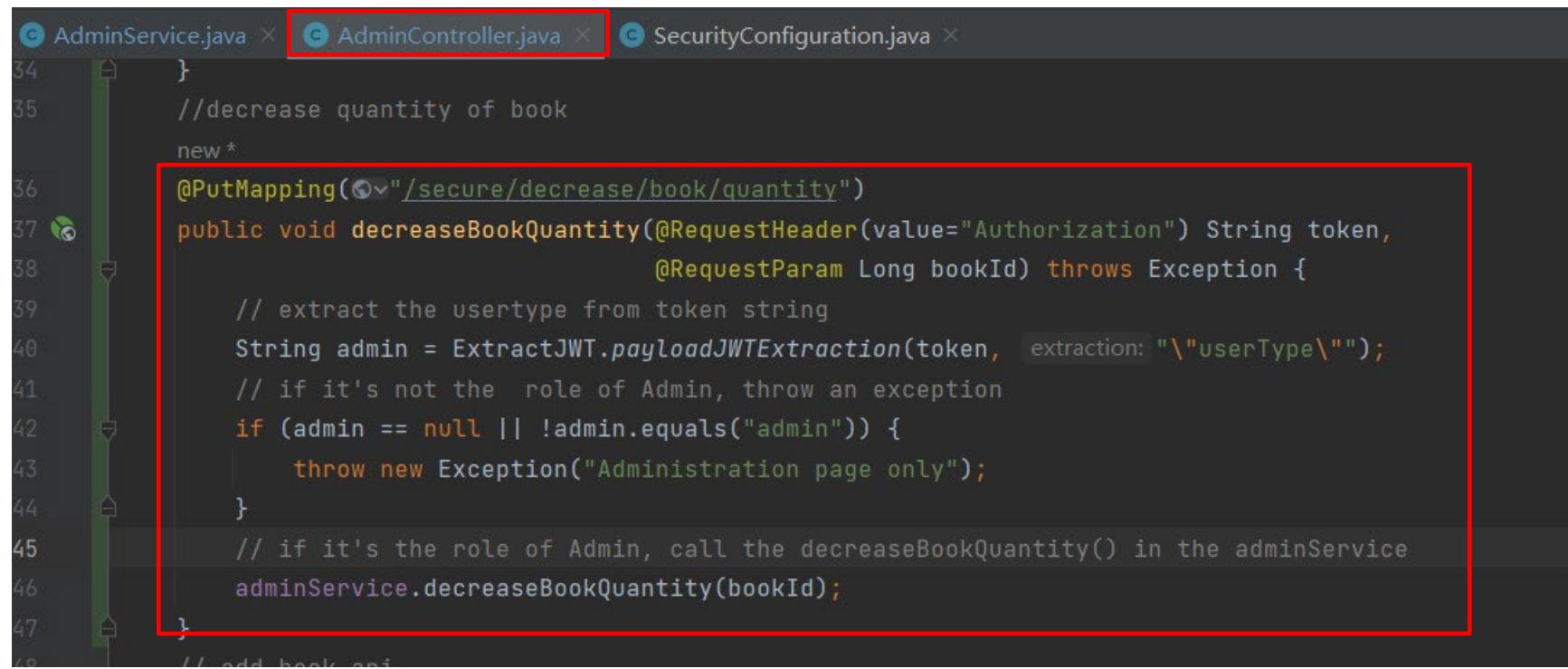
Step 1, So let's open up service directory and jump into our AdminService file.  
Create a new function decreaseBookQuantity().



The screenshot shows a code editor with three tabs at the top: AdminService.java, AdminController.java, and SecurityConfiguration.java. The AdminService.java tab is active and highlighted with a red box. The code in AdminService.java defines a method named decreaseBookQuantity. This method first checks if a book with the given ID exists and if its available copies are greater than zero. If both conditions are met, it decreases the available copies by one and saves the updated book back to the database. The entire body of the decreaseBookQuantity method is also highlighted with a red box.

```
40
41
42     // decrease increase the number of the book
43     new *
44
45     public void decreaseBookQuantity(Long bookId) throws Exception {
46         //save the book which we found in our database by passing bookId.
47         Optional<Book> book = bookRepository.findById(bookId);
48         //if the book does not exist, or number of copies or copies available is not greater than 0
49         if (!book.isPresent() || book.get().getCopiesAvailable() <= 0 || book.get().getCopies() <= 0) {
50             throw new Exception("Book not found or quantity locked");
51         }
52         // if exists, and the number can be decreased, decrease them by 1
53         book.get().setCopiesAvailable(book.get().getCopiesAvailable() - 1);
54         book.get().setCopies(book.get().getCopies() - 1);
55         // save the book to the database.
56         bookRepository.save(book.get());
57     }
58 }
```

Step 2 Let's jump into "controller" package, "AdminController" file, use "PUT" method to create endpoint for decreasing the book quantity.

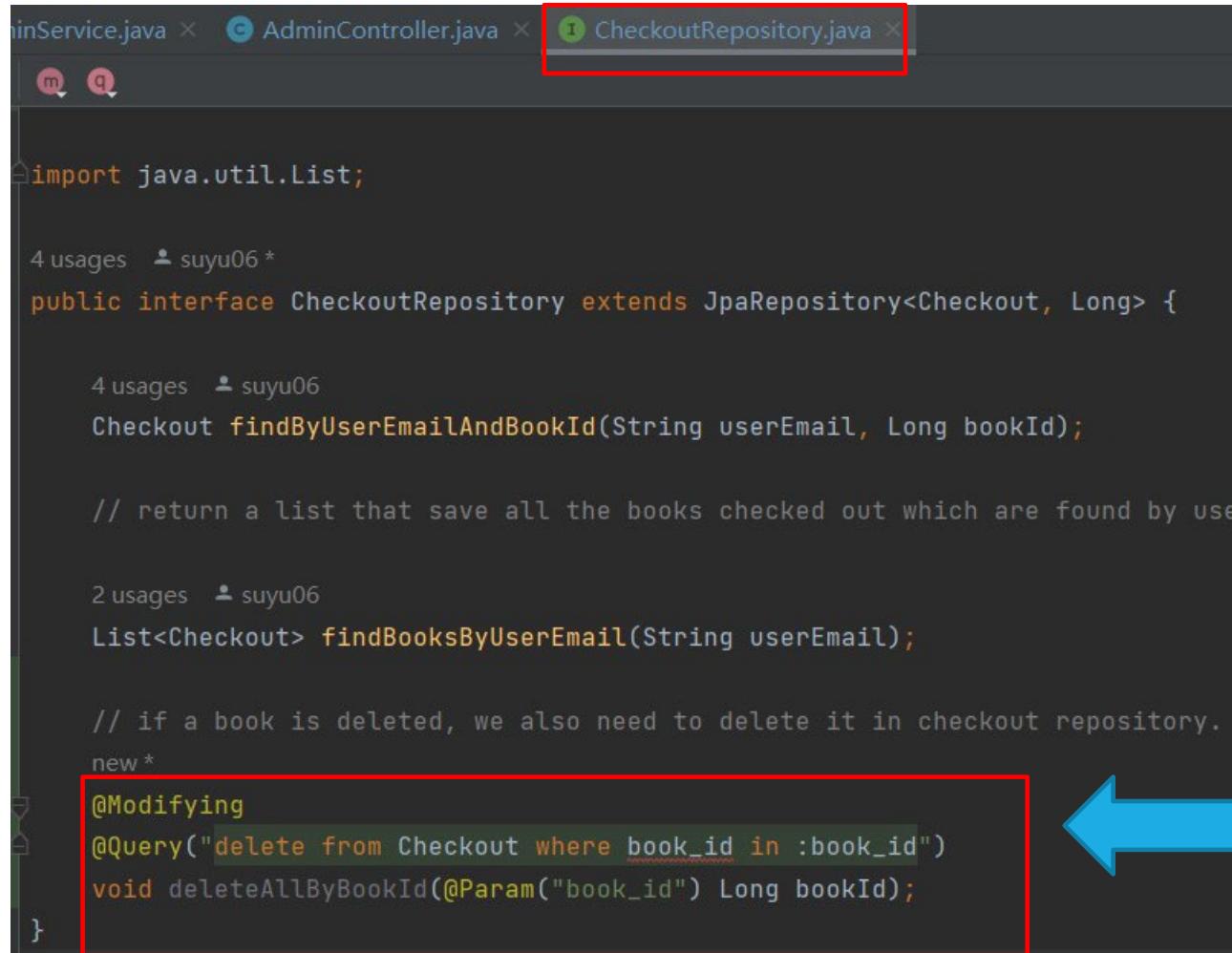


```
AdminController.java
34     }
35     //decrease quantity of book
36     new *
37     @PutMapping("/secure/decrease/book/quantity")
38     public void decreaseBookQuantity(@RequestHeader(value="Authorization") String token,
39                                     @RequestParam Long bookId) throws Exception {
40         // extract the usertype from token string
41         String admin = ExtractJWT.payloadJWTExtraction(token, extraction: "\"userType\"");
42         // if it's not the role of Admin, throw an exception
43         if (admin == null || !admin.equals("admin")) {
44             throw new Exception("Administration page only");
45         }
46         // if it's the role of Admin, call the decreaseBookQuantity() in the adminService
47         adminService.decreaseBookQuantity(bookId);
48     }
49     // add book api
```

---

---

## **6. SPRINGBOOT ENHANCE REPOSITORIES FOR DELETE**



```
minService.java × AdminController.java ×  CheckoutRepository.java ×

import java.util.List;

4 usages  suyu06 *
public interface CheckoutRepository extends JpaRepository<Checkout, Long> {

    4 usages  suyu06
    Checkout findByUserEmailAndBookId(String userEmail, Long bookId);

    // return a list that save all the books checked out which are found by user
    2 usages  suyu06
    List<Checkout> findBooksByUserEmail(String userEmail);

    // if a book is deleted, we also need to delete it in checkout repository.
    new*
    @Modifying
    @Query("delete from Checkout where book_id in :book_id")
    void deleteAllByBookId(@Param("book_id") Long bookId);
}
```

Now, deleting a book is easy. We can delete automatically by just calling our repositories, delete function and passing in the primary key. Well, there is a slight issue when we delete the book.

We also want to delete all of the reviews that have that book and all of the current checkouts that have that book.

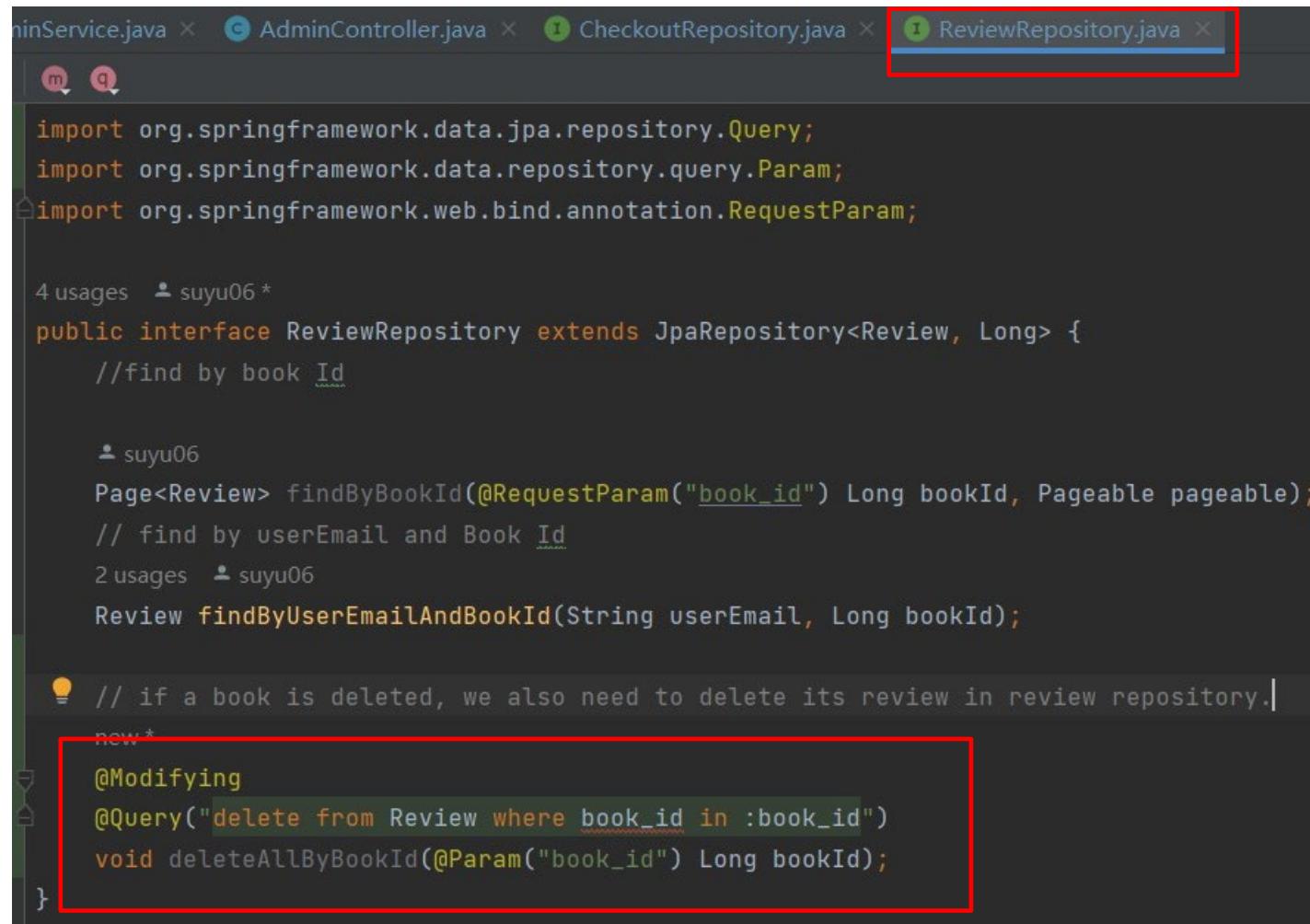
So what we need to do is modify our repositories.

### Step1

Go into dao package, open up CheckoutRepository, Create an interface function deleteAllByBookId.

And add two annotations to it:

```
@Modifying  
@Query
```



```
ReviewRepository.java
```

```
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.web.bind.annotation.RequestParam;

4 usages  suyu06 *
public interface ReviewRepository extends JpaRepository<Review, Long> {
    //find by book Id

    suyu06
    Page<Review> findByBookId(@RequestParam("book_id") Long bookId, Pageable pageable);
    // find by userEmail and Book Id
    2 usages  suyu06
    Review findByUserEmailAndBookId(String userEmail, Long bookId);

    // if a book is deleted, we also need to delete its review in review repository.
    now*
    @Modifying
    @Query("delete from Review where book_id in :book_id")
    void deleteAllByBookId(@Param("book_id") Long bookId);
}
```

## Step 2,

we want to do the exact same thing for our review repository.

So let's now open up our ReviewRepository file, Create an interface function deleteAllByBookId. And add two annotations to it:

@Modifying

@Query

---

---

## **7. SPRINGBOOT DELETE BOOK SERVICE AND CONTROLLER LAYER**



```
Service.java × C AdminController.java × I CheckoutRepository.java × I ReviewReposito
6 usages
private BookRepository bookRepository;

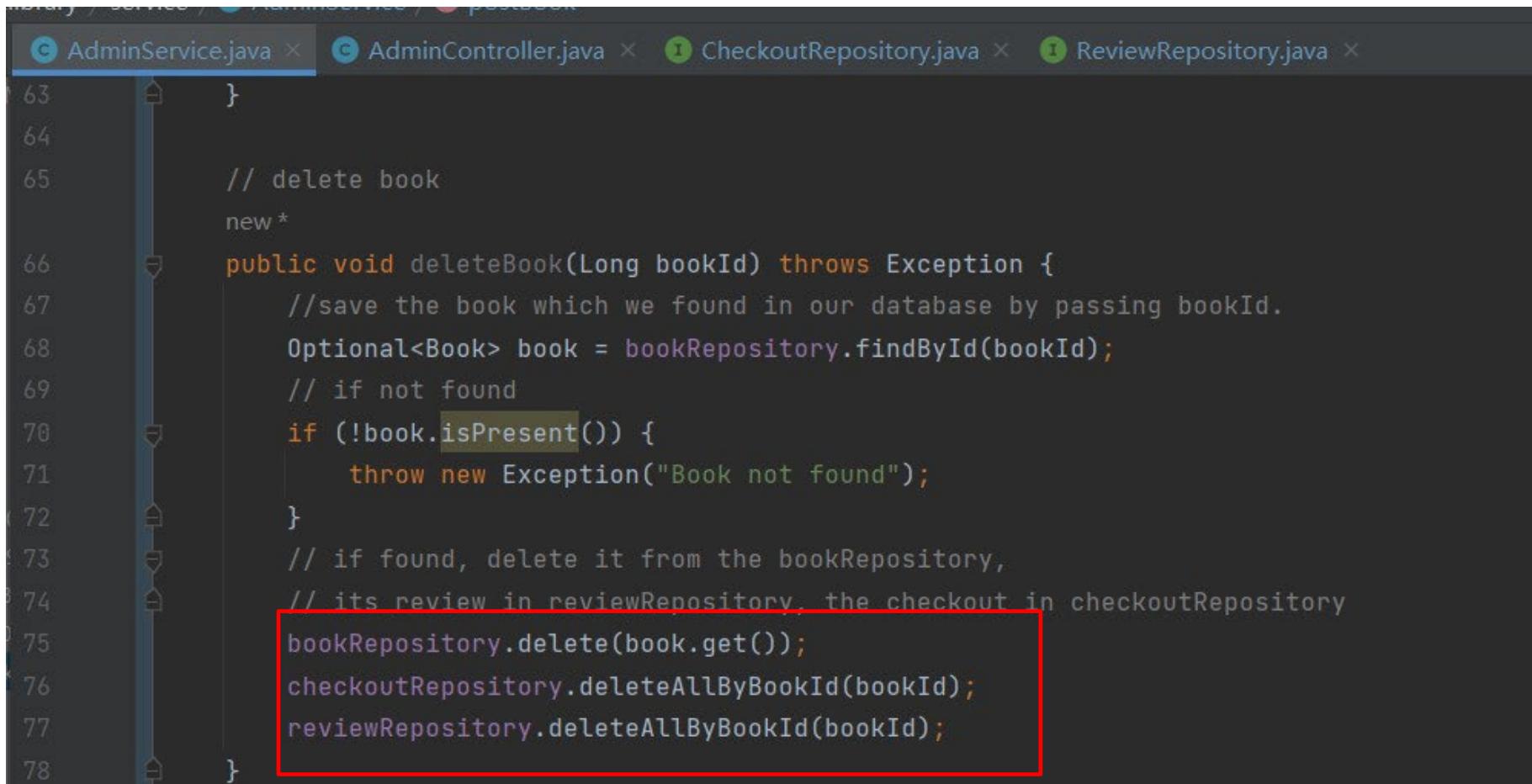
// inject CheckoutRepository and ReviewRepository for updating
1 usage
private ReviewRepository reviewRepository;

1 usage
private CheckoutRepository checkoutRepository;

suyu06 *
@Autowired
public AdminService(BookRepository bookRepository,
                    ReviewRepository reviewRepository,
                    CheckoutRepository checkoutRepository
) {
    this.bookRepository = bookRepository;
    this.reviewRepository = reviewRepository;
    this.checkoutRepository = checkoutRepository;
}
```

Step 1, So let's open up our "service" directory and jump into our "**AdminService**" file. Inject CheckoutRepository and ReviewRepository ,then autowire them to AdminService constructor.

Step 2, In “**AdminService**” file, add deleteBook() function. Find the book by the Id ,then delete the data from the book repository , checkout repository and review repository.



```
63     }
64
65     // delete book
66     new *
67
68     public void deleteBook(Long bookId) throws Exception {
69         //save the book which we found in our database by passing bookId.
70         Optional<Book> book = bookRepository.findById(bookId);
71         // if not found
72         if (!book.isPresent()) {
73             throw new Exception("Book not found");
74         }
75         // if found, delete it from the bookRepository,
76         // its review in reviewRepository, the checkout in checkoutRepository
77         bookRepository.delete(book.get());
78         checkoutRepository.deleteAllByBookId(bookId);
79         reviewRepository.deleteAllByBookId(bookId);
80     }
81 }
```

```
AdminController.java
...
53     String admin = ExtractJWT.payloadJWTExtraction(token, extraction: "\"userType\"");
54     //if not role of Admin, can not use this function
55     if (admin == null || !admin.equals("admin")) {
56         throw new Exception("Administration page only");
57     }
58     //call postBook method in adminService class to save the book from the request to our database
59     adminService.postBook(addBookRequest);
60 }
61
62 @DeleteMapping("/secure/delete/book")
63 public void deleteBook(@RequestHeader(value="Authorization") String token,
64                         @RequestParam Long bookId) throws Exception {
65     // extract user type from the token
66     String admin = ExtractJWT.payloadJWTExtraction(token, extraction: "\"userType\"");
67     //if not role of Admin, can not use this function
68     if (admin == null || !admin.equals("admin")) {
69         throw new Exception("Administration page only");
70     }
71     //if the role is authenticated, call the deleteBook() in adminService.
72     adminService.deleteBook(bookId);
73 }
74 }
```

Step 3 Let's jump into “controller” package, “AdminController” file, use “Delete” method to create endpoint for delete the book.

---

---

---

## **8.REACT-ADD INCREASE FUNCTIONALITY**



Add the functionality for increasing quantity of a book to be able to call our spring boot back into application.

Step1 Go to ChangeQuantityOfBook.tsx, add an async function increaseQuantity.

```
22
23     // call increase quantity endpoint
24     async function increaseQuantity() {
25         // url string
26         const url = `http://localhost:8080/api/admin/secure/increase/book/quantity/?bookId=${props.book?.id}`;
27         // specify method and headers
28         const requestOptions = {
29             method: 'PUT',
30             headers: {
31                 Authorization: `Bearer ${authState?.accessToken?.accessToken}`,
32                 'Content-Type': 'application/json'
33             }
34         };
35         // fetch data
36         const quantityUpdateResponse = await fetch(url, requestOptions);
37         // failure
38         if (!quantityUpdateResponse.ok) {
39             throw new Error('Something went wrong!');
40         }
41         // add copies and copies available by one
42         setQuantity(quantity + 1);
43         setRemaining(remaining + 1);
44     }
```

Step 2 add onClick listener to the “Add Quantity”button. Each time we click the button, the async increaseQuantity() will be called.

```
87      </div>
88      <button className='m1 btn btn-md main-color text-white'>Add Quantity</button>
89      <button className='m1 btn btn-md btn-warning'>Decrease Quantity</button>
90    </div>
```

```
ChangeQuantityOfBooks.tsx 1  ChangeQuantityOfBook.tsx 1, M × ManageLibraryPage.tsx
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx [ChangeQuantityOfBook]
78      </div>
79      <div className='d-flex justify-content-center align-items-center'>
80        <p>Books Remaining: <b>{remaining}</b></p>
81      </div>
82    </div>
83    <div className='mt-3 col-md-1'>
84      <div className='d-flex justify-content-start'>
85        <button className='m-1 btn btn-md btn-danger'>Delete</button>
86      </div>
87    </div>
88    <button className='m1 btn btn-md main-color text-white' onClick={increaseQuantity}>Add Quantity</button>
89    <button className='m1 btn btn-md btn-warning'>Decrease Quantity</button>
90  </div>
91</div>
```

React App    ×    +

localhost:3000/admin

Radio-Canada.ca |... Canada's federal... Sign in to your IR... Chenelière Éducat... Montreal zdm Nos incontournab... TED TED: Ideas worth... 在线课程 - 时

JAC Read Home Search Books Shelf

## Manage Library

Add new book Change quantity Messages

Number of results: (23)  
1 to 5 of 23 items:

Luv, Zofia  
**JavaScript Cookbook**  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin risus tortor, condimentum eget sapien ac, dapibus varius ligula. Maecenas justo erat, semper sed nunc vel, vulputate eleifend dui. Integer id ipsum vitae nisi malesuada feugiat. Proin sit amet quam laoreet, feugiat mi vitae, vestibulum dui. Aliquam erat volutpat. Etiam hendrerit erat nec mi auctor elementum. Curabitur vestibulum lectus a ante tempor tincidunt et sed orci. Proin maximus tortor in risus auctor efficitur. Phasellus quam mauris, laoreet et feugiat ac, imperdiet at quam. Nullam sollicitudin nec diam vel finibus.

Total Quantity: **10**  
Books Remaining: **10**

Delete

Add Quantity

Decrease Quantity

Now, with admin user logged in status, visit <http://localhost:3000/admin>, Click on the “change quantity” tab,

Scroll to the last book we just added , click “Add Quantity” button,

New Author1  
**Advanced Tech in C#**  
great C# book.

Total Quantity: 6  
Books Remaining: 5

Add Quantity

Decrease Quantity

Both the number of total quantity and books remaining are increased by 1.

New Author1  
**Advanced Tech in C#**  
great C# book.

Total Quantity: 7  
Books Remaining: 6

Add Quantity

Decrease Quantity

Let's check in our database.

The two numbers are increased by 1 too !

13	Exploring DevOps	Luv, Mariana	Lorem ipsum dolor sit amet, consectetur adipisci...	20	20	DevOps	BLOB
14	Introduction to SQL	Luv, Hugo	Pellentesque varius aliquam lacus quis rhoncus. ...	20	20	Data	BLOB
15	The Expert Guide to JavaScript	Luv, Elena	Lorem ipsum dolor sit amet, consectetur adipisci...	10	10	FE	BLOB
16	Exploring React.js	Luv, Filip	Pellentesque varius aliquam lacus quis rhoncus. ...	15	15	FE	BLOB
17	Advanced Techniques in React.js	Luv, Mary	Proin at urna faucibus, pretium mi in, dapibus m...	10	10	FE	BLOB
18	Introduction to C#	Luv, Bill	Nunc eget lorem ac neque tincidunt mollis. Fusc...	12	12	BE	BLOB
19	Crash Course in JavaScript	Luv, Frank	Morbi eu tempus eros, in imperdiet sem. Nulla s...	20	20	FE	BLOB
20	Introduction to Machine Learning	Luv, Camila	Lorem ipsum dolor sit amet, consectetur adipisci...	20	20	Data	BLOB
21	Become a Guru in Java	Luv, Priya	Lorem ipsum dolor sit amet, consectetur adipisci...	20	20	BE	BLOB
22	Introduction to Python	Luv, Juan	Pellentesque varius aliquam lacus quis rhoncus...	20	20	BE	BLOB
23	Advanced Tech in C#	New Author1	great C# book.	7	6	BE	BLOB

---

---

---

## **9.REACT- ADD DECREASE FUNCTIONALITY**



Same process as the increase quantity.

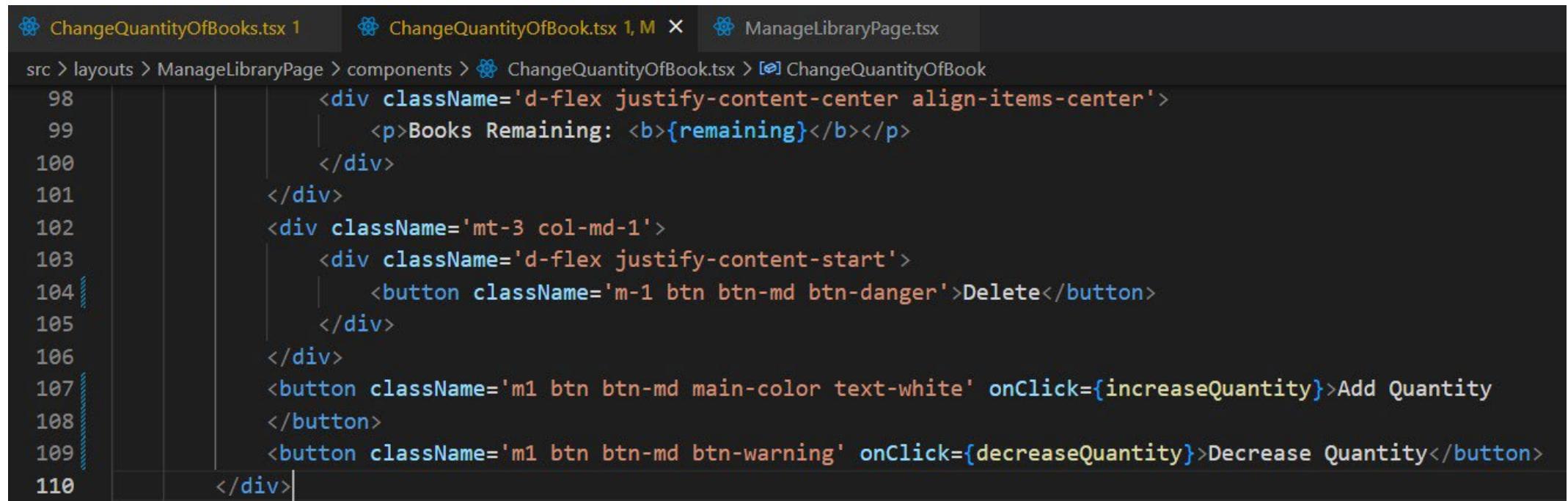
Step1 Go to ChangeQuantityOfBook.tsx, add an async function decreaseQuantity().

The screenshot shows a code editor with several tabs at the top: 'ChangeQuantityOfBooks.tsx 1', 'ChangeQuantityOfBook.tsx 2, M X', and 'ManageLibraryPage.tsx'. The active tab is 'ChangeQuantityOfBook.tsx 2, M'. The code is written in TypeScript and is part of a component named 'ChangeQuantityOfBook'. The function 'decreaseQuantity' is defined as follows:

```
44     }
45     // call decrease quantity endpoint
46     async function decreaseQuantity() {
47       const url = `http://localhost:8080/api/admin/secure/decrease/book/quantity/?bookId=${props.book?.id}`;
48       const requestOptions = {
49         method: 'PUT',
50         headers: {
51           Authorization: `Bearer ${authState?.accessToken?.accessToken}`,
52           'Content-Type': 'application/json'
53         }
54       };
55
56       const quantityUpdateResponse = await fetch(url, requestOptions);
57       if (!quantityUpdateResponse.ok) {
58         throw new Error('Something went wrong!');
59       }
60       setQuantity(quantity - 1);
61       setRemaining(remaining - 1);
62     }
63   }
```

The code uses the Fetch API to send a PUT request to the specified URL with the provided headers and body. If the response is not successful, it throws an error. Otherwise, it updates the state with the new quantity and remaining values.

Step 2 add onClick listener to the “Decrease Quantity” button. Each time we click the button, the async decreaseQuantity() will be called.



```
ChangeQuantityOfBooks.tsx 1 ChangeQuantityOfBook.tsx 1, M ManageLibraryPage.tsx
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx > [?] ChangeQuantityOfBook
  98     <div className='d-flex justify-content-center align-items-center'>
  99       <p>Books Remaining: <b>{remaining}</b></p>
 100      </div>
 101    </div>
 102    <div className='mt-3 col-md-1'>
 103      <div className='d-flex justify-content-start'>
 104        <button className='m-1 btn btn-md btn-danger'>Delete</button>
 105      </div>
 106    </div>
 107    <button className='m1 btn btn-md main-color text-white' onClick={increaseQuantity}>Add Quantity
 108      </button>
 109      <button className='m1 btn btn-md btn-warning' onClick={decreaseQuantity}>Decrease Quantity</button>
 110    </div>
```

Scroll to the last book we just added , click “Decrease Quantity” button,

New Author1  
**Advanced Tech in C#**  
great C# book.

Total Quantity: 7  
Books Remaining: 6

Delete

Add Quantity

Decrease Quantity

Both the number of total quantity and books remaining are decreased by 1.

New Author1  
**Advanced Tech in C#**  
great C# book.

Total Quantity: 6  
Books Remaining: 5

Delete

Add Quantity

Decrease Quantity

Let's check in our database.

The two numbers are decreased by 1 too !

#	Title	Author	Description	Rating	Reviews	Status	Data
15	The Expert Guide to JavaScript	Luv, Mayo	Pellentesque varius aliquam lacus quis rhoncus. ...	10	10	FE	BLOB
16	Exploring React.js	Luv, Elena	Lorem ipsum dolor sit amet, consectetur adipisci...	15	15	FE	BLOB
17	Advanced Techniques in React.js	Luv, Filip	Pellentesque varius aliquam lacus quis rhoncus. ...	10	10	FE	BLOB
18	Introduction to C#	Luv, Mary	Proin at urna faucibus, pretium mi in, dapibus m...	12	12	BE	BLOB
19	Crash Course in JavaScript	Luv, Bill	Nunc eget lorem ac neque tincidunt mollis. Fusc...	20	20	FE	BLOB
20	Introduction to Machine Learning	Luv, Frank	Morbi eu tempus eros, in imperdiet sem. Nulla s...	20	20	Data	BLOB
21	Introduction to Python	Luv, Camila	Lorem ipsum dolor sit amet, consectetur adipisci...	20	20	BE	BLOB
22	Become a Guru in Java	Luv, Priya	Lorem ipsum dolor sit amet, consectetur adipisci...	20	20	BE	BLOB
23	Advanced Tech in C#	New Author1	great C# book.	6	5	BE	BLOB

ok 2 ×

---

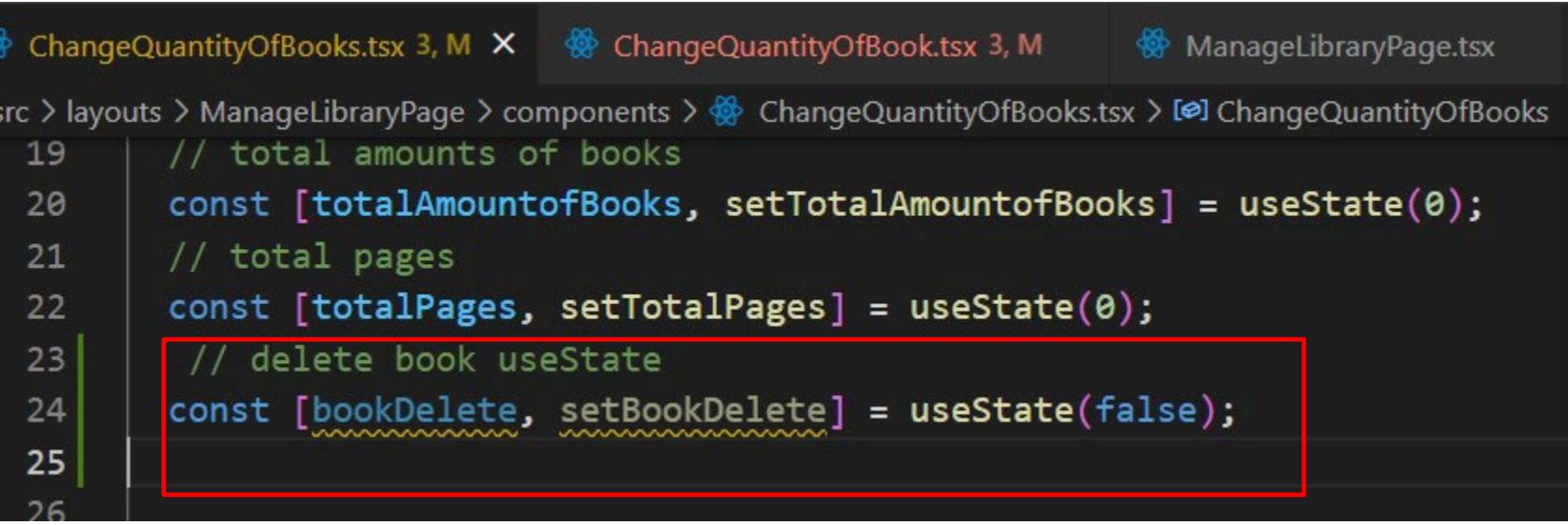
---

---

# **10. REACT-IMPLEMENTATION OF DELETE**



Step1 Go to ChangeQuantityOfBooks.tsx,create bookDelete useState.



The screenshot shows a code editor with three tabs at the top: 'ChangeQuantityOfBooks.tsx 3, M X', 'ChangeQuantityOfBook.tsx 3, M', and 'ManageLibraryPage.tsx'. Below the tabs, the file path is shown as 'src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > [o] ChangeQuantityOfBooks'. The code itself is as follows:

```
19 // total amounts of books
20 const [totalAmountofBooks, setTotalAmountofBooks] = useState(0);
21 // total pages
22 const [totalPages, setTotalPages] = useState(0);
23 // delete book useState
24 const [bookDelete, setBookDelete] = useState(false);
25
26
```

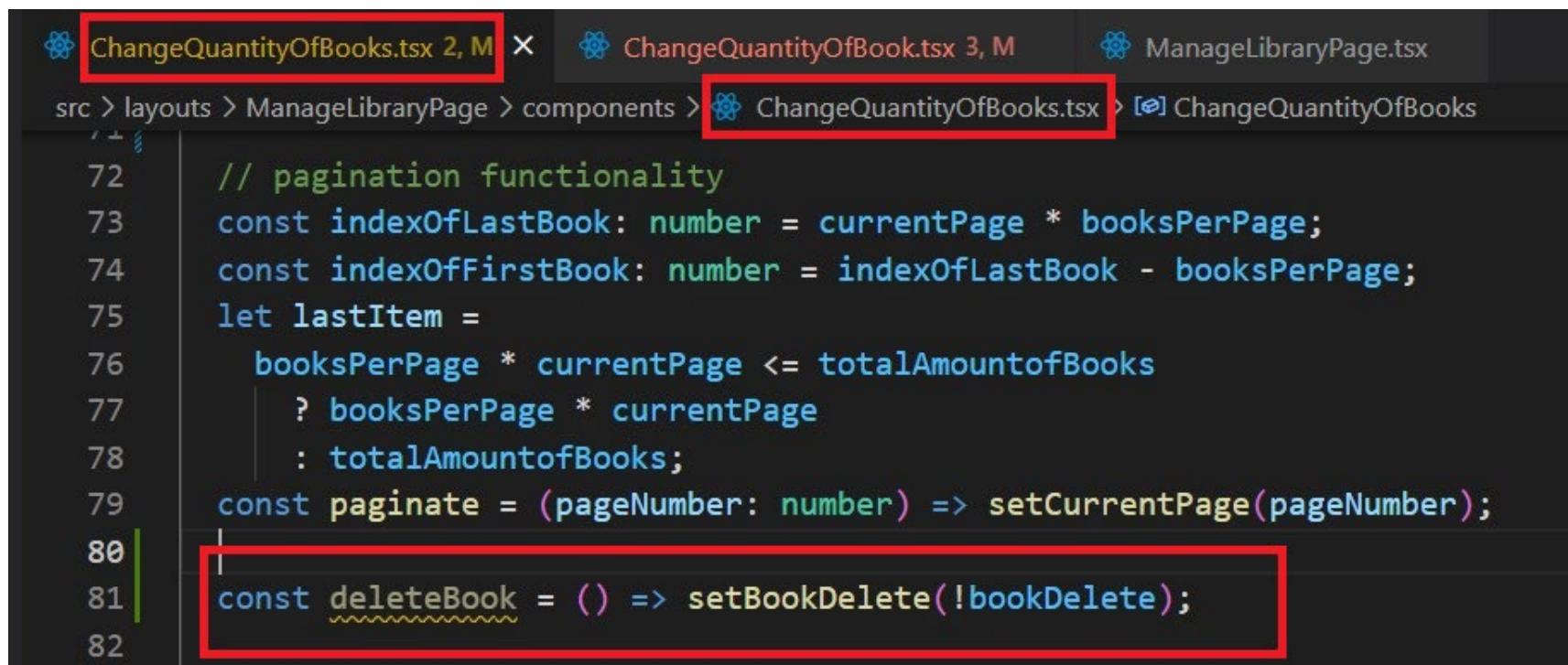
A red rectangular box highlights the entire line 24, which defines the state variable 'bookDelete' using the useState hook.

Step 2, add this useState as a state change factor in the end of our array of use effect. So every time this state changes, the app will recall the useEffect.

```
ChangeQuantityOfBooks.tsx 2, M × ChangeQuantityOfBook.tsx 3, M ManageLibraryPage.tsx
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > [o] ChangeQuantityOfBooks
22 const [totalPages, setTotalPages] = useState(0);
23 // delete book useState
24 const [bookDelete, setBookDelete] = useState(false);
25
26
27 // copy useEffect, isLoading and httpError from SearchBooksPage.tsx
28 > useEffect(() => {
29   },
30   [currentPage, bookDelete]);
31 // pagination functionality
32 const indexOfLastBook: number = currentPage * booksPerPage;
```

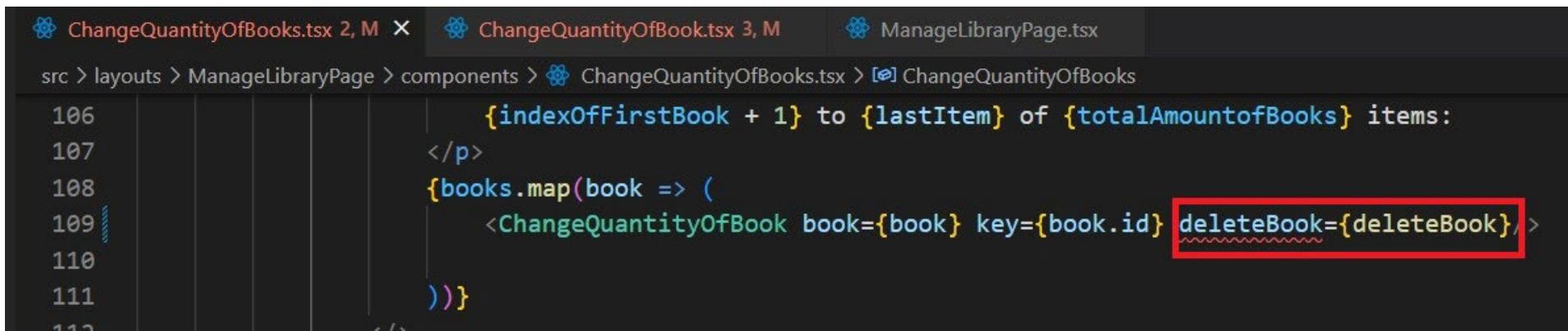
Step 3, add

```
const deleteBook = () => setBookDelete(!bookDelete);
```



```
// pagination functionality
const indexOfLastBook: number = currentPage * booksPerPage;
const indexOfFirstBook: number = indexOfLastBook - booksPerPage;
let lastItem =
  booksPerPage * currentPage <= totalAmountofBooks
    ? booksPerPage * currentPage
    : totalAmountofBooks;
const paginate = (pageNumber: number) => setCurrentPage(pageNumber);
const deleteBook = () => setBookDelete(!bookDelete);
```

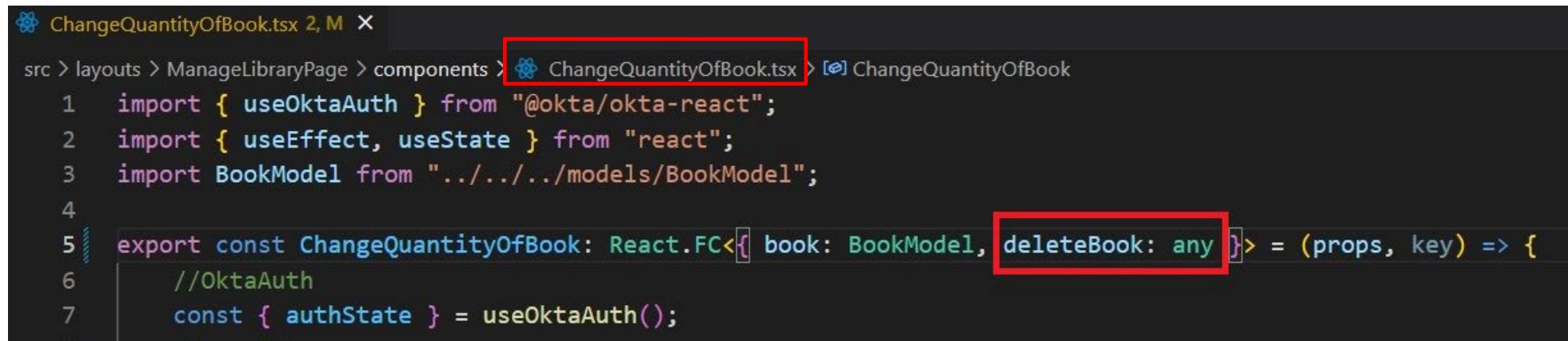
Step 4, go to books.map function, passing deleteBook to it.



The screenshot shows a code editor with a dark theme. The file is `ChangeQuantityOfBooks.tsx`. The code is a part of a `ChangeQuantityOfBook` component, specifically the `ChangeQuantityOfBooks` section. It contains a `map` function that iterates over the `books` array. The `map` function takes a `book` as a parameter and renders a `ChangeQuantityOfBook` component for each book. The `ChangeQuantityOfBook` component is passed the `book`, `key`, and `deleteBook` props. The `deleteBook` prop is highlighted with a red box.

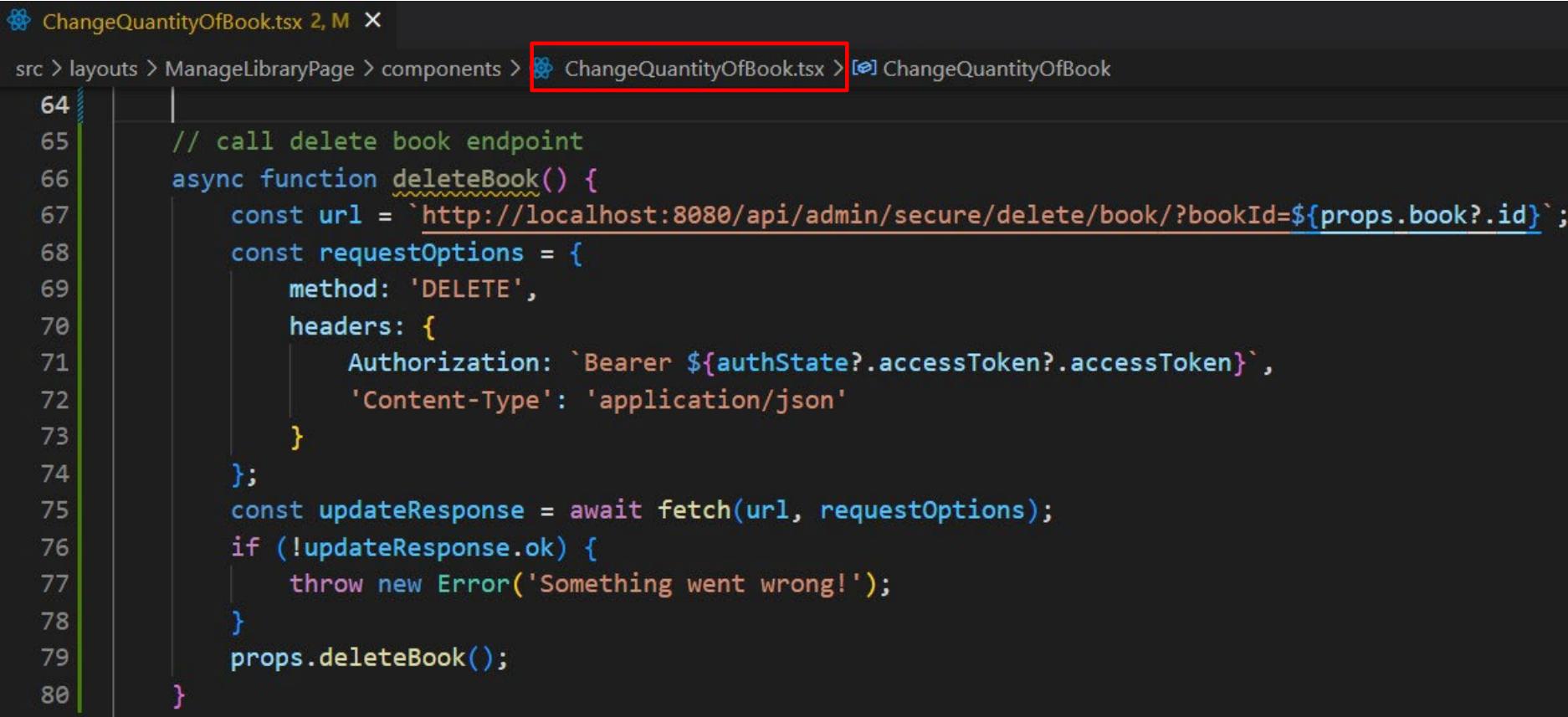
```
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBooks.tsx > ChangeQuantityOfBooks
106           {indexOfFirstBook + 1} to {lastItem} of {totalAmountofBooks} items:
107           </p>
108           {books.map(book => (
109             <ChangeQuantityOfBook book={book} key={book.id} deleteBook={deleteBook}>
110               ))}
111           </div>
```

Step 5 Go to ChangeQuantityOfBook.tsx, passing deleteBook another props.



```
ChangeQuantityOfBook.tsx 2, M X
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx > ChangeQuantityOfBook
1 import { useOktaAuth } from "@okta(okta-react";
2 import { useEffect, useState } from "react";
3 import BookModel from "../../models/BookModel";
4
5 export const ChangeQuantityOfBook: React.FC<{ book: BookModel, deleteBook: any }> = (props, key) => {
6   //OktaAuth
7   const { authState } = useOktaAuth();
8 }
```

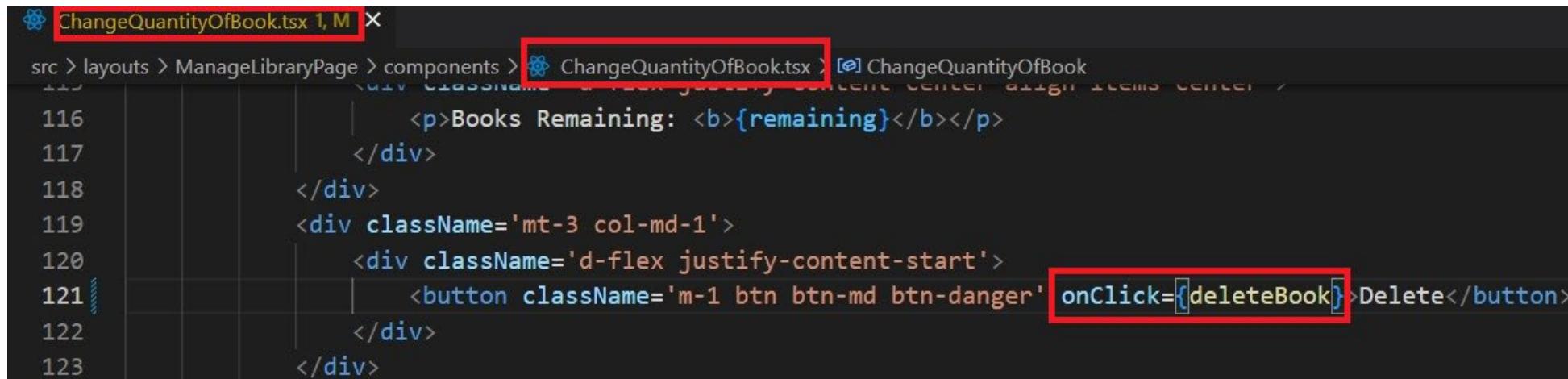
Step 6, add an async function deleteBook().



```
ChangeQuantityOfBook.tsx 2, M X
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx [e] ChangeQuantityOfBook

64
65 // call delete book endpoint
66 async function deleteBook() {
67   const url = `http://localhost:8080/api/admin/secure/delete/book/?bookId=${props.book?.id}`;
68   const requestOptions = {
69     method: 'DELETE',
70     headers: {
71       Authorization: `Bearer ${authState?.accessToken?.accessToken}`,
72       'Content-Type': 'application/json'
73     }
74   };
75   const updateResponse = await fetch(url, requestOptions);
76   if (!updateResponse.ok) {
77     throw new Error('Something went wrong!');
78   }
79   props.deleteBook();
80 }
```

Step 7 add onClick listener to the button “Delete”.  
So each time we click the button, the async deleteBook() will be called.



```
ChangeQuantityOfBook.tsx 1, M X
src > layouts > ManageLibraryPage > components > ChangeQuantityOfBook.tsx [ChangeQuantityOfBook]
  116   |     <p>Books Remaining: <b>{remaining}</b></p>
  117   |   </div>
  118 </div>
  119 <div className='mt-3 col-md-1'>
  120   <div className='d-flex justify-content-start'>
  121     <button className='m-1 btn btn-md btn-danger' onClick={deleteBook}>Delete</button>
  122   </div>
  123 </div>
```

Now, with admin user logged in status, visit <http://localhost:3000/admin>, Click on the “change quantity” tab.

Scroll down the page, find the book we added recently. Click on the button “Delete”.

The screenshot shows a web browser window with the title "React App" and the URL "localhost:3000/admin". The page displays a list of books. One book, "Advanced Tech in C#", has a green callout bubble pointing to its "Delete" button. The book details are as follows:

New Author	Total Quantity: 6
Advanced Tech in C#	Books Remaining: 5
great C# book.	

Below the book details is a control panel with "Add Quantity" and "Decrease Quantity" buttons. The entire screenshot is framed by a thick black border.

Now, the page only show us 22 books instead of 23.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/admin". The page has a blue header bar with the text "JAC Read" and navigation links for "Home", "Search Books", and "Shelf". Below the header, the main content area is titled "Manage Library". A red box highlights the text "Number of results: (22)". Below this, it says "21 to 22 of 22 items:". There is a card for a book titled "Become a Guru in Java" by "Luv, Priya". The card includes the book's cover image, the author's name, the title, a short description, and the quantities "Total Quantity: 20" and "Books Remaining: 20".

React App

localhost:3000/admin

Radio-Canada.ca |... Canada's federal... G Sign in to your IR... Chenelière Éducat... Montreal zdm Nos incontournab... TED TED: Ideas worth... 在线

JAC Read Home Search Books Shelf

Manage Library

Add new book Change quantity Messages

Number of results: (22)

21 to 22 of 22 items:

Luv, Priya

Become a Guru in Java

Maecenas justo erat, semper sed nunc vel, vulputate eleifend

Books Remaining: 20

Total Quantity: 20

Open book table in the database, we have 22 books here too. The book whose Id is 23 disappeared.

The screenshot shows the MySQL Workbench interface. At the top, there are tabs for 'book', 'review', 'checkout', 'history', 'messages', 'checkout', 'history', 'messages', and 'review'. The 'review' tab is highlighted with a red box. Below the tabs is a toolbar with various icons. A query editor window displays the SQL command: 'SELECT \* FROM reactlibrarydatabase.review;'. The results grid shows a table with the following data:

	id	user_email	date	rating	book_id	review_description
▶	1	example1user@email.com	2023-02-05 23:11:33.000000	4.00	1	First book is pretty good book overall
	2	example2user@email.com	2023-02-05 23:11:33.000000	4.50	2	Second books is pretty good book overall
	3	testuser2@email.com	2023-02-12 23:29:44.548000	5.00	4	Very good book!
	4	testuser2@email.com	2023-02-13 03:41:05.256000	3.00	2	this book is pretty average.
	5	testuser2@email.com	2023-02-13 03:43:04.235000	5.00	3	
	6	testuser2@email.com	2023-02-13 03:47:48.924000	4.50	5	
*	7	adminuser@email.com	2023-02-16 17:01:58.784000	5.00	23	great book
		NULL	NULL	NULL	NULL	NULL

Before we click on the “Delete”, in our “review” table, there is a piece of data showing admin user has a review for the 23<sup>rd</sup> book, it’s has a rating of five star and a description “great book”.

The screenshot shows a MySQL Workbench interface. At the top, there are tabs for 'book', 'review', 'checkout', 'history', 'messages', 'checkout', 'history', 'messages', and 'review'. The 'review' tab is active. Below the tabs is a toolbar with various icons for file operations, search, and navigation. A query editor window contains the SQL command: `1 • SELECT * FROM reactlibrarydatabase.review;`. The result grid below displays the following data:

	id	user_email	date	rating	book_id	review_description
▶	1	example1user@email.com	2023-02-05 23:11:33.000000	4.00	1	First book is pretty good book overall
▶	2	example2user@email.com	2023-02-05 23:11:33.000000	4.50	2	Second books is pretty good book overall
▶	3	testuser2@email.com	2023-02-12 23:29:44.548000	5.00	4	Very good book!
▶	4	testuser2@email.com	2023-02-13 03:41:05.256000	3.00	2	this book is pretty average.
▶	5	testuser2@email.com	2023-02-13 03:43:04.235000	5.00	3	
▶	6	testuser2@email.com	2023-02-13 03:47:48.924000	4.50	5	
*	NULL	NULL	NULL	NULL	NULL	NULL

After click on the “Delete”,  
Review from the admin for  
the book whose Id is 23 has  
been deleted too.

The screenshot shows a MySQL Workbench interface. At the top, there are tabs: book, review, checkout, history, messages, checkout (which is selected), and history. Below the tabs is a toolbar with various icons. A query editor window contains the SQL command: `1 • SELECT * FROM reactlibrarydatabase.checkout;`. The result grid below shows the data from the 'checkout' table:

	id	user_email	checkout_date	return_date	book_id
▶	8	testuser2@email.com	2023-02-14	2023-02-21	5
*	9	adminuser@email.com	2023-02-16	2023-02-23	23

The row for user 'adminuser' (id 9) is highlighted with a red box.

Before we click on the “Delete”, in our “checkout” table, there is a piece of data showing admin user has checked out the book whose Id is 23.

book review checkout history messages **checkout** x history

1 • SELECT \* FROM reactlibrarydatabase.checkout;

Result Grid | Filter Rows: | Edit: | Export/Import:

	id	user_email	checkout_date	return_date	book_id
▶	8	testuser2@email.com	2023-02-14	2023-02-21	5
*	NULL	NULL	NULL	NULL	NULL

After click on the “Delete”,  
Checkout data for the book whose Id is  
23 has been deleted too.

---

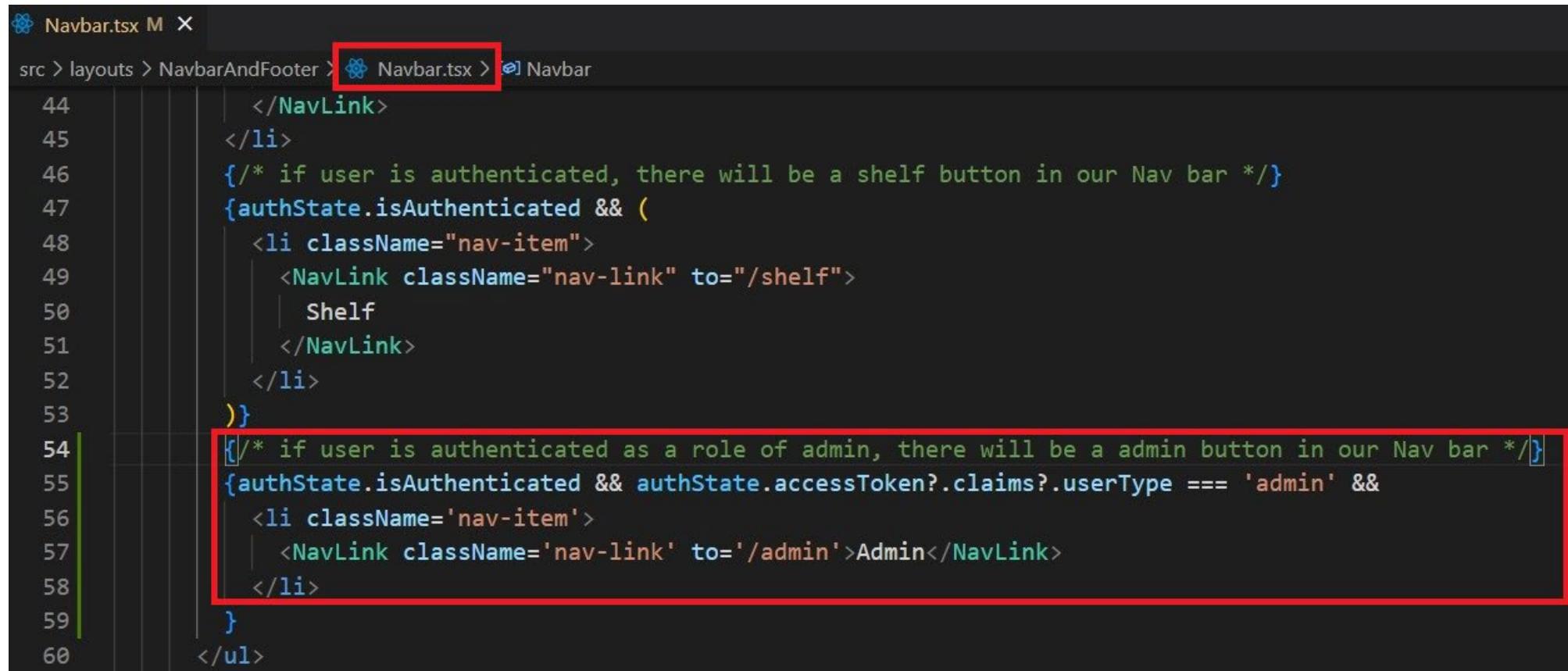
---

# **11. REACT ENHANCE NAVIGATION BAR FOR ADMIN**



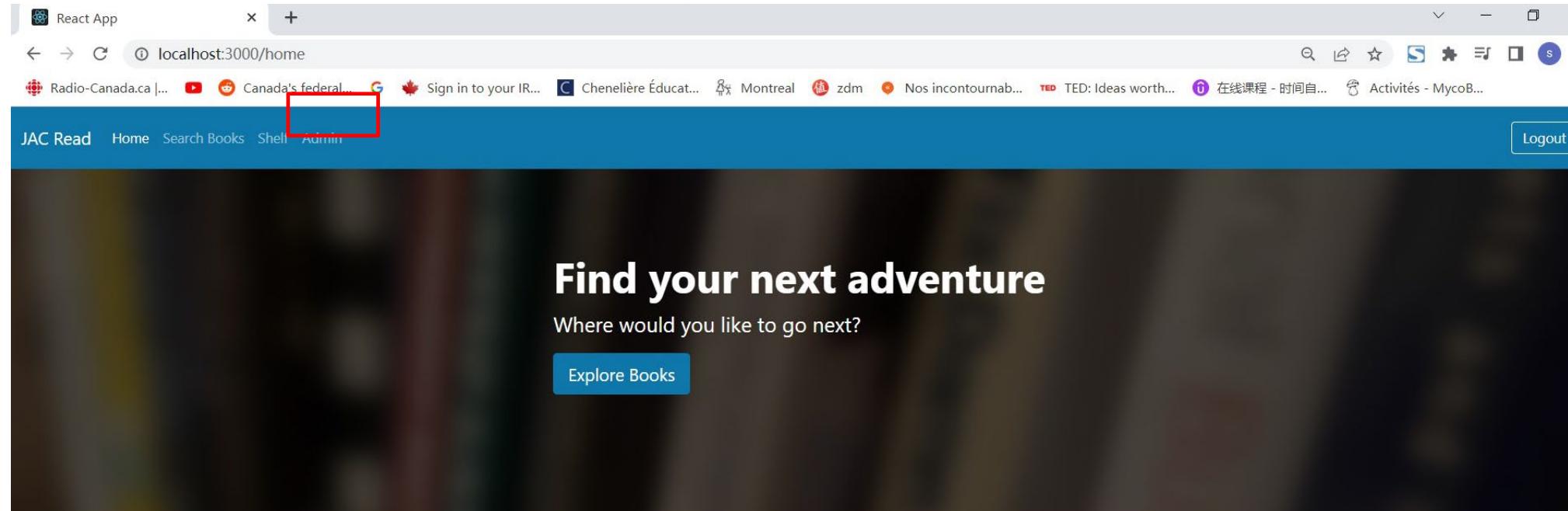
So let's go to layouts->NavbarAndFooter, and open up our Navbar.tsx., add a judgement for the "Admin" Link in the navigation bar.

Only if the user is logged in and the user is Admin, then the "Admin" will show up in the navigation bar.



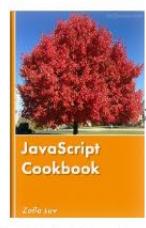
```
Navbar.tsx M X
src > layouts > NavbarAndFooter > Navbar.tsx > Navbar
44     |   </NavLink>
45     | </li>
46     | /* if user is authenticated, there will be a shelf button in our Nav bar */
47     | {authState.isAuthenticated && (
48     |   <li className="nav-item">
49       |     <NavLink className="nav-link" to="/shelf">
50         |       Shelf
51       |     </NavLink>
52     |   </li>
53   )} 
54   | /* if user is authenticated as a role of admin, there will be a admin button in our Nav bar */
55   | {authState.isAuthenticated && authState.accessToken?.claims?.userType === 'admin' &&
56   |   <li className='nav-item'>
57     |     <NavLink className='nav-link' to='/admin'>Admin</NavLink>
58   |   </li>
59   }
60 </ul>
```

Still logged in as an admin user, we can see there is a link “Admin” in the navigation bar.  
Click on it.



A screenshot of a web browser window titled "React App". The address bar shows "localhost:3000/home". The navigation bar includes links for "JAC Read", "Home", "Search Books", "Shelf", and "Admin". The "Admin" link is highlighted with a red box. The main content area features a dark background with the text "Find your next adventure" and "Where would you like to go next?". A blue button labeled "Explore Books" is visible. The URL in the address bar is "localhost:3000/home".

Find your next "I stayed up too late reading" book



JavaScript Cookbook



Become a Guru in JavaScript



Exploring Vue.js

Here brings us to the Admin page. We can add new book.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/admin". The page has a dark blue header with the text "JAC Read" and navigation links "Home", "Search Books", "Shelf", and "Admin". On the far right of the header is a "Logout" button. Below the header, the main content area is titled "Manage Library". It features a form for adding a new book, with fields for "Title", "Author", "Category", "Description", "Copies", and a file upload section. The "Add Book" button is at the bottom of the form.

Manage Library

Add new book   Change quantity   Messages

Add a new book

Title	Author	Category
<input type="text"/>	<input type="text"/>	<input type="button" value="Category ▾"/>

Description

Copies

No file chosen

React App

localhost:3000/admin

Radio-Canada.ca |... Canada's federal... Sign in to your IR... Chenelière Éducat... Montreal zdm Nos incontournab... TED Ideas worth... 在线课程 - 时间自...

JAC Read Home Search Books Shelf Admin

## Manage Library

[Add new book](#)[Change quantity](#)[Messages](#)

Number of results: (22)

1 to 5 of 22 items:



Luv, Zofia

**JavaScript Cookbook**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin risus tortor, condimentum eget sapien ac, dapibus varius ligula. Maecenas justo erat, semper sed nunc vel, vulputate eleifend dui. Integer id ipsum vitae nisi malesuada feugiat. Proin sit amet quam laoreet, feugiat mi vitae, vestibulum dui. Aliquam erat volutpat. Etiam hendrerit erat nec mi auctor elementum. Curabitur vestibulum lectus a ante tempor tincidunt et sed orci. Proin maximus tortor in risus auctor efficitur. Phasellus quam mauris, laoreet et feugiat ac, imperdiet at quam. Nullam sollicitudin nec diam vel finibus.

Total Quantity: **10**

Books Remaining: **10**

[Delete](#)[Add Quantity](#)[Decrease Quantity](#)

We can change  
the book's quantity.

React App

localhost:3000/admin

JAC Read Home Search Books Shelf Admin

## Manage Library

Add new book Change quantity Messages

Pending Q/A:

Case #2: test title  
testuser2@email.com  
test question

Response:  
Description

Submit Response

Case #3: test move banner  
testuser2@email.com  
successful or not

Response:  
Description

We can manage messages.

Then, we log out.

React App

localhost:3000/home

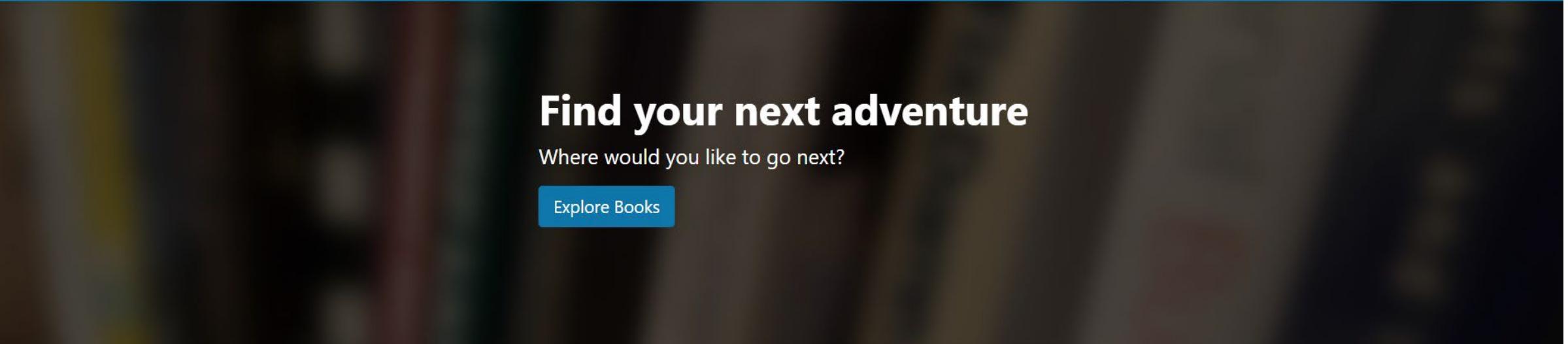
Radio-Canada.ca |... Canada's federal... Sign in to your IR... Chenelière Éducat... Montreal zdm Nos incontournab... TED TED: Ideas worth... 在线课程 - 时间自... Activités - MycoB...

JAC Read Home Search Books Sign in

# Find your next adventure

Where would you like to go next?

Explore Books



Find your next "I stayed up too late reading" book



Log in with the [testuser2@email.com](mailto:testuser2@email.com) as a non admin user,

A screenshot of a web browser window titled "React App". The address bar shows "localhost:3000/login". Below the address bar, there is a horizontal row of various website icons and names. The main content area has a dark blue header with the text "JAC Read" and "Home Search Books".

The login form is contained within a light gray rectangular box. It features a title "Sign In" at the top center. Below it is a "Username" label with an input field containing "testuser2@email.com". Underneath is a "Password" label with an input field containing "\*\*\*\*\*". A "Remember me" checkbox is checked and located below the password field. At the bottom of the form is a large blue "Sign In" button. Below the button, the text "Need help signing in?" is displayed.

Sign In	
Username	testuser2@email.com
Password	*****
<input checked="" type="checkbox"/> Remember me	
<b>Sign In</b>	
Need help signing in?	

Now, in the navigation bar, there are only three links: "Home", "Search Books", "Shelf". Non admin user can not reach the link "Admin" and thus has no right to manage books and respond the messages.

A screenshot of a web browser window showing a simplified navigation bar. The address bar displays "localhost:3000/home". The navigation bar includes links for "Home", "Search Books", and "Shelf", which are highlighted with a red border. Other links like "Admin", "Logout", and various social sharing icons are also present but not highlighted.

JAC Read Home Search Books Shelf Logout

# Find your next adventure

Where would you like to go next?

Explore Books

Find your next "I stayed up too late reading" book

