Week 11 Portfolio

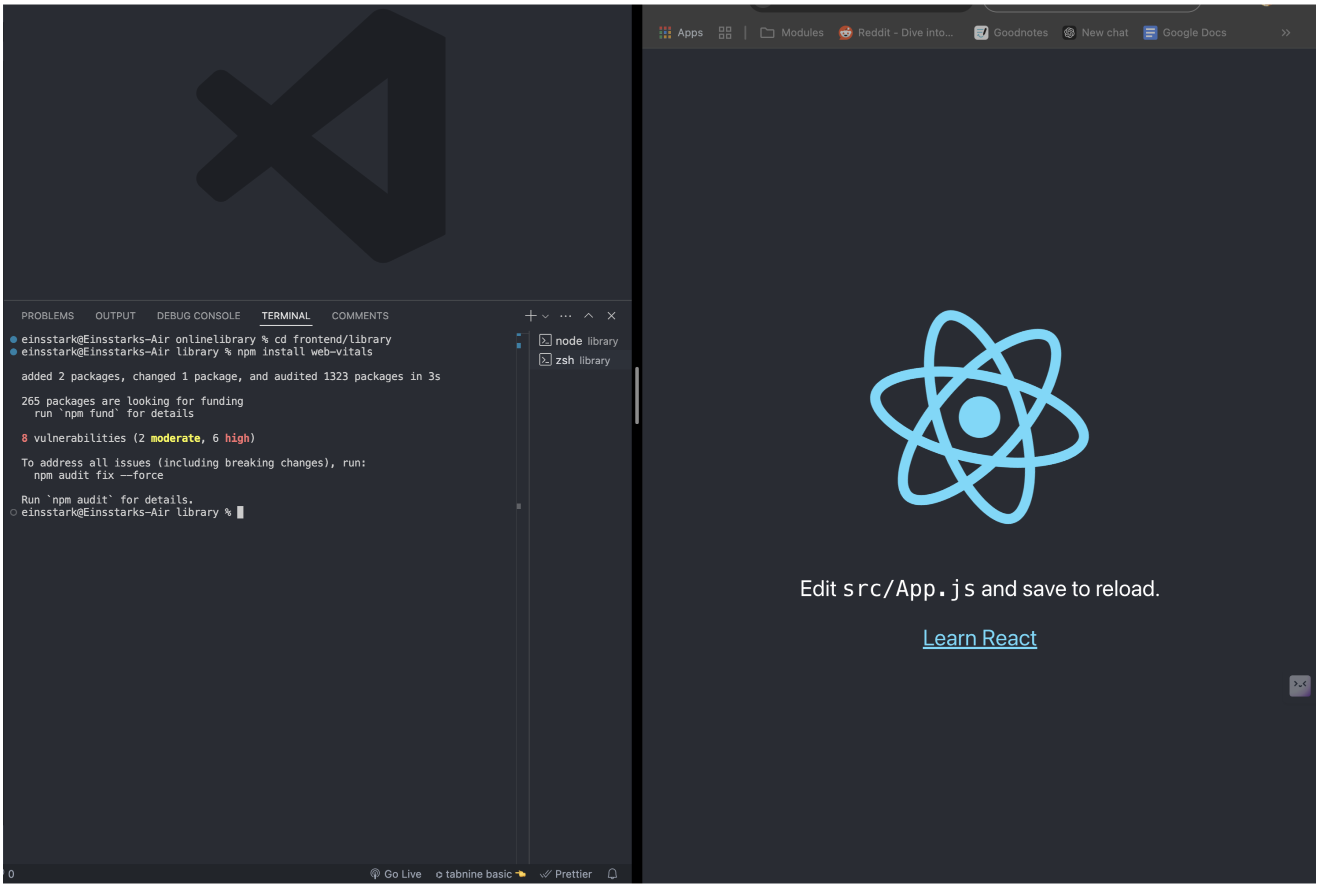
Name: Deepak Kumar
Student ID: 2284279

Frontend for Online library

1. Screenshots

a. React application set up

I created a React app using `npx create-react-app` and made sure it worked by running `npm start`. Although I had to install web vitals since it was missing. It opened on `localhost:3000`.

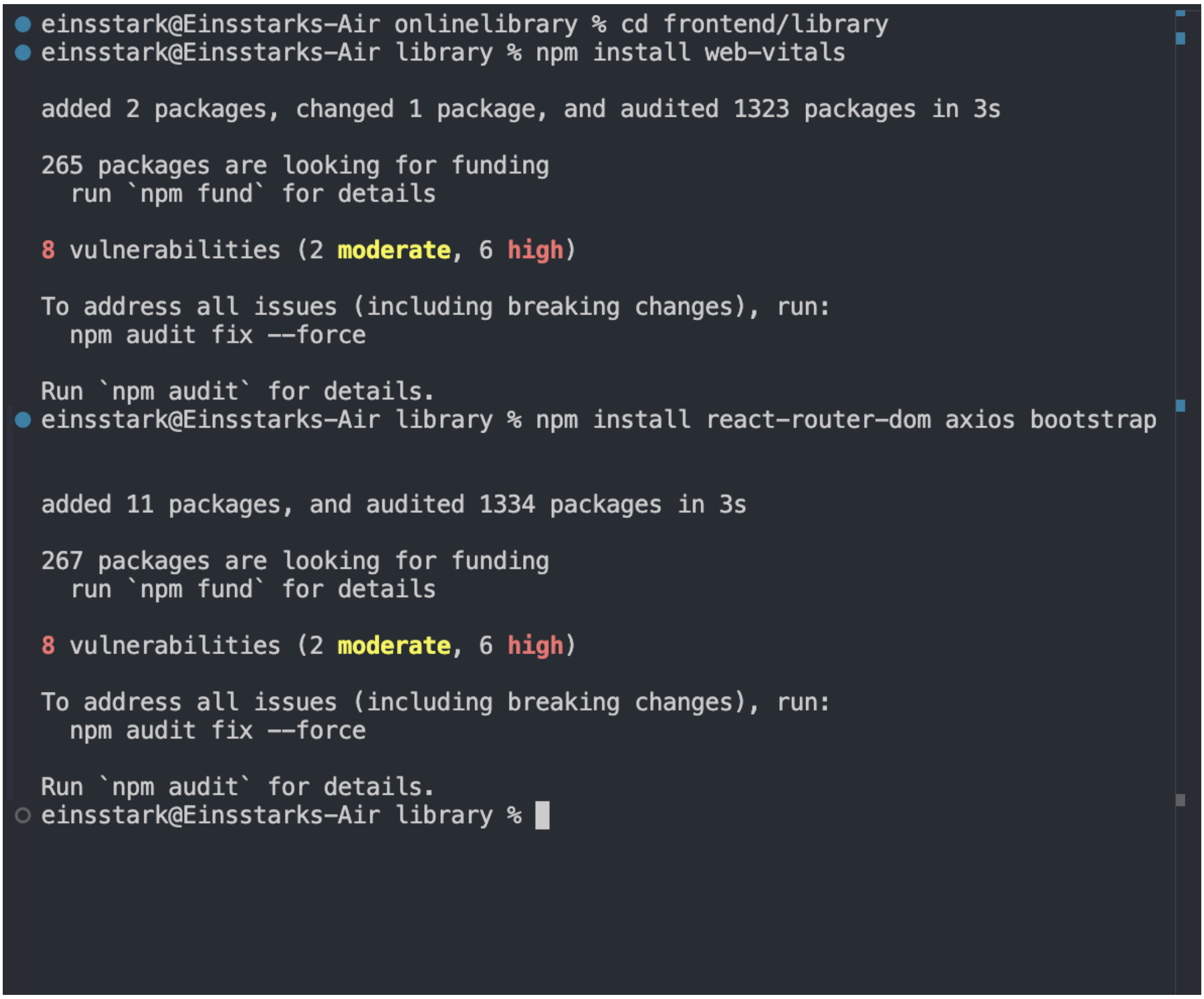


```
einsstark@Einsstarks-Air onlinelibrary % cd frontend/library
einsstark@Einsstarks-Air library % npm install web-vitals

added 2 packages, changed 1 package, and audited 1323 packages in 3s

265 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
einsstark@Einsstarks-Air library % 
```

Edit `src/App.js` and save to reload.

b. Required Packages installing

I installed `react-router-dom`, `axios`, and `bootstrap` using `npm install`. These packages will help with navigation, backend communication, and styling.



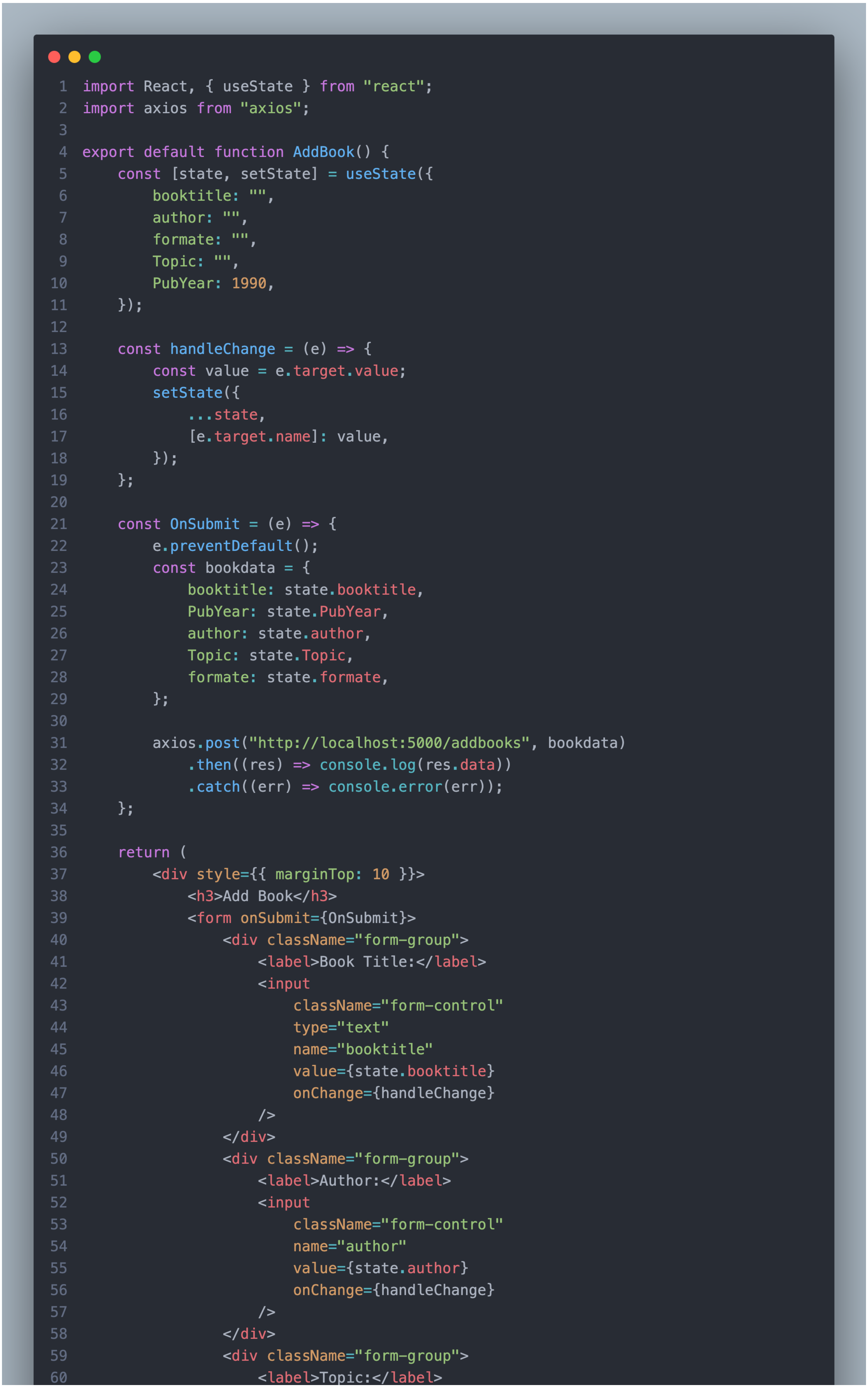
```
einsstark@Einsstarks-Air onlinelibrary % cd frontend/library
einsstark@Einsstarks-Air library % npm install web-vitals

added 2 packages, changed 1 package, and audited 1323 packages in 3s

265 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
einsstark@Einsstarks-Air library % npm install react-router-dom axios bootstrap


added 11 packages, and audited 1334 packages in 3s

267 packages are looking for funding
  run `npm fund` for details

8 vulnerabilities (2 moderate, 6 high)

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
einsstark@Einsstarks-Air library % 
```

c. Components Folder

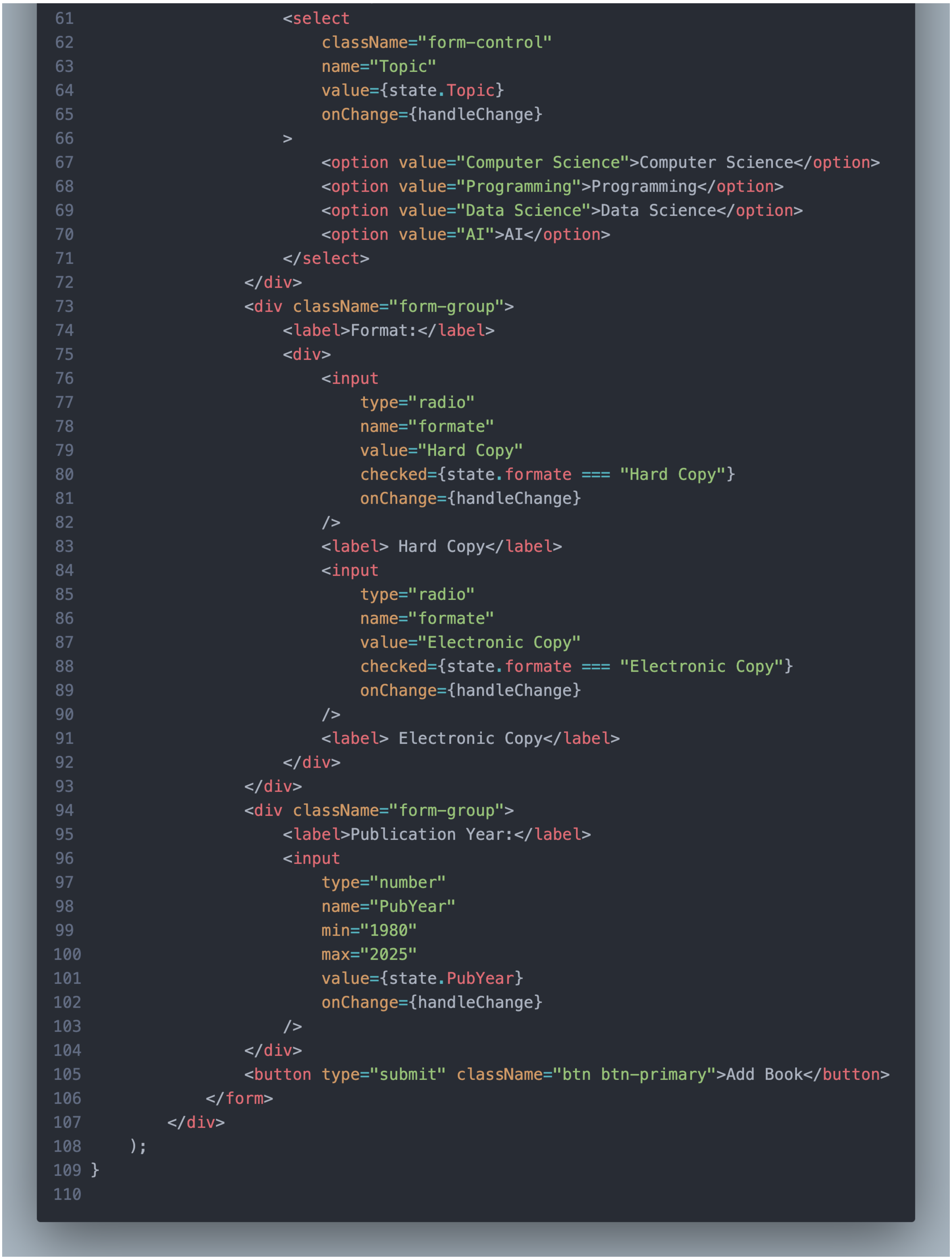
I made a `components` folder in the `src` directory to organize all of my js files i will create in next steps.

1. AddBook.js

This component handles adding a book to the library.



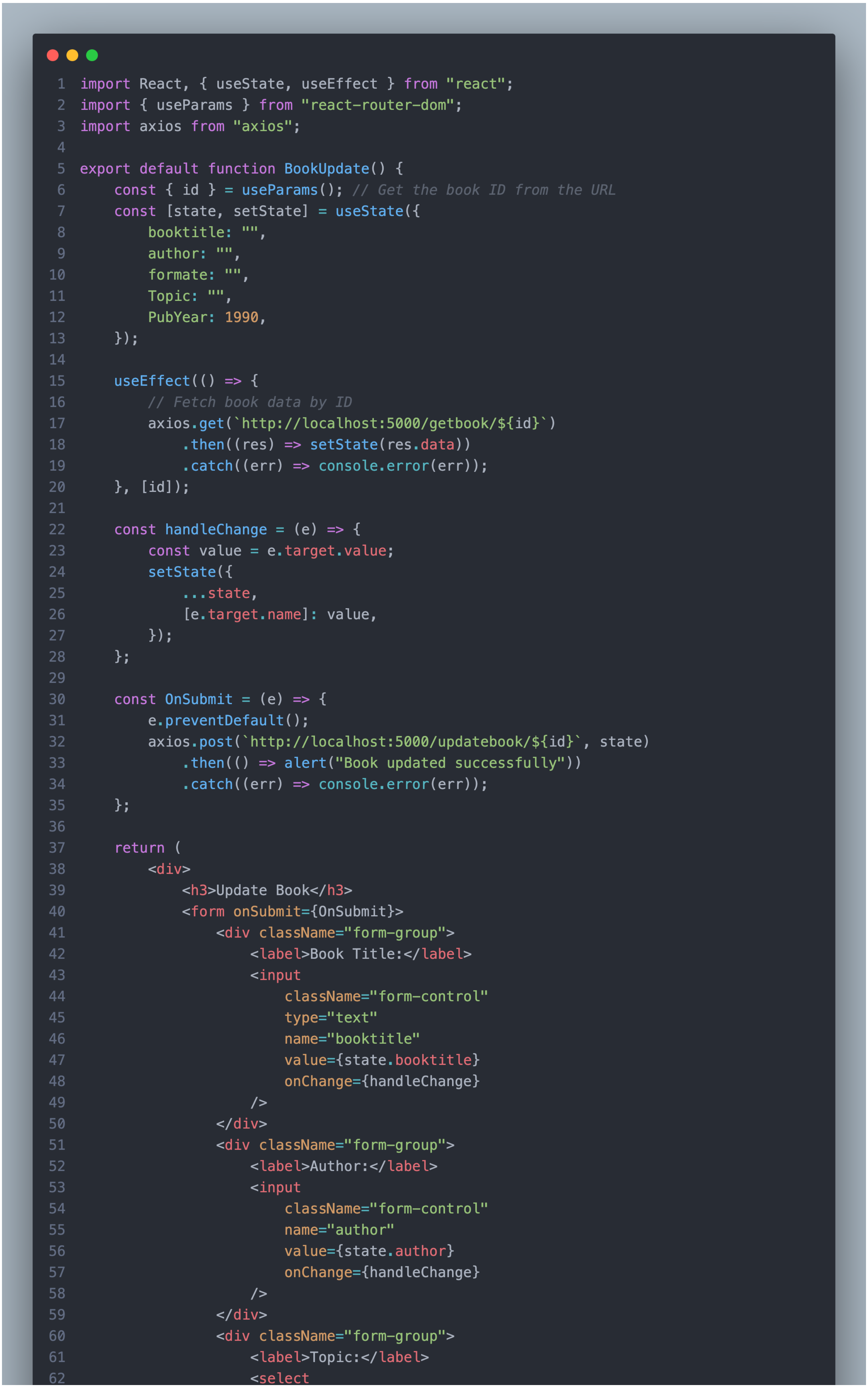
```
import React, { useState } from "react";
import axios from "axios";

export default function AddBook() {
    const [state, setState] = useState({
        booktitle: "",
        author: "",
        formate: "",
        Topic: "",
        PubYear: 1990,
    });

    const handleChange = (e) => {
        const value = e.target.value;
        setState({
            ...state,
            [e.target.name]: value,
        });
    };

    const OnSubmit = (e) => {
        e.preventDefault();
        const bookdata = {
            booktitle: state.booktitle,
            PubYear: state.PubYear,
            author: state.author,
            Topic: state.Topic,
            formate: state.formate,
        };

        axios.post("http://localhost:5000/addbooks", bookdata)
            .then((res) => console.log(res.data))
            .catch((err) => console.error(err));
    };

    return (
        <div style={{ marginTop: 10 }}>
            <h3>Add Book</h3>
            <form onSubmit={OnSubmit}>
                <div className="form-group">
                    <label>Book Title:</label>
                    <input
                        className="form-control"
                        type="text"
                        name="booktitle"
                        value={state.booktitle}
                        onChange={handleChange}
                    />
                </div>
                <div className="form-group">
                    <label>Author:</label>
                    <input
                        className="form-control"
                        name="author"
                        value={state.author}
                        onChange={handleChange}
                    />
                </div>
                <div className="form-group">
                    <label>Topic:</label>
```



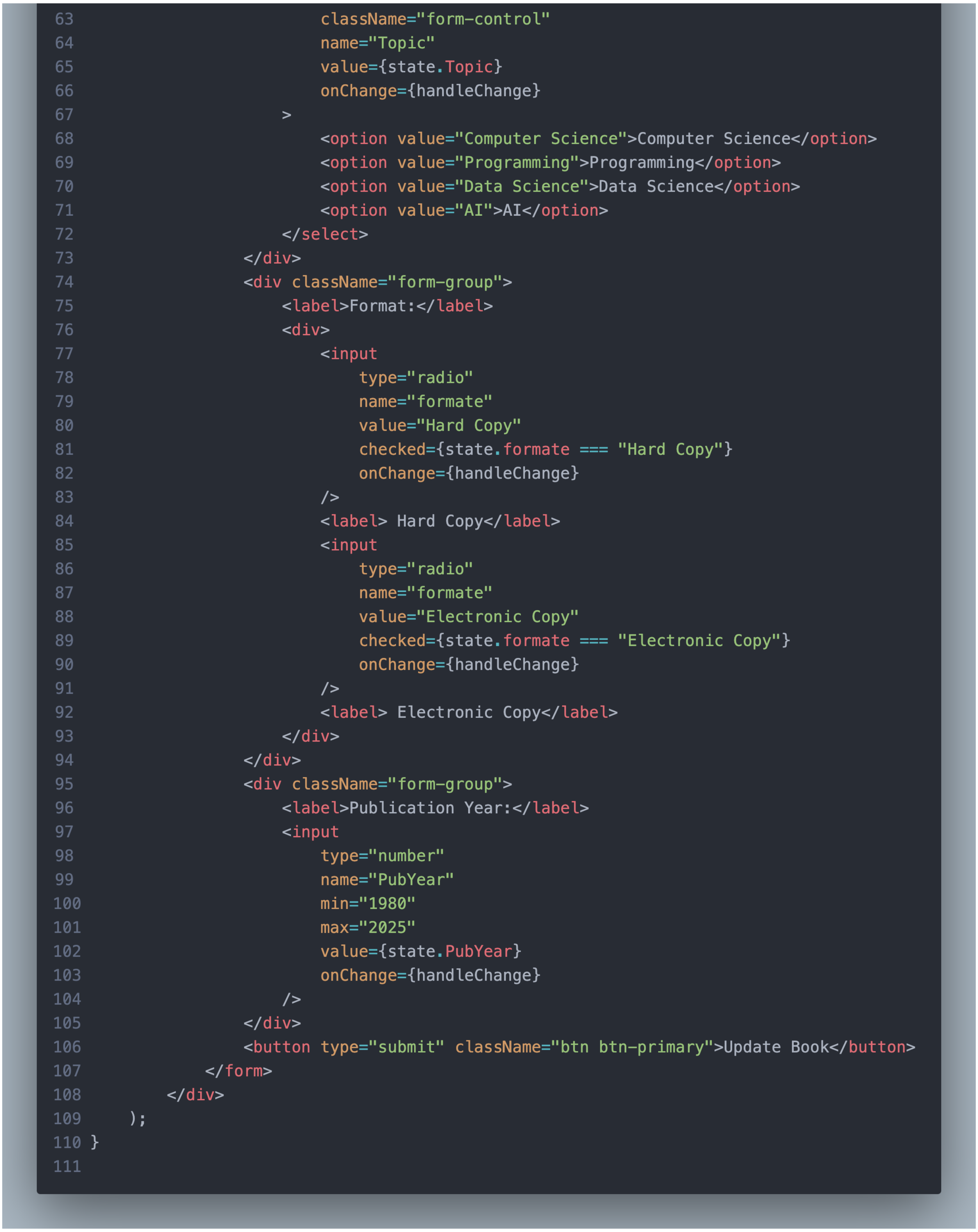
```
                    <select
                        className="form-control"
                        name="Topic"
                        value={state.Topic}
                        onChange={handleChange}
                    >
                        <option value="Computer Science">Computer Science</option>
                        <option value="Programming">Programming</option>
                        <option value="Data Science">Data Science</option>
                        <option value="AI">AI</option>
                    </select>
                </div>
                <div className="form-group">
                    <label>Format:</label>
                    <div>
                        <input
                            type="radio"
                            name="formate"
                            value="Hard Copy"
                            checked={state.formate === "Hard Copy"}
                            onChange={handleChange}
                        />
                        <label> Hard Copy</label>
                        <input
                            type="radio"
                            name="formate"
                            value="Electronic Copy"
                            checked={state.formate === "Electronic Copy"}
                            onChange={handleChange}
                        />
                        <label> Electronic Copy</label>
                    </div>
                </div>
                <div className="form-group">
                    <label>Publication Year:</label>
                    <input
                        type="number"
                        name="PubYear"
                        min="1980"
                        max="2025"
                        value={state.PubYear}
                        onChange={handleChange}
                    />
                </div>
                <button type="submit" className="btn btn-primary">Add Book</button>
            </form>
        </div>
    );
}

```

2. BookUpdate.js

This component allows updating book details based on the book's ID.



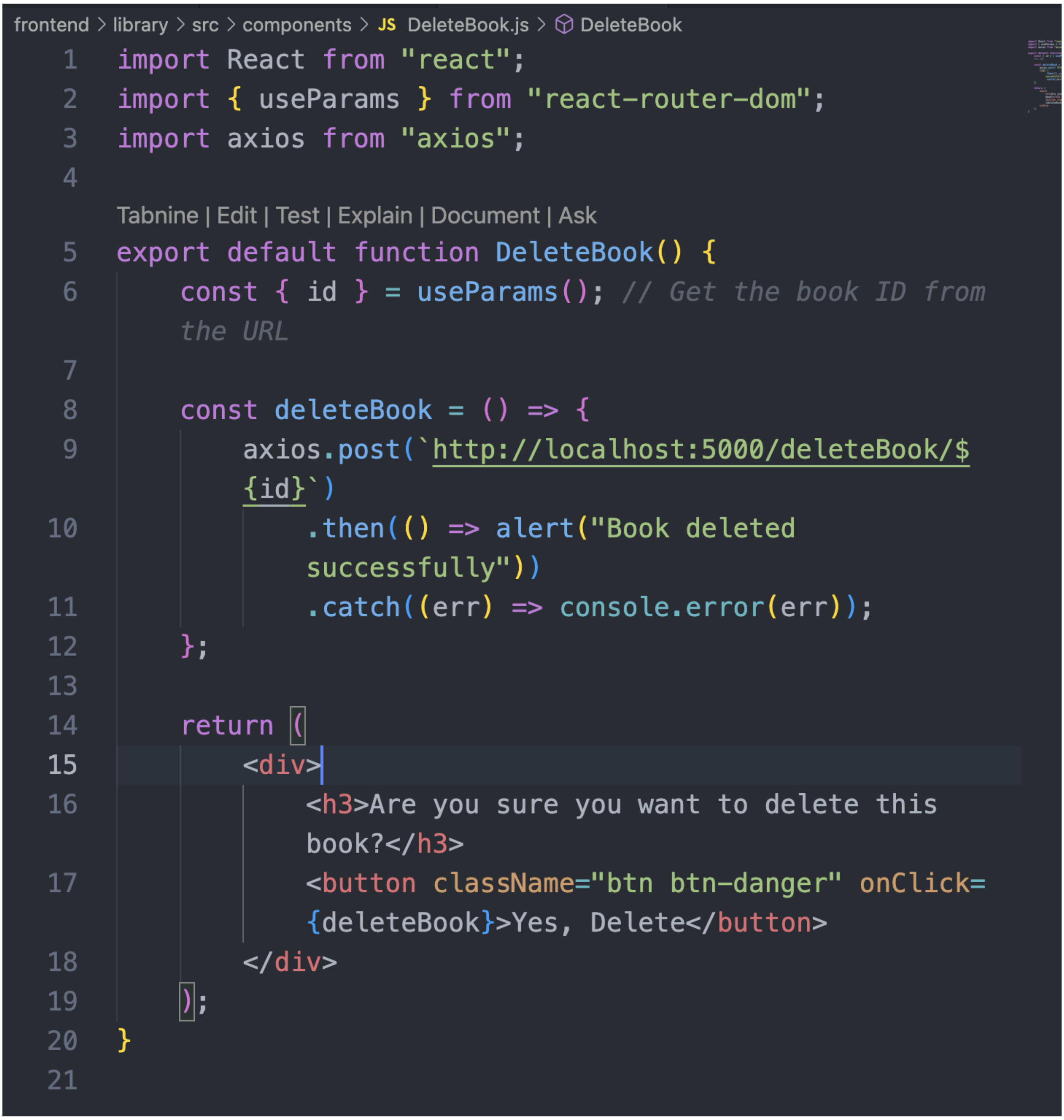
```
import React, { useState, useEffect } from "react";
import { useParams } from "react-router-dom";
import axios from "axios";

export default function BookUpdate() {
    const { id } = useParams(); // Get the book ID from the URL
    const [state, setState] = useState({
        booktitle: "",
        author: "",
        formate: "",
        Topic: "",
        PubYear: 1990,
    });

    useEffect(() => {
        // Fetch book data by ID
        axios.get(`http://localhost:5000/getbook/${id}`)
            .then((res) => setState(res.data))
            .catch((err) => console.error(err));
    }, [id]);

    const handleChange = (e) => {
        const value = e.target.value;
        setState({
            ...state,
            [e.target.name]: value,
        });
    };

    const OnSubmit = (e) => {
        e.preventDefault();
        axios.post(`http://localhost:5000/updatebook/${id}`, state)
            .then(() => alert("Book updated successfully"))
            .catch((err) => console.error(err));
    };

    return (
        <div>
            <h3>Update Book</h3>
            <form onSubmit={OnSubmit}>
                <div className="form-group">
                    <label>Book Title:</label>
                    <input
                        className="form-control"
                        type="text"
                        name="booktitle"
                        value={state.booktitle}
                        onChange={handleChange}
                    />
                </div>
                <div className="form-group">
                    <label>Author:</label>
                    <input
                        className="form-control"
                        name="author"
                        value={state.author}
                        onChange={handleChange}
                    />
                </div>
                <div className="form-group">
                    <label>Topic:</label>
                    <select
```



```
                        className="form-control"
                        name="Topic"
                        value={state.Topic}
                        onChange={handleChange}
                    >
                        <option value="Computer Science">Computer Science</option>
                        <option value="Programming">Programming</option>
                        <option value="Data Science">Data Science</option>
                        <option value="AI">AI</option>
                    </select>
                </div>
                <div className="form-group">
                    <label>Format:</label>
                    <div>
                        <input
                            type="radio"
                            name="formate"
                            value="Hard Copy"
                            checked={state.formate === "Hard Copy"}
                            onChange={handleChange}
                        />
                        <label> Hard Copy</label>
                        <input
                            type="radio"
                            name="formate"
                            value="Electronic Copy"
                            checked={state.formate === "Electronic Copy"}
                            onChange={handleChange}
                        />
                        <label> Electronic Copy</label>
                    </div>
                </div>
                <div className="form-group">
                    <label>Publication Year:</label>
                    <input
                        type="number"
                        name="PubYear"
                        min="1980"
                        max="2025"
                        value={state.PubYear}
                        onChange={handleChange}
                    />
                </div>
                <button type="submit" className="btn btn-primary">Update Book</button>
            </form>
        </div>
    );
}

```

3. DeleteBook.js

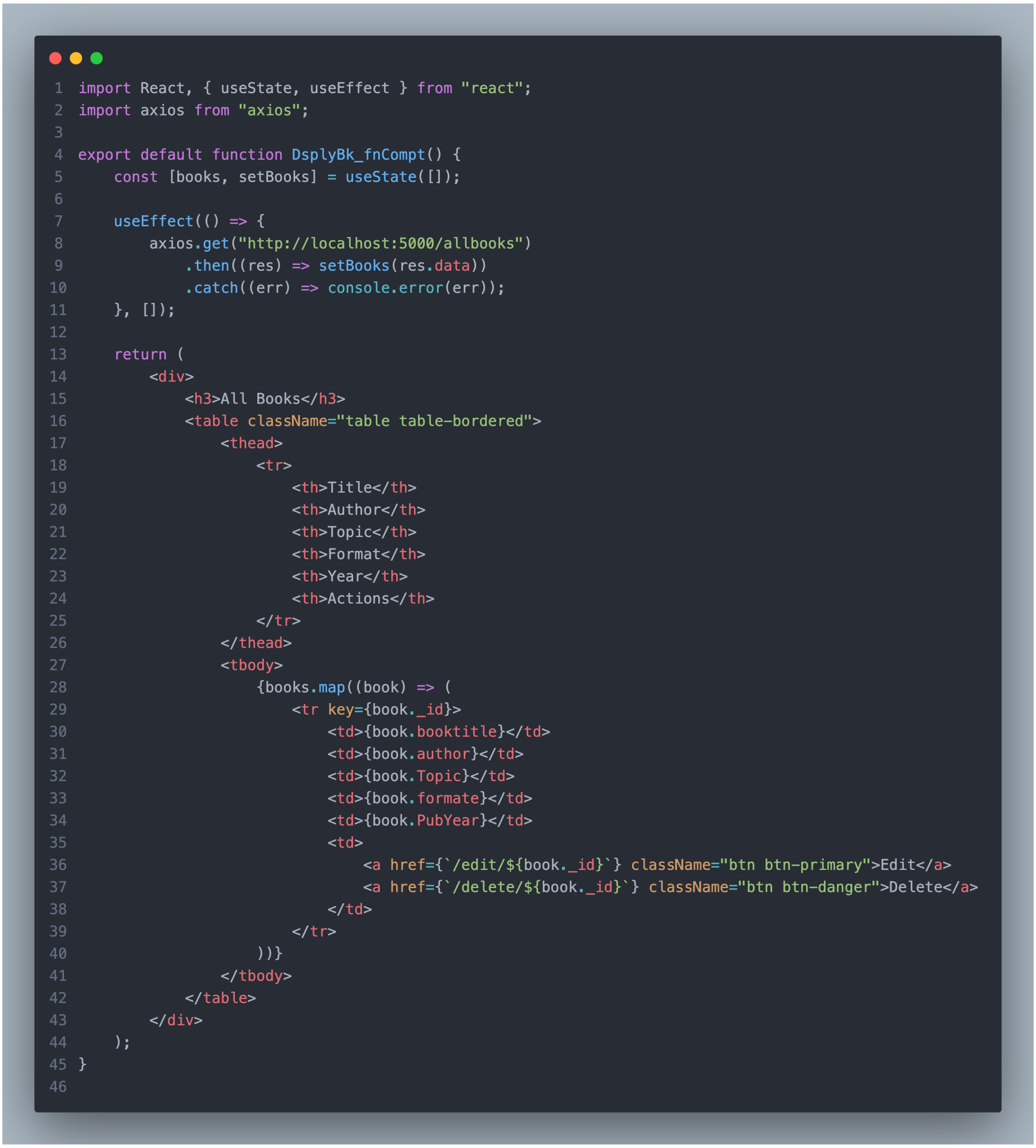
This component handles deleting a book by its ID.


frontend > library > src > components > JS DeleteBook.js > DeleteBook
Tabnine | Edit | Test | Explain | Document | Ask

```javascript
import React from "react";
import { useParams } from "react-router-dom";
import axios from "axios";

export default function DeleteBook() {
    const { id } = useParams(); // Get the book ID from the URL

    const deleteBook = () => {
        axios.post(`http://localhost:5000/deleteBook/${id}`)
            .then(() => alert("Book deleted successfully"))
            .catch((err) => console.error(err));
    };

    return (
        <div>
            <h3>Are you sure you want to delete this book?</h3>
            <button className="btn btn-danger" onClick={deleteBook}>Yes, Delete</button>
        </div>
    );
}
```

4. DsplyBk_fnCompt.js

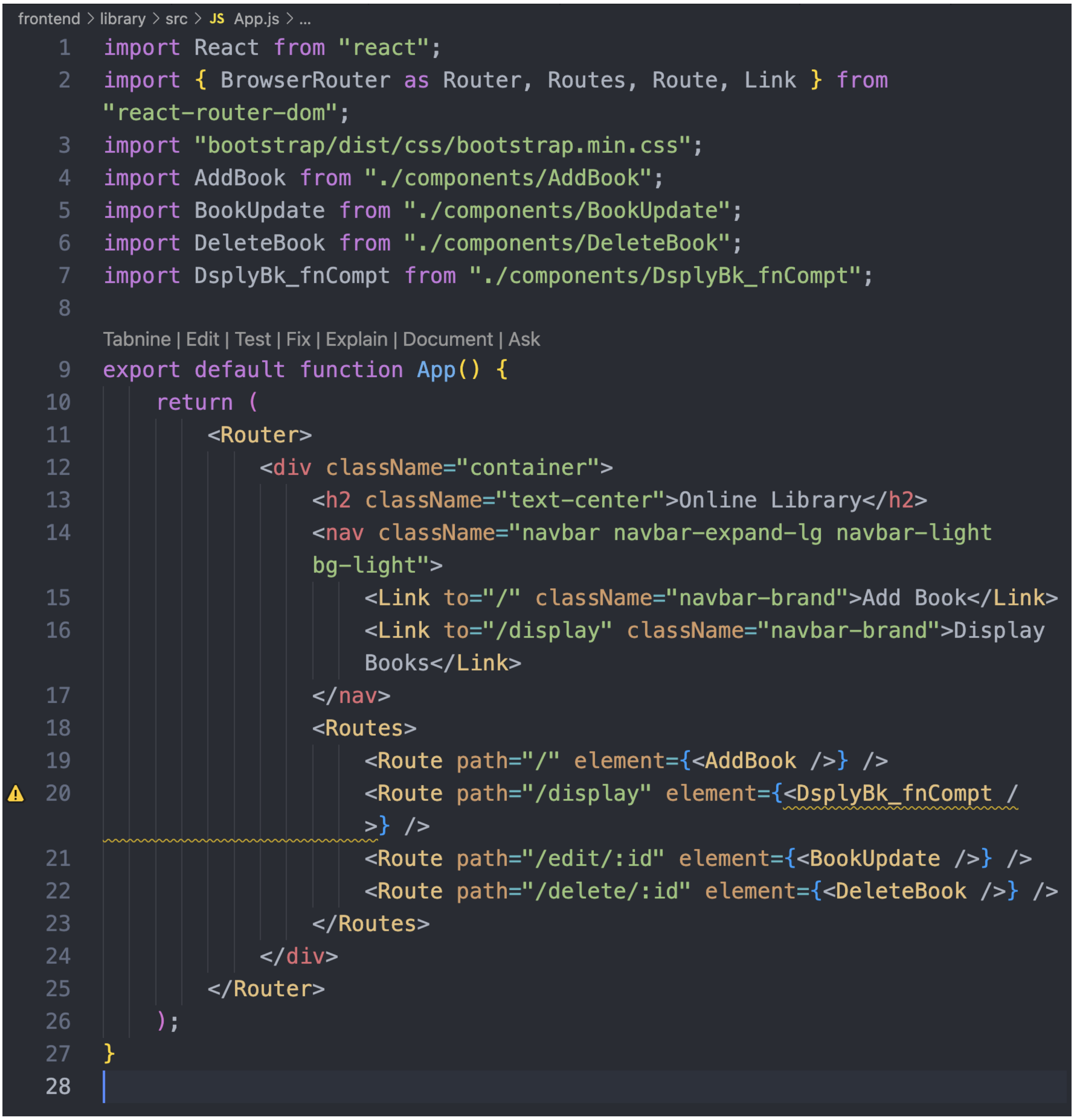
This component fetches and displays all books from the backend.



```
import React, { useState, useEffect } from "react";
import axios from "axios";

export default function DsplyBk_fnCompt() {
    const [books, setBooks] = useState([]);

    useEffect(() => {
        axios.get("http://localhost:5000/allbooks")
            .then((res) => setBooks(res.data))
            .catch((err) => console.error(err));
    }, []);

    return (
        <div>
            <h3>All Books</h3>
            <table className="table table-bordered">
                <thead>
                    <tr>
                        <th>Title</th>
                        <th>Author</th>
                        <th>Topic</th>
                        <th>Format</th>
                        <th>Year</th>
                        <th>Actions</th>
                    </tr>
                </thead>
                <tbody>
                    {books.map((book) => (
                        <tr key={book._id}>
                            <td>{book.booktitle}</td>
                            <td>{book.author}</td>
                            <td>{book.Topic}</td>
                            <td>{book.formate}</td>
                            <td>{book.PubYear}</td>
                            <td>
                                <a href={`/edit/${book._id}`} className="btn btn-primary">Edit</a>
                                <a href={`/delete/${book._id}`} className="btn btn-danger">Delete</a>
                            </td>
                        </tr>
                    ))}
                </tbody>
            </table>
        </div>
    );
}

```

d. Routing in App.js

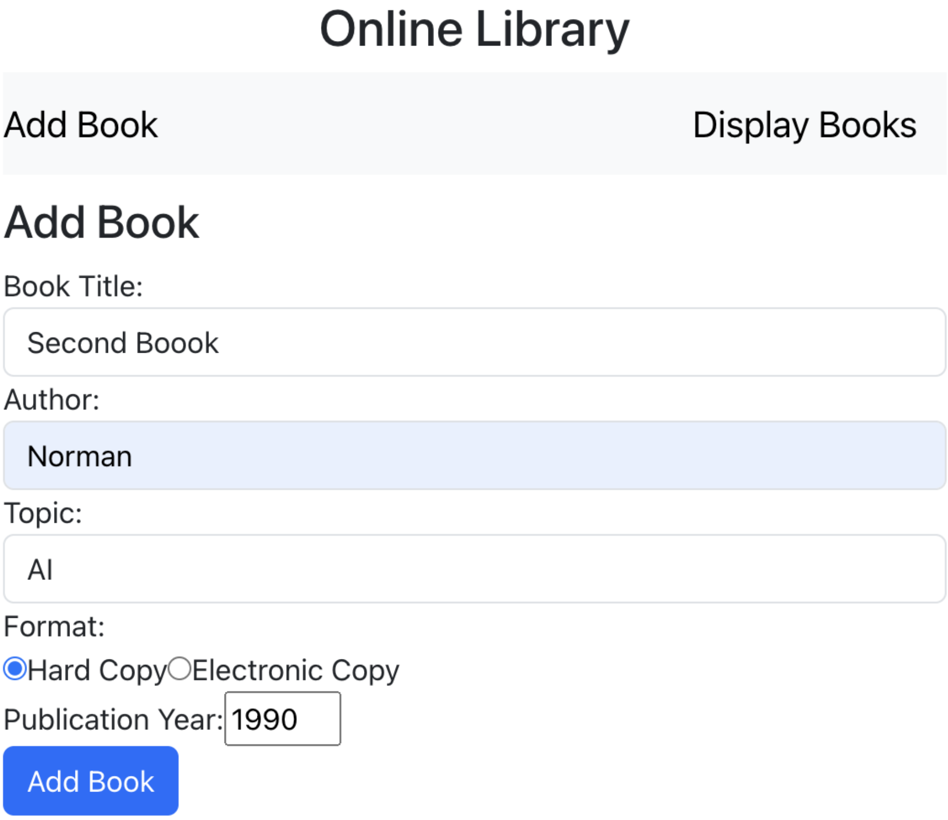
Used React Router in `App.js` to connect components. I added `<Link>` for navigation and `<Route>` for showing different pages.



```
import React from "react";
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
import "bootstrap/dist/css/bootstrap.min.css";
import AddBook from "./components/AddBook";
import BookUpdate from "./components/BookUpdate";
import DeleteBook from "./components/DeleteBook";
import DsplyBk_fnCompt from "./components/DsplyBk_fnCompt";

export default function App() {
    return (
        <Router>
            <div className="container">
                <h2 className="text-center">Online Library</h2>
                <nav className="navbar navbar-expand-lg navbar-light bg-light">
                    <Link to="/" className="navbar-brand">Add Book</Link>
                    <Link to="/display" className="navbar-brand">Display Books</Link>
                </nav>
                <Routes>
                    <Route path="/" element={<AddBook />} />
                    <Route path="/display" element={<DsplyBk_fnCompt />} />
                    <Route path="/edit/:id" element={<BookUpdate />} />
                    <Route path="/delete/:id" element={<DeleteBook />} />
                </Routes>
            </div>
        </Router>
    );
}
```

Testing

I ran `npm start` again and tested all features.



Online Library

Add Book Display Books

Add Book

Book Title: Second Boook

Author: Norman

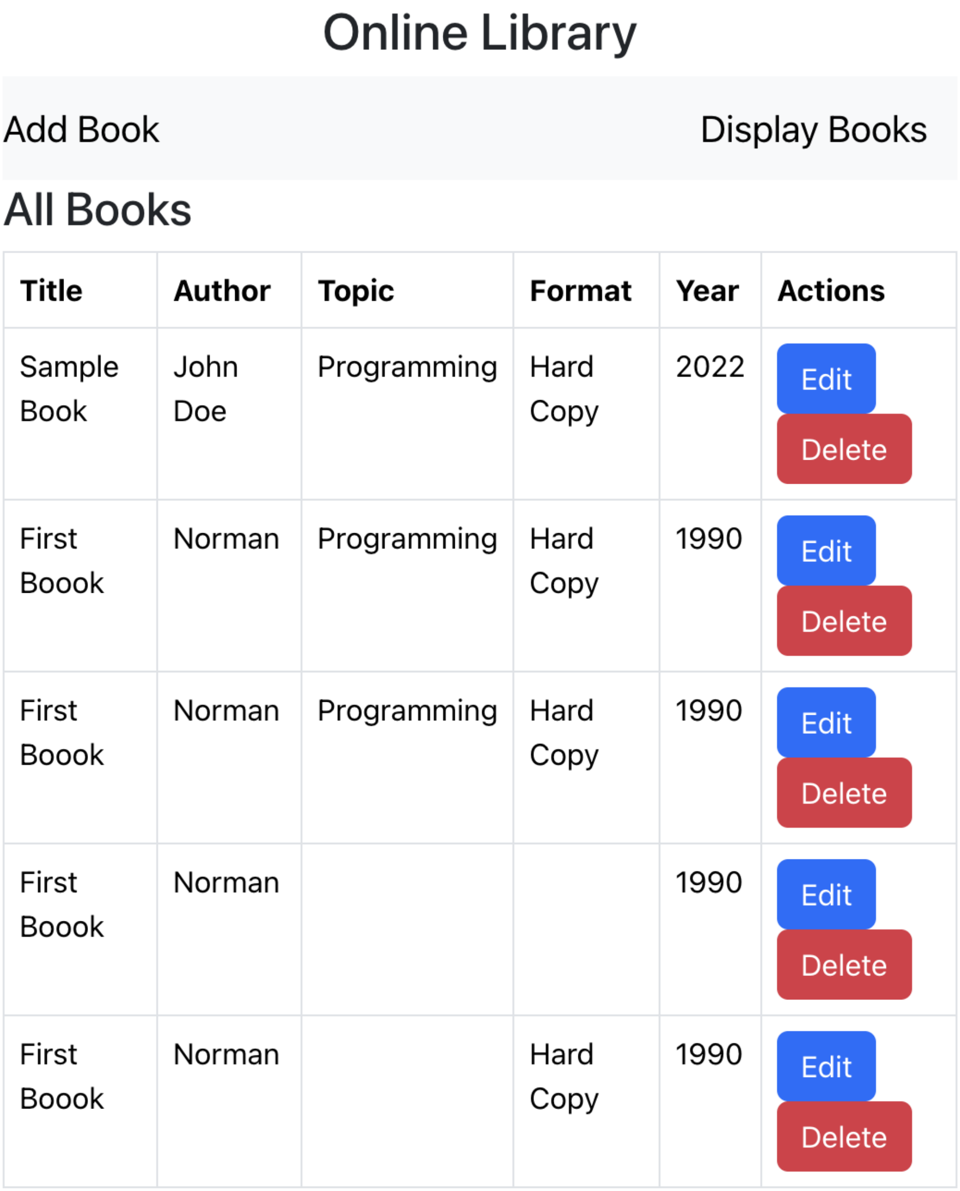
Topic: AI

Format:

● Hard Copy ○ Electronic Copy

Publication Year: 1990

Adding the book and displaying it.



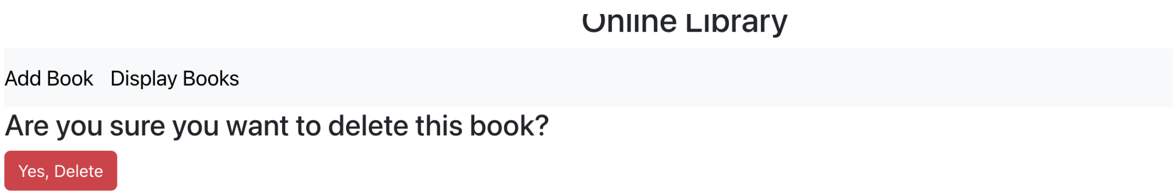
Online Library

Add Book Display Books

All Books

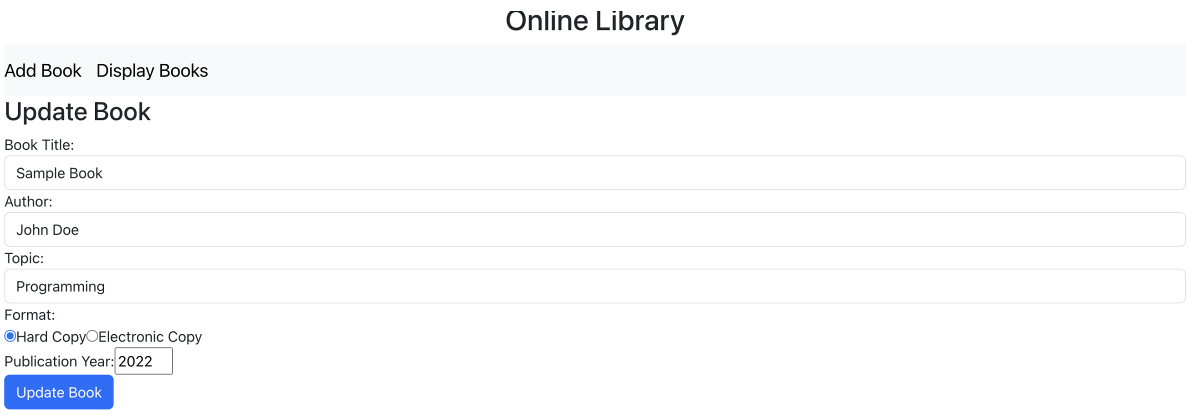
Title	Author	Topic	Format	Year	Actions
Sample Book	John Doe	Programming	Hard Copy	2022	Edit Delete
First Boook	Norman	Programming	Hard Copy	1990	Edit Delete
First Boook	Norman	Programming	Hard Copy	1990	Edit Delete
First Boook	Norman			1990	Edit Delete
First Boook	Norman		Hard Copy	1990	Edit Delete

Deleting the book:



Are you sure you want to delete this book?

Updating the book:



Update Book

Book Title: Sample Book

Author: John Doe

Topic: Programming

Format: ● Hard Copy ○ Electronic Copy

Publication Year: 2022

2. Reflection

Reflection

Building the online library project was both challenging and rewarding. On the back end, I struggled initially with setting up the MongoDB database and connecting it to the server using Mongoose. Understanding how schemas work and ensuring the data structure matched the inputs from the front end took time. Another difficulty was debugging the routes when the `POST` requests weren't working. I solved this by using tools like Postman to test endpoints and adding `console.log` statements in my backend code to trace the issues and gave it to chatgpt for solutions.

On the front end, learning React felt overwhelming at first (during week 6, 7) because managing state with hooks like `useState` and `useEffect` was new to me. It was tricky to fetch data from the backend and render it dynamically in components like

`DsplyBk_fnCompt`. The app wasn’t displaying books at one point, but by logging data in the browser console and double-checking the Axios requests, I was able to identify and fix the issues. Another challenge was setting up React Router, but once I understood how routes and components connect, it became clear.

Despite these difficulties, working with React, MongoDB, and REST APIs gave me a solid understanding of how full-stack development works. These skills now make me feel confident about building more complex and functional applications in the future.