**Project Outline: Online Bookstore**

**Features**

1. **User Authentication**: Users can sign up, log in, and log out.
2. **Book Management**: Admins can add, update, and delete books.
3. **Book Browsing**: Users can browse and search for books.
4. **Cart Management**: Users can add books to their cart and view their cart.
5. **Order Management**: Users can place orders, and admins can manage orders.

**Tech Stack**

* **Frontend**: React
* **Backend**: Node.js with Express.js
* **Database**: MySQL or PostgreSQL

**Steps to Implement**

**1. Setup the Project**

**Frontend (React)**:

* Create a new React project using create-react-app.

bash

Copy code

npx create-react-app online-bookstore

cd online-bookstore

**Backend (Node.js)**:

* Create a new Node.js project.

bash

Copy code

mkdir backend

cd backend

npm init -y

* Install the necessary packages.

bash

Copy code

npm install express mysql2 sequelize cors dotenv

**2. Database Setup**

* Create a database in MySQL/PostgreSQL.

sql

Copy code

CREATE DATABASE online\_bookstore;

* Create tables for users, books, carts, and orders.

sql

Copy code

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

password VARCHAR(100)

);

CREATE TABLE books (

id INT AUTO\_INCREMENT PRIMARY KEY,

title VARCHAR(255),

author VARCHAR(255),

price DECIMAL(10, 2),

description TEXT

);

CREATE TABLE carts (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

book\_id INT,

quantity INT,

FOREIGN KEY (user\_id) REFERENCES users(id),

FOREIGN KEY (book\_id) REFERENCES books(id)

);

CREATE TABLE orders (

id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

total\_price DECIMAL(10, 2),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id)

);

**3. Backend Implementation**

* Create an Express server and setup Sequelize ORM for database interactions.

**backend/index.js**:

* Set up an Express server.
* Connect to the database using Sequelize.
* Set up basic middlewares like cors and express.json.

**backend/config/database.js**:

* Configure the Sequelize instance to connect to your SQL database using environment variables.

**backend/models/User.js**:

* Define the User model with fields: name, email, and password.
* Implement APIs for user authentication, book management, cart management, and order management.

**Routes to implement**:

* /api/users/register: Register a new user.
* /api/users/login: Log in a user and return a JWT.
* /api/books: CRUD operations for books.
* /api/cart: CRUD operations for cart items.
* /api/orders: Manage orders.

**4. Frontend Implementation**

* Create React components for user authentication, book browsing, cart management, and order management.

**src/App.js**:

* Set up React Router for navigation between different pages (Home, Login, Register, Books, Cart, Orders).

**Components to implement**:

* Login: Form for users to log in.
* Register: Form for users to register.
* Books: Display a list of books with search functionality.
* Cart: Display user's cart items.
* Orders: Display user's past orders.

**API calls**:

* Use axios or fetch to make API requests to the backend.
* Handle user authentication by storing the JWT token in local storage or React context.

**5. Run the Application**

* Start the backend server.

bash

Copy code

cd backend

node index.js