GDGoC Web Dev Bootcamp 2024

Assignment Day 02

Enhance Library Manager with JSX, Components, and Props

Description:

Extend the Library Manager project from Assignment 01 to utilize JSX, reusable components, and props for a dynamic UI.

Task Details:

1. Create a new folder named Assignment-02.
2. Create a React project using Vite.
    - Your App.js will be your top-most parent component that will serve the same purpose as your manager.js file in the first assignment.
    - Create another component named BookCard.jsx.

Step 01 - Setup Books Data:

• In books.js, create an array of objects named Books with the following data:

    - id (number 1-20)
    - title (string)
    - author (string)
    - rating (number 1-5)
    - genre (string)

- Export the Books array.

Step 02 - BookCard Component Structure:

- In BookCard.jsx, create a functional component named BookCard that takes props and displays the book's details (title, author, rating, genre) using JSX.
- Use destructuring to extract the book details from props.

Step 03: Create Functional Components

- Create four more components, each for the specified functions in Assignment 1:
    - HighRatedBooks.jsx: Filters and displays books with a rating higher than 4.5.
    - BookDetails.jsx: Iterates over an array of books and prints their details (title, author, rating, genre).
    - FictionBooks.jsx: Filters and displays books with the genre "fiction".
    - BookAuthors.jsx: Displays an array of book authors.

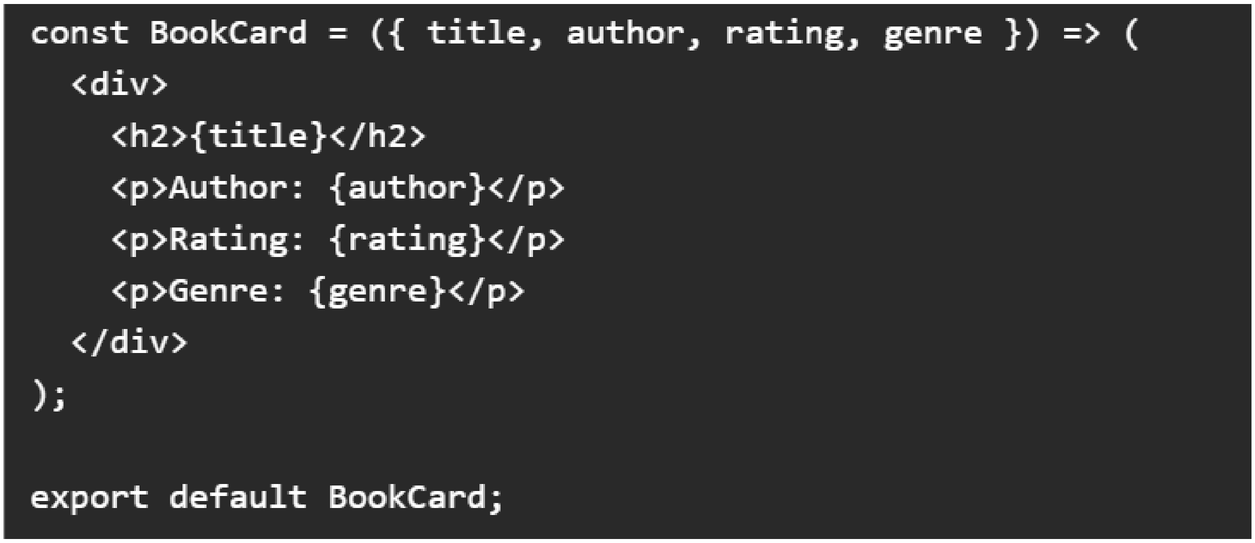
Step 04: Use BookCard as Child Component

- In each of these components, BookCard will be a child component that takes all the same props as described in Step 02.

Step 05: Import Components in App.js

- In App.js, import these four components and display their results.

Example BookCard Component:



```
const BookCard = ({ title, author, rating, genre }) => (
  <div>
    <h2>{title}</h2>
    <p>Author: {author}</p>
    <p>Rating: {rating}</p>
    <p>Genre: {genre}</p>
  </div>
);

export default BookCard;
```