

## **Day 8 - “BookVerse: Author Details and Styling the Components”**

### **1. Project Overview**

#### **Project Title:**

**Day 8 - BookVerse: Author Details and Styling the Components**

#### **Objective:**

The objective of Day 8 is to enhance the BookVerse React application by enabling users to click on a book and view detailed information about its author. This activity focuses on component composition, class components, lifecycle methods, refs (uncontrolled components), PropTypes validation, and styling using Bootstrap and CSS.

### **2. User Story**

#### **As a user,**

I want to click on a book and view its author details,  
So that I can learn more about the author and their similar works.

### **3. Key Features Implemented**

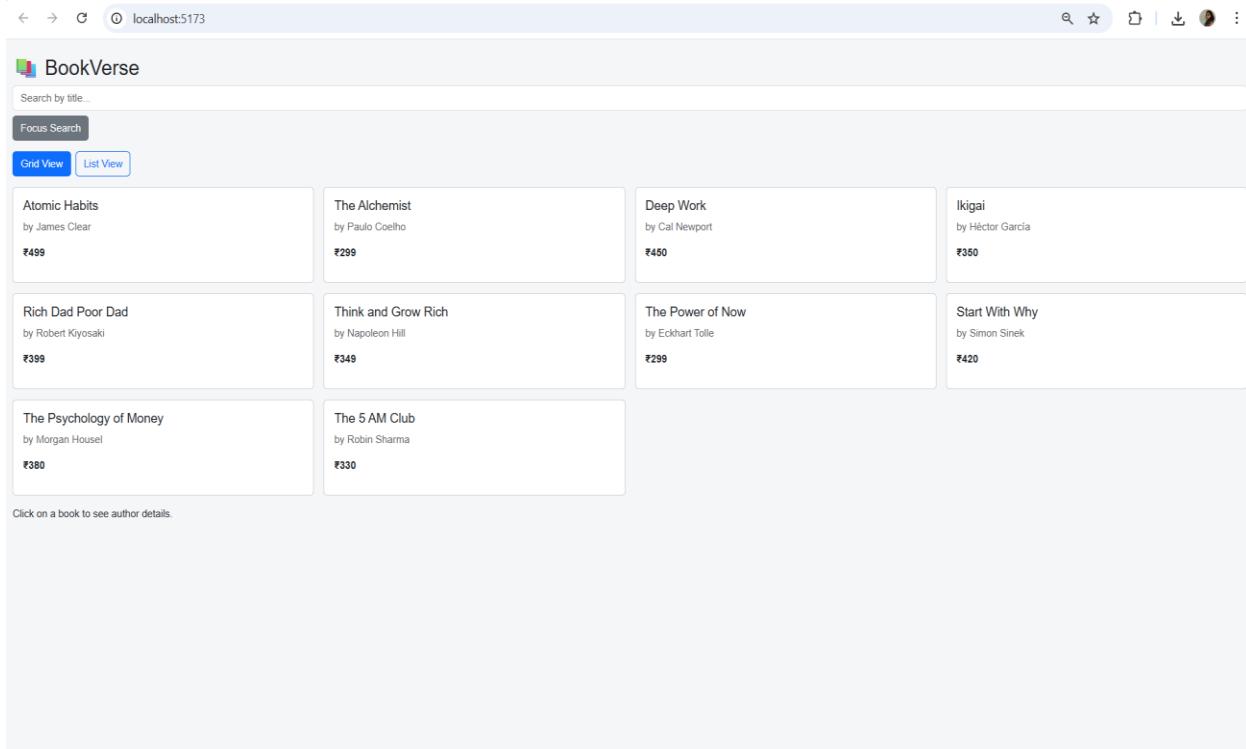
- Author details displayed using a **class component**
- Author biography and top three books shown dynamically
- **Refs (Uncontrolled Components)** used to focus the search input field
- **PropTypes** used to validate component props
- **Lifecycle methods** demonstrated using componentDidMount
- **Bootstrap** used for layout, cards, and grid styling
- **Component composition** using BookList, BookCard, AuthorInfo, and SearchBox
- Grid and List view toggle implemented using state

### **4. Technologies Used**

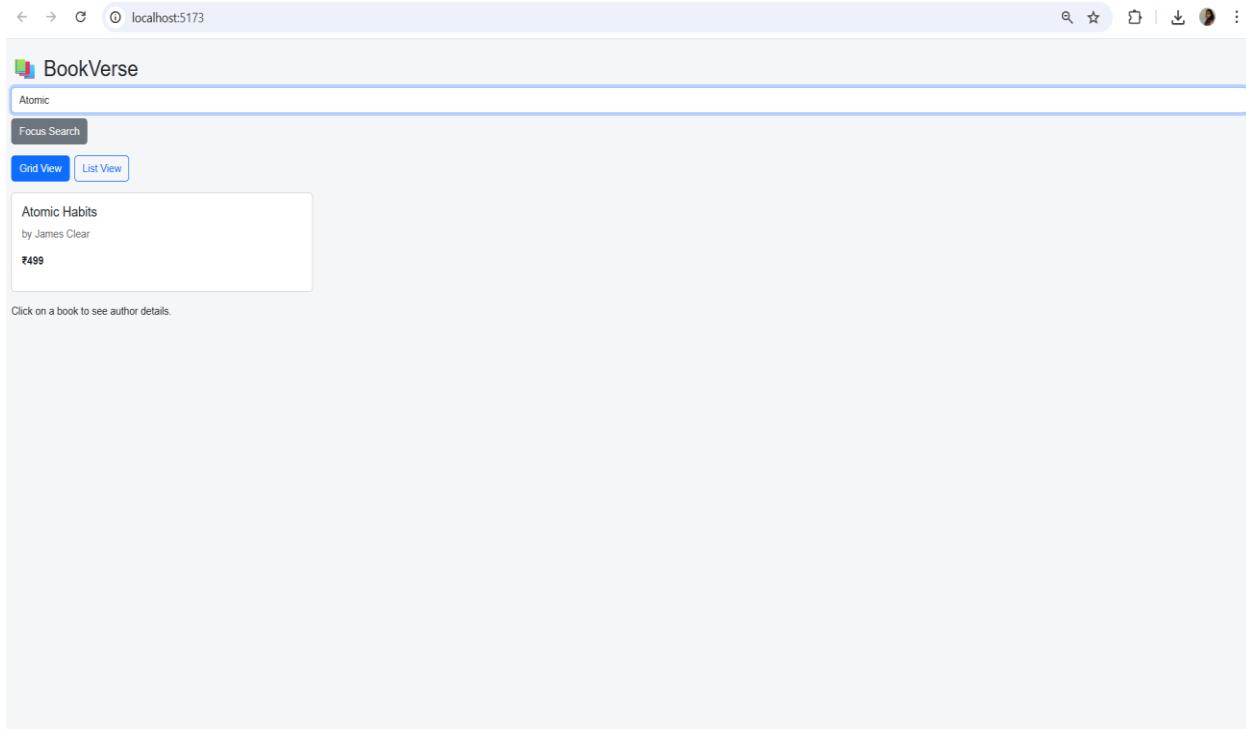
- React (Vite)
- JavaScript (ES6)
- Bootstrap
- CSS
- PropTypes

## 5. Application Output Screenshots

### Screenshot 1: Home Page



### Screenshot 2: Focus Search Button Functionality



**Screenshot 3:** Author Details Section after Clicking a Book

The screenshot shows a web application interface for "BookVerse". At the top, there is a header bar with a search bar, a "Focus Search" button, and navigation icons. Below the header is a grid view of books. The books listed are:

| Atomic Habits<br>by James Clear<br>₹499             | The Alchemist<br>by Paulo Coelho<br>₹299        | Deep Work<br>by Cal Newport<br>₹450          | Ikigai<br>by Héctor García<br>₹350       |
|---|---|--|--|
| Rich Dad Poor Dad<br>by Robert Kiyosaki<br>₹399     | Think and Grow Rich<br>by Napoleon Hill<br>₹349 | The Power of Now<br>by Eckhart Tolle<br>₹299 | Start With Why<br>by Simon Sinek<br>₹420 |
| The Psychology of Money<br>by Morgan Housel<br>₹380 | The 5 AM Club<br>by Robin Sharma<br>₹330        |  |  |

Below the grid, a sidebar titled "Author Details" displays information for Cal Newport. It includes his bio, which states he is a professor and author who writes about productivity, focus, and meaningful work. It also lists his top 3 books: Deep Work, Digital Minimalism, and So Good They Can't Ignore You.

## 6. Components

### 6.1 App.jsx

```
import { useState } from "react";

import booksData from "./data/booksData";
import authorsData from "./data/authorsData";
import BookList from "./components/BookList";
import SearchBox from "./components/SearchBox";
import AuthorInfo from "./components/AuthorInfo";

function App() {

  const [search, setSearch] = useState("");
  const [view, setView] = useState("grid");
  const [selectedAuthor, setSelectedAuthor] = useState(null);

  return (
    <div>
      <SearchBox search={search} setSearch={setSearch}></SearchBox>
      <BookList view={view} books={booksData} selectedAuthor={selectedAuthor} setSelectedAuthor={setSelectedAuthor}></BookList>
      {selectedAuthor ? <AuthorInfo author={authorsData[selectedAuthor]}></AuthorInfo> : null}
    </div>
  );
}

export default App;
```

```
const filteredBooks = booksData.filter((book) =>
  book.title.toLowerCase().includes(search.toLowerCase())
);

const handleSelectBook = (book) => {
  setSelectedAuthor(authorsData[book.author]);
};

return (
  <div className="container-fluid mt-4">
    <h2>  BookVerse </h2>

    <SearchBox onSearch={setSearch} />

    <div className="mb-3">
      <button
        className={`btn me-2 ${{
          view === "grid" ? "btn-primary" : "btn-outline-primary"
        }}`}
        onClick={() => setView("grid")}
      >
        Grid View
      </button>

      <button
        className={`btn ${{
          view === "list" ? "btn-primary" : "btn-outline-primary"
        }}`}
      >
```

```

    `}`}
    onClick={() => setView("list")}
  >
  List View
</button>
</div>

<BookList
  books={filteredBooks}
  onSelect={handleSelectBook}
  view={view}
/>

<AuthorInfo author={selectedAuthor} />
</div>
);
}

export default App;

```

## 6.2 BookList.jsx

```

import BookCard from "./BookCard";

function BookList({ books, onSelect, view }) {
  if (view === "list") {
    return (
      <ul className="list-group">
        {books.map((book) => (

```

```

<li
  key={book.id}
  className="list-group-item"
  style={{ cursor: "pointer" }}
  onClick={() => onSelect(book)}
>
  <strong>{book.title}</strong> by {book.author} — ₹{book.price}
</li>
)})}
</ul>
);

}

return (
<div className="row g-3">
{books.map((book) => (
<div className="col-md-3" key={book.id}>
<BookCard book={book} onSelect={onSelect} />
</div>
))})
</div>
);
}

export default BookList;

```

### 6.3 BookCard.jsx

```
import PropTypes from "prop-types";
```

```
function BookCard({ book, onSelect }) {  
  return (  
    <div  
      className="card p-3 h-100"  
      style={{ cursor: "pointer" }}  
      onClick={() => onSelect(book)}  
    >  
      <h5>{book.title}</h5>  
      <p className="text-muted">by {book.author}</p>  
      <p className="fw-bold">₹{book.price}</p>  
    </div>  
  );  
}  
  
BookCard.propTypes = {  
  book: PropTypes.shape({  
    title: PropTypes.string.isRequired,  
    author: PropTypes.string.isRequired,  
    price: PropTypes.number.isRequired,  
  }).isRequired,  
  onSelect: PropTypes.func.isRequired,  
};  
  
export default BookCard;
```

#### 6.4 AuthorInfo.jsx

```
import { Component } from "react";
```

```
class AuthorInfo extends Component {  
  componentDidMount() {  
    console.log("Author information loaded");  
  }  
  
  render() {  
    const { author } = this.props;  
  
    if (!author) {  
      return <p className="mt-3">Click on a book to see author details.</p>;  
    }  
  
    return (  
      <div className="card mt-4 p-3">  
        <h4>Author Details</h4>  
        <p><strong>Bio:</strong> {author.bio}</p>  
        <p><strong>Top 3 Books:</strong></p>  
        <ul>  
          {author.topBooks.map((book, index) => (  
            <li key={index}>{book}</li>  
          ))}  
        </ul>  
      </div>  
    );  
  }  
}
```

```
export default AuthorInfo;
```

## 6.5 SearchBox.jsx

```
import { useRef } from "react";
```

```
function SearchBox({ onSearch }) {
```

```
  const inputRef = useRef();
```

```
  return (
```

```
    <div className="mb-3">
```

```
      <input
```

```
        ref={inputRef}
```

```
        type="text"
```

```
        className="form-control"
```

```
        placeholder="Search by title..."
```

```
        onChange={(e) => onSearch(e.target.value)}
```

```
    />
```

```
    <button
```

```
      className="btn btn-secondary mt-2"
```

```
      onClick={() => inputRef.current.focus()}
```

```
    >
```

```
      Focus Search
```

```
    </button>
```

```
  </div>
```

```
);
```

```
}
```

```
export default SearchBox;
```

## 9. Summary

The Day 8 implementation of BookVerse successfully enhances the application by introducing interactive author details, class components, lifecycle methods, refs, and PropTypes validation. The project demonstrates effective component composition and consistent UI styling using Bootstrap.