BookVerse – Day 1 Documentation Report

This report documents the implementation of Day 1 for the React-based project 'BookVerse', focusing on React Basics and Component Fundamentals. It demonstrates React functional components, props, state management, event handling, and controlled components.

1. Project Overview

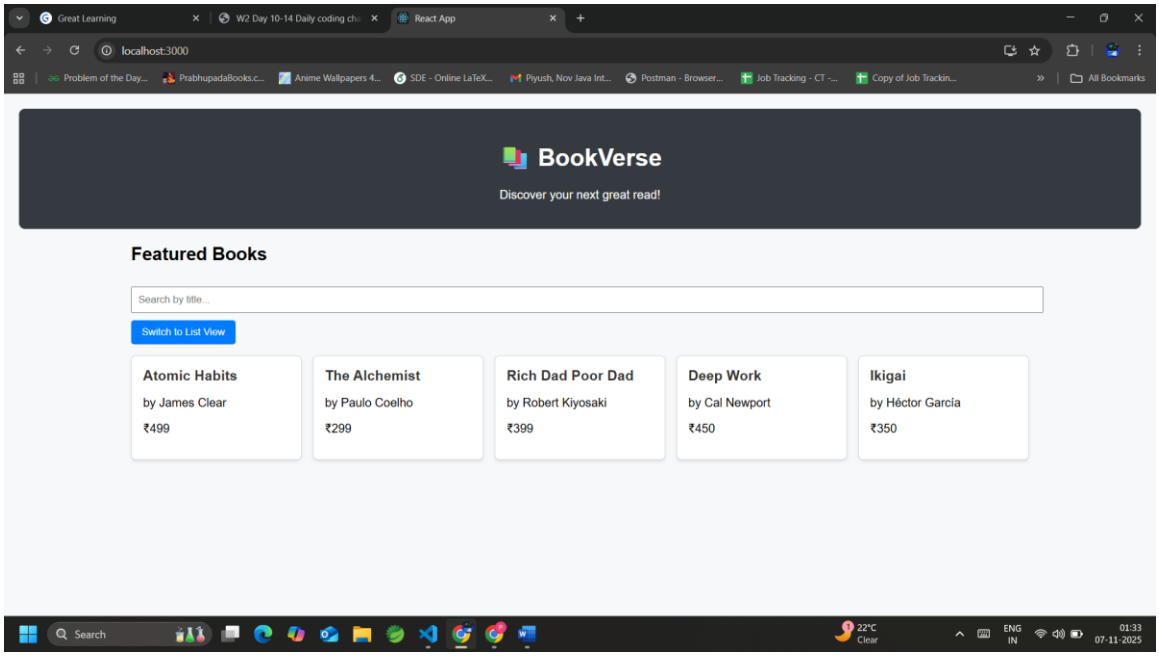
Title: BookVerse – Welcome Page and Featured Books Section

Objective: Display a list of featured books with two layout options (Grid/List) and a search filter.

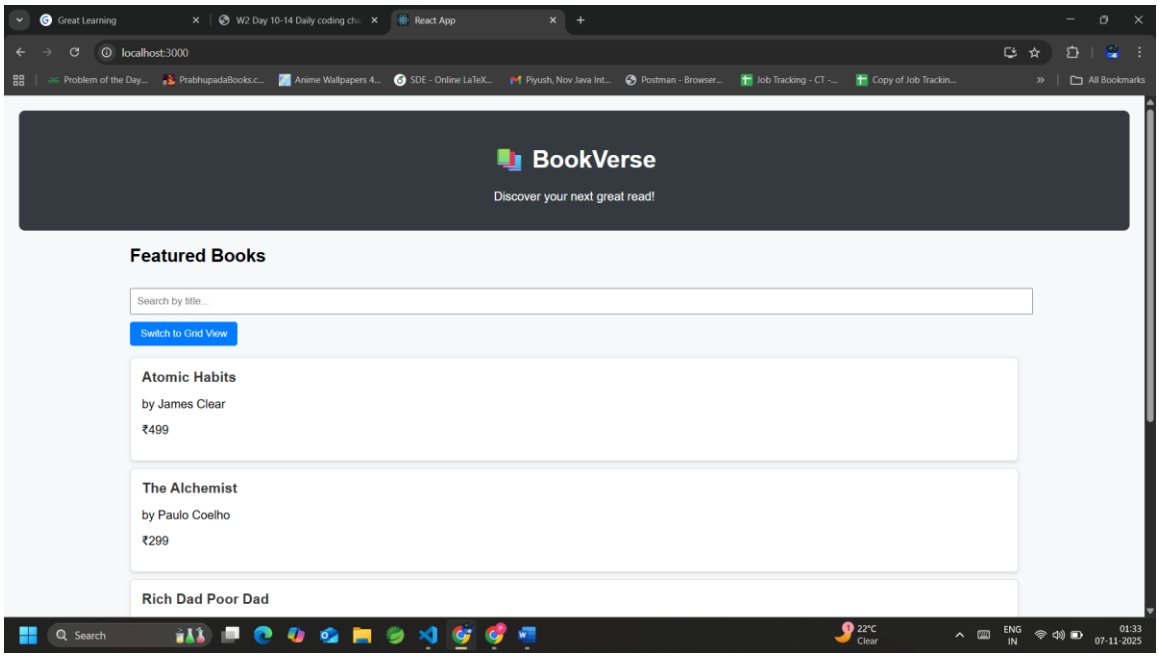
Key Features:
- Toggle view between Grid and List layouts.
- Controlled search input for filtering books.
- Reusable BookCard and SearchBox components.

2. Output Screenshots (Placeholders)

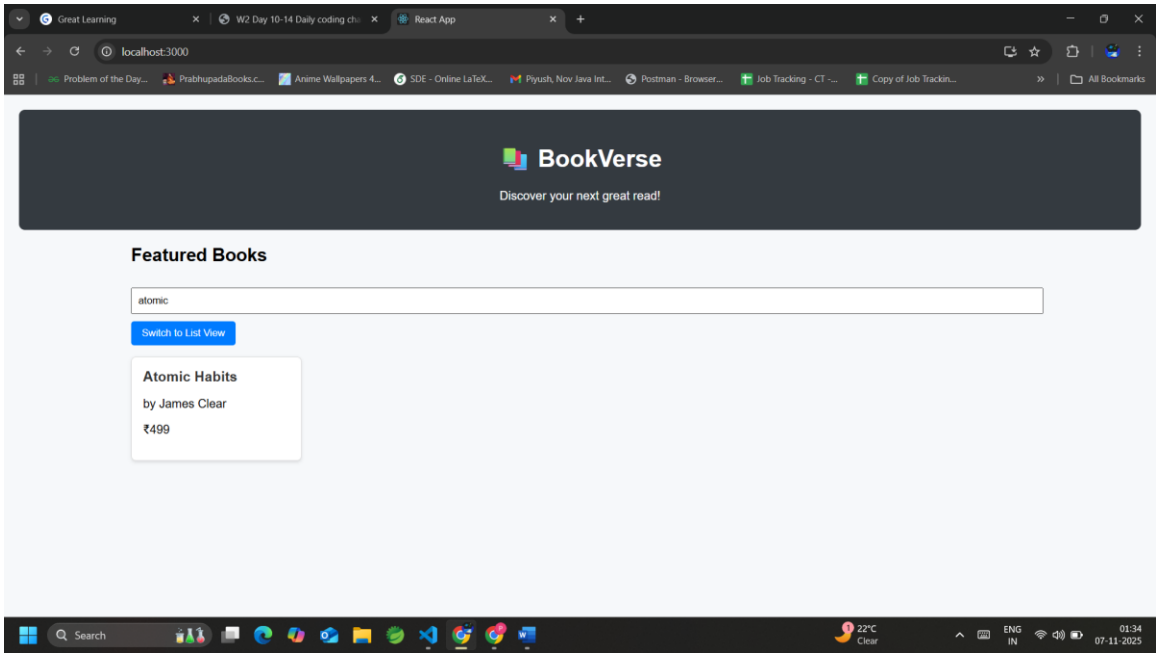
[Screenshot 1: Home Page with Grid View Placeholder]



[Screenshot 2: List View Placeholder]



[Screenshot 3: Search Filter Active Placeholder]



3. Components and Code Snippets

3.1 App.jsx

Parent component rendering the BookList and overall page header.

```
// Parent component to render the BookList and display page header

import React from "react";
import BookList from "./components/BookList";
import "./App.css";

const App = () => {
  return (
    <div className="App">
      <header>
        <h1>📚 BookVerse</h1>
        <p>Discover your next great read!</p>
      </header>

      {/* Render the main BookList section */}
      <BookList />
    </div>
  );
};

export default App;
```

Explanation: The App component serves as the main container that renders the BookList and provides the header section.

3.2 BookList.jsx

Parent component managing layout state, search filter, and rendering multiple BookCards.

```
// Parent component that renders multiple BookCard components

import React, { useState } from "react";
import BookCard from "./BookCard";
import SearchBox from "./SearchBox";
import booksData from "../data/booksData";
// import "./BookList.css";

const BookList = () => {
    // State for list of books and layout mode
    const [layout, setLayout] = useState("grid"); // grid or list
    const [searchTerm, setSearchTerm] = useState("");

    // Function to handle layout toggle
    const toggleLayout = () => {
        setLayout(layout === "grid" ? "list" : "grid");
```

```
    };

    // Filter books based on search input
    const filteredBooks = booksData.filter((book) =>
        book.title.toLowerCase().includes(searchTerm.toLowerCase())
    );

    return (
        <div className="book-list-container">
            <h2>Featured Books</h2>

            {/* Controlled input for search */}
            <SearchBox searchTerm={searchTerm}
setSearchTerm={setSearchTerm} />

            {/* Layout toggle button */}
            <button onClick={toggleLayout} className="toggle-btn">
                Switch to {layout === "grid" ? "List" : "Grid"} View
            </button>

            {/* Render filtered books */}
            <div className={`book-list ${layout}`}>
                {filteredBooks.length > 0 ? (
                    filteredBooks.map((book) => (
                        <BookCard
                            key={book.id}
                            title={book.title}
                            author={book.author}
                            price={book.price}
                            layout={layout}
                        />
                    ))
                ) : (
                    <p>No books found.</p>
                )}
            </div>
        </div>
    );
};

export default BookList;
```

Explanation: The BookList component manages state for layout and search functionality, dynamically rendering BookCard components.

3.3 BookCard.jsx

Functional component displaying book title, author, and price details.

```
// Functional component to display individual book details

import React from "react";
// import "./BookCard.css"; // optional styling file if you want to style
each card

const BookCard = ({ title, author, price, layout }) => {
    return (
        <div className={`book-card ${layout}`}>
            <h3 className="book-title">{title}</h3>
            <p className="book-author">by {author}</p>
            <p className="book-price">₹{price}</p>
        </div>
    );
};

export default BookCard;
```

Explanation: The BookCard component uses props to display book data and adapts its styling based on the selected layout.

3.4 SearchBox.jsx

Controlled component for handling user input to filter books by title.

```
// Controlled component for managing search input

import React from "react";

const SearchBox = ({ searchTerm, setSearchTerm }) => {
    // Handle input change event
    const handleChange = (e) => {
        setSearchTerm(e.target.value);
    };

    return (
        <div className="search-box">
            <input
                type="text"
```

```
                placeholder="Search by title..."
                value={searchTerm}
                onChange={handleChange}
            />
        </div>
    );
};

export default SearchBox;
```

Explanation: The SearchBox maintains controlled input by reflecting its value from state and updating via event handler.

3.5 booksData.js

Static data file storing featured books to be displayed on the homepage.

```
// Sample data for featured books used in BookList

const booksData = [
    { id: 1, title: "Atomic Habits", author: "James Clear", price: 499 },
    { id: 2, title: "The Alchemist", author: "Paulo Coelho", price: 299 },
    { id: 3, title: "Rich Dad Poor Dad", author: "Robert Kiyosaki", price:
399 },
    { id: 4, title: "Deep Work", author: "Cal Newport", price: 450 },
    { id: 5, title: "Ikigai", author: "Héctor García", price: 350 },
];

export default booksData;
```

Explanation: The data file provides sample content used by BookList to render featured books.

4. Summary

Day 1 implementation successfully demonstrates React fundamentals, including component composition, state management, event handling, and controlled components. The BookVerse app provides a user-friendly interface for browsing featured books with dynamic layout switching and live search filtering.