BIBLIO - BOOK APP

A SIMPLE BOOK INFO APP

INTRODUCTION

“The biblio book App “ is a simple yet elegant app , which gives you details about books , authors , short summary . It also allows the user to add books , along with its description and stock images . This app can be used commercially or as a personal app to make note of the books we’ve read before to create our own virtual library , it can also be recommended to someone .

PROBLEM STATEMENT

Create an angular SPA using the following feature

1. Components
2. Parent Child components
3. Structural and attribute directives
4. @Input
5. Basic Routing
6. Services
7. Observables
8. API Calls
9. Template and Reactive Forms
10. Pipes

1. COMPONENTS

- The application is built using reusable components:
    - BookListComponent – Displays all books.
    - BookDetailComponent – Shows detailed information about a selected book.
    - AddBookComponent – Provides a form to add new books.
    - EditBookComponent – Allows editing existing book details.

2. Parent-Child Components

- BookListComponent (Parent) passes selected book details to BookDetailComponent (Child) using @Input().

3. @Input Decorator

- Used in BookDetailComponent to receive book data from BookListComponent.

4. Structural and Attribute Directives

- *ngFor displays the list of books dynamically.
- *ngIf ensures that the Book Details Section is displayed only when a book is selected.
- [ngClass] highlights out-of-stock books differently.

5. Basic Routing

Configured routes to navigate between book-related pages.

6. Services

BookService centralizes book-related logic and API interactions.

7. Observables

Used in BookService to handle book data as an observable stream.

8. API calls

BookService fetches data from a backend API using HttpClientModule.

9. Template and Reactive Forms

- Template form for searching books.
- Reactive form for adding/editing books.

APPLICATION WORKFLOW

Step 1: Application Initialization

- The application starts with main.ts, bootstrapping the AppModule.
- The AppComponent serves as the root component, containing the navigation bar and routing outlet.

Step 2: Navigation and Routing

- Implemented Angular Router in app-routing.module.ts to enable navigation between:
  - Home Page – Displays an overview of the bookstore.
  - Book Details Page – Provides more information on a selected book.
  - Add/Edit Book Page – Allows users to add new books or update existing ones.

Step 3: Data Flow and Communication

- The Book List Component (parent) passes selected book data to the Book Details Component (child) using @Input().
- Services facilitate data sharing between unrelated components, such as book selection and updates.

Step 4: Fetching and Displaying Data

- The Book Service handles API calls to fetch books from a mock backend.
- The book data is displayed using *ngFor within the Book List Component.

Step 5: User Interaction and Forms

- Template-driven form is used for the search bar, allowing users to filter books by title or author.
- Reactive form is used in the Add/Edit Book Page, enabling real-time validation before submission.

Step 6: Dynamic Content and UI Styling

- Structural directives (*ngIf, *ngFor) manage book listings dynamically.
- Attribute directives ([ngClass], [ngStyle]) highlight books based on availability.

Step 7: Handling Asynchronous Data

- The Book Service returns book data as an observable.
- Components subscribe to this observable to update the UI when new books are added or modified.

CONCLUSION

This Angular SPA Book App successfully integrates key Angular features to create an interactive and functional bookstore. With structured components, efficient data handling, and smooth navigation