High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **BookFinder - Library Search Interface**. The purpose of this project is to provide users with a responsive web interface to search, filter, and view book details in a library catalog.

---

1. System Architecture Overview

Architecture Description:
BookFinder is a single-page application (SPA) built with React and TypeScript. The frontend interacts with a backend API (not in scope) to fetch and display book data. The system is modular, with clear separation of concerns between UI, state management, and data access.

Module/Component	Role/Responsibility
Search Bar	Accepts user queries and triggers search actions
Filter Panel	Allows users to filter search results (e.g., genre)
Book List	Displays search/filter results in a responsive grid
Book Detail Modal	Shows detailed info for a selected book
State Management	Manages UI and data state (search, filters, results)
API Service Layer	Handles communication with backend API
UI Layout/Styling	Ensures responsive, accessible design (Tailwind)

---

2. Component Interactions

Source Component	Target Component	Interaction Description
Search Bar	API Service Layer	Sends search queries
Filter Panel	API Service Layer	Sends filter parameters
API Service Layer	State Management	Updates state with fetched book data
State Management	Book List	Provides data for rendering
Book List	Book Detail Modal	Triggers modal with selected book details
UI Layout/Styling	All UI Components	Applies responsive and consistent styling

Sequence Flow:
User enters a query/filter → Search/Filter triggers API call → API returns data → State updates → Book List renders results → User selects a book → Book Detail Modal displays info.

---

3. Data Flow Overview

Data Source	Data Destination	Data Description
User Input	State Management	Search terms, filter selections
State Management	API Service Layer	Search/filter parameters for API requests
API Service Layer	State Management	Book data from backend API
State Management	UI Components	Book list, selected book details

---

4. Technology Stack

Layer/Area	Technology/Frameworks
UI Framework	React (with TypeScript)
Styling	Tailwind CSS, HTML, CSS
State Management	React Context/State
API Communication	Fetch API or Axios
Tooling	Vite or Create React App, ESLint

---

5. Scalability & Reliability

- **Scalability:**
  The frontend is modular and stateless, enabling easy scaling via static hosting/CDN. Component-based design supports future feature expansion.
- **Reliability:**
  Error boundaries and input validation ensure robust user experience. API errors are gracefully handled and communicated to users.
- **Security:**
  No sensitive data is handled on the frontend. Standard best practices (e.g., input sanitization) are followed.

End of Document