# Book Finder – Project Notes

## Overview

Book Finder is a modern and responsive book search web application built using React, Vite, and Tailwind CSS. It allows users to explore the Open Library database, search for books by title or author, and refine results with advanced filters such as first publish year, language, and full-text availability. The app focuses on delivering a clean, user-friendly experience and works seamlessly on both desktop and mobile devices.

## Plan & Structure

The development of Book Finder followed a structured plan:  
1. **Setup & Initialization**: Create a React + Vite project and integrate Tailwind CSS for styling.  
2. **Core Layout Design**: Build the main page with a header, search bar, book list display area, and footer.  
3. **State Management**: Use React Context API to manage global state, including search results, loading, and error handling.  
4. **API Integration**: Connect with the Open Library Search API to fetch and display books based on user queries and filters.  
5. **Feature Enhancement**: Add advanced filters (language, first publish year, full-text availability), responsive UI components, and toast notifications for user feedback.  
6. **Testing & Deployment**: Test the app in production mode and deploy using platforms like CodeSandbox for public access.

## Objectives

- Build a fast and responsive web application for searching books.  
- Provide a smooth and intuitive user experience with real-time results.  
- Allow users to filter and explore book data from a large public database.  
- Ensure the application is easily deployable and requires no API keys for configuration.

## Features Breakdown & Explanation

1. **Book Search**:  
 - Users can search for books by title or author.  
 - Results are fetched in real-time from the Open Library API.  
  
2. **Advanced Filters**:  
 - Filter by First Publish Year: Helps narrow down results to a specific time period.  
 - Language Filter: Users can select languages (e.g., English, French, German) using a convenient dropdown.  
 - Full-Text Availability: Option to show only books with complete readable text.  
  
3. **Book List Display**:  
 - Books are displayed in a responsive grid with cover image, title, author(s), language, and publication year.  
 - Graceful fallbacks handle missing cover images.  
  
4. **Responsive UI**:  
 - Layout and design adapt seamlessly to mobile and desktop screens.  
  
5. **Error Handling**:  
 - React Toastify is used to show toast notifications for errors, empty queries, or network issues.  
  
6. **State Management**:  
 - React Context API provides a global state for search results, loading status, and errors.  
 - Simplifies data flow across components.

## Tech Stack

- **Frontend**: React + Vite for fast development and optimized builds.  
- **Styling**:\*Tailwind CSS for responsive and modern UI design.  
- **State Management**: React Context API for centralized state handling.  
- **Notifications**: React Toastify for elegant toast messages.  
- **Icons**: Lucide React for clean and scalable icons.  
- **API**: Open Library Search API for fetching book data without authentication.

## Deployment

The app can be run locally using the development server (`npm run dev`) and tested in production mode using the build and preview commands (`npm run build` and `npm run preview`). A live demo is deployed on CodeSandbox for easy public access.