

# BOOK FINDER APPLICATION

~By Aman Kumar Sinha

## Document Version Control

Date Issued	Version	Description	Author
11-07-2024	1.0	Initial HLD	Aman Kumar Sinha
18-07-2024	1.5	Final Draft	Aman Kumar Sinha
08-08-2024	2.0	Final Version	Aman Kumar Sinha

## Contents

Document Version Control .....	2
Abstract .....	4
1. Introduction .....	5
1.1 Why this High-Level Design Document? .....	5
1.2 Scope .....	5
1.3 Definitions .....	5
2. General Description .....	6
2.1 Product Perspective .....	6
2.2 Problem statement .....	6
2.3 Proposed Solution .....	6
2.4 Further Improvements .....	6
2.5 Project requirements .....	6
2.6 Tools Used .....	6
2.7 Constraints .....	7
2.8 Assumptions .....	7
3. Design details .....	8
3.1 Process Flow .....	8
3.2 Development Process .....	8
3.3 Error handling .....	9
4. Performance and Accessibility .....	10
4.1 Reusability .....	10
4.2 Application compatibility .....	10
4.3 Resource utilization .....	10
4.4 Deployment .....	10
5. Conclusion .....	11

## Abstract

The Book Finder application is a user-friendly website designed to help readers discover trending books and explore a vast collection of literary works. By leveraging the Google Books API, the platform provides detailed metadata, allowing users to search for books by title or its author. Whether you're looking for the latest bestsellers or hidden gems, this application offers an intuitive way to find and learn about books that match your interests. Users can easily browse trending titles, read in-depth details, and make informed decisions on what to read next.

## 1. Introduction

### 1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include the design features and the architecture of the project
- List and describe the non-functional attributes like:
  - Security
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

### 1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

### 1.3 Definitions

Term	Description
API	Application Programming Interface
CRA	Create React App
HTML	Hyper Text Markup Language
CSS	Cascading Stylesheet
JS	JavaScript
VCS	Version Control System

## 2. General Description

### 2.1 Product Perspective

The Book Finder Application assists users in discovering their next read by showcasing trending books and providing detailed information on any book they're interested in exploring.

### 2.2 Problem statement

The goal is to create a book meta-search site that aggregates books in one place. When a user searches for a book, the application will look it up and, if found, return the results.

- Designing a page where users can search for books.
- Initiating a search query, and if the book is found, displaying the results on the front end.

### 2.3 Proposed Solution

The proposed solution is a web application that utilizes the Google Books API to retrieve book metadata. It also features an explore section showcasing a curated list of books from the New York Times Best Sellers.

### 2.4 Further Improvements

This project can be improved by adding New York Times API to fetch trending books and automate the curation of the trending books to show for users.

### 2.5 Project requirements

In this project we need Meta data of books and trending books data can be curated from the New York Times bestselling books data.

- Meta data can be fetched using Google Books API
- Trending books list is obtained from New York Times Best Selling books data

### 2.6 Tools Used

In this project we are using ReactJS Library to build this application because of its ease of use and rapid development environment it provides.

- VS Code as a Code Editor
- Google Chrome as a web browser
- ReactJS to build Frontend of the application
- React-Router-DOM for client-side routing
- React-Icons for social media icons.
- React-Type-Animation for Text Animation
- Tailwind CSS to handle the styling of the elements on our website
- Google Books API to fetch metadata of the books
- Draftbit API for book recommendation.
- GitHub as Version Control System

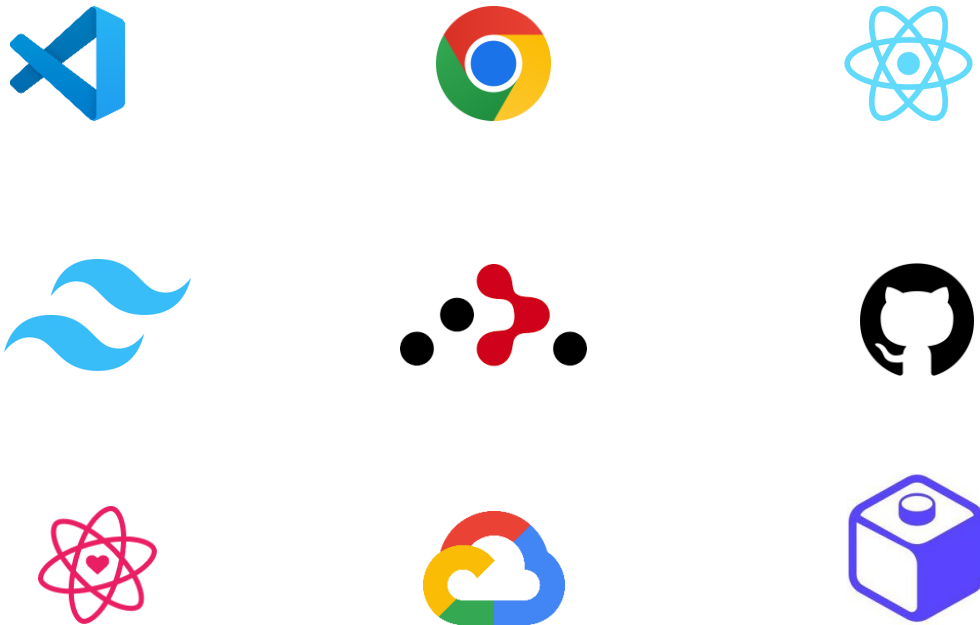


Figure 1. Technologies Used

## 2.7 Constraints

The book finder application is a user-friendly web application which helps user to find metadata data of the books they wanted to search without any distractions.

## 2.8 Assumptions

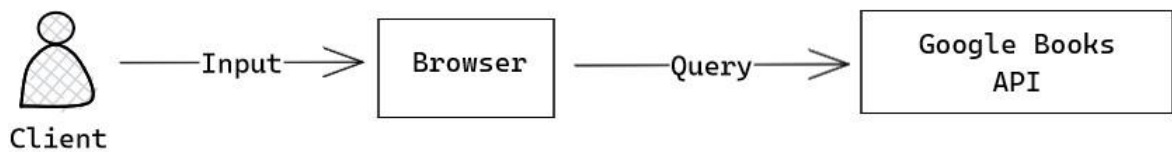
The primary objective of this project is to offer users a distraction-free environment for searching books. The search query, provided by the user, assumes that they have some knowledge of book titles or authors and seek additional information or metadata about the books they are interested in reading. By focusing on a streamlined search experience, the application allows users to efficiently find detailed information about specific books, helping them make informed decisions without unnecessary distractions. This approach enhances the user experience by providing relevant book details quickly and effectively.

### 3. Design details

#### 3.1 Process Flow

To retrieve a book's metadata, we take input from the user and send a request to the Google Books API. The response is then presented to the user in an intuitive and user-friendly manner. The process flow diagram below illustrates this workflow.

##### 1. Client requesting server



##### 2. Server sending response

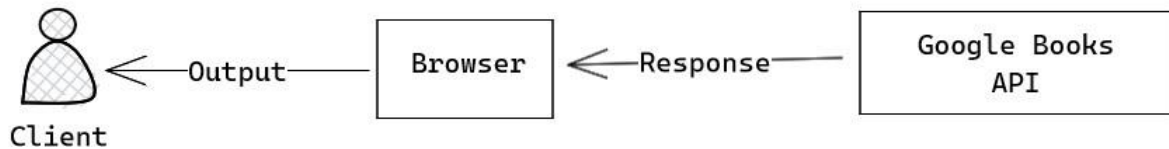


Figure 2. Process flow

#### 3.2 Development Process

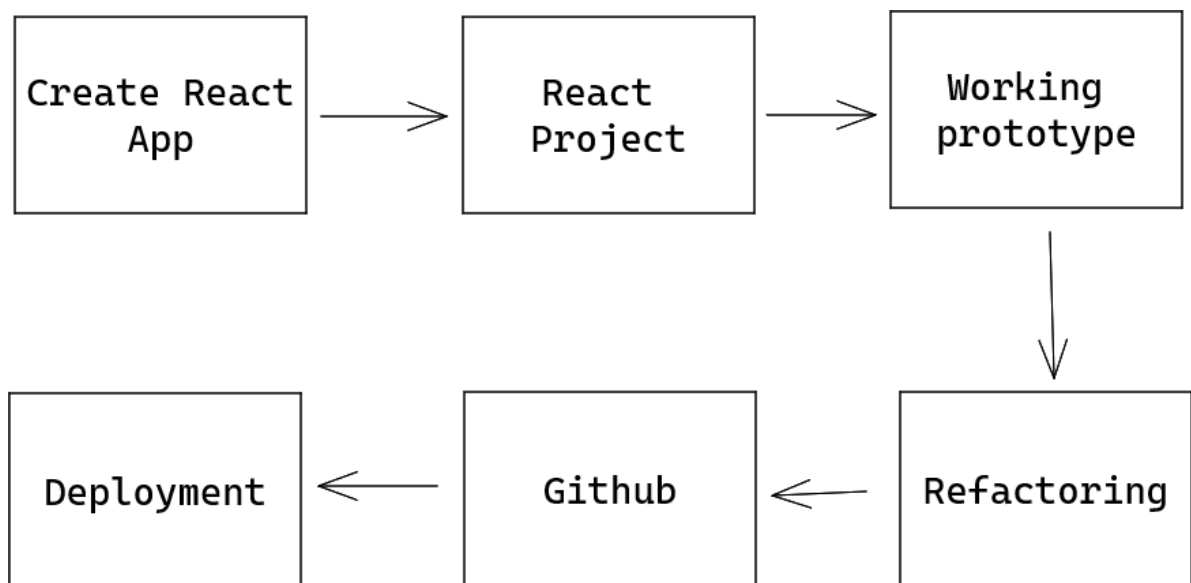


Figure 3. Development flow



### 3.3 Error handling

If any errors occur, an explanation will be provided to clarify what went wrong. An error is defined as any event that deviates from the normal and intended usage of the application.

## 4. Performance and Accessibility

The Book Finder Application should deliver high performance, with fast page loads and minimal payload in API calls. It must also adhere to best practices for accessibility, ensuring an efficient and inclusive user experience.

### 4.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

### 4.2 Application compatibility

This project should be cross platform supporting i.e., we should support mobile and desktops also. To resolve this, we are adapting mobile first development while building our application.

### 4.3 Resource utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

### 4.4 Deployment

We chose Vercel to host this application due to its ease of use and cost-effectiveness compared to other hosting alternatives.



*Figure 4. Vercel Hosting provider*

## 5. Conclusion

The Book Finder Application offers an enhanced way for users to discover and access information about the books they want to read, all within a distraction-free environment. By delivering a clean and intuitive user interface, the application ensures that users can focus on finding the details they need without unnecessary interruptions. The platform is designed to streamline the search process, making it easier for users to explore book metadata and make informed reading choices. This thoughtful approach to design and functionality creates a more enjoyable and efficient experience for book lovers.