BOOK FINDER APPLICATION

By Bobby Sadhwani

Document Version Control

Date Issued	Version	Description	Author
21-01-2023	1.0	Initial HLD	Bobby Sadhwani
13-01-2023	1.5	Final Draft	Bobby Sadhwani
13-01-2023	2.0	Final Version	Bobby Sadhwani

Contents

Abstract

The book finder application is a website where users can find any eBook and they can view detailed information about the books.
This application uses the Google Books API to fetch the meta data of the books.

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 utilisation
    - 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 will help the users to find the next book they want to read by showing them the trending books in the space and also helps them to find details of books they want to read

2.2 Problem statement

To create a book meta-search site. The books are aggregated in one place and when user searches for a book then the application will search for the book and if found then return it

- To design a page where users can search for the book
- After the search is initiated check the book and if found display the books on the front end

2.3 Proposed Solution

The solution proposed here is a web application which uses Google Books API to fetch the meta data of the books. And an explore section where we show curated list of books from 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
- Tending 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
- Tailwind CSS to handle the styling of the elements on our website
- Google Books API to fetch metadata of the books
- Github as Version Control System

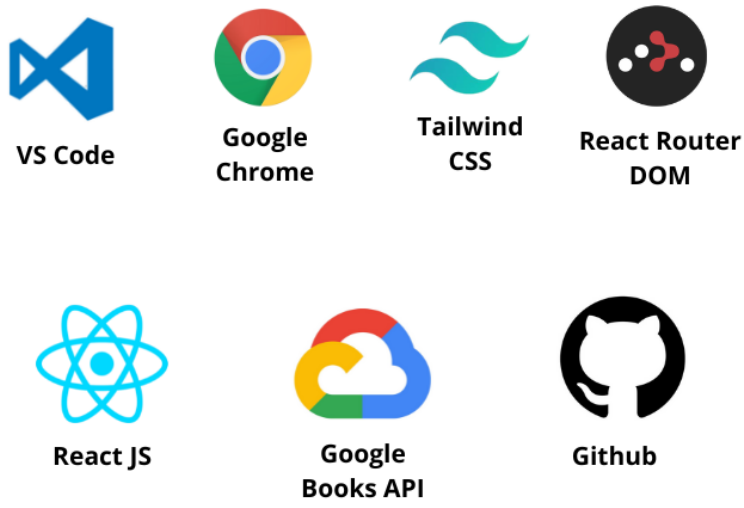

Figure 1. Technologies Used

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 main of objective of the project is to provide users a distraction free environment to search books (2.2 Problem Statement). The input search query comes from the user, we assume that the users are aware of the book titles or authors and wants to find the meta data i.e., more information about the book they wanted to read.

3. Design details

3.1 Process Flow

To fetch meta data of a book we take input from the client and request Google Books API then we show that response to client in the intuitive way. Process flow diagram is shown below.

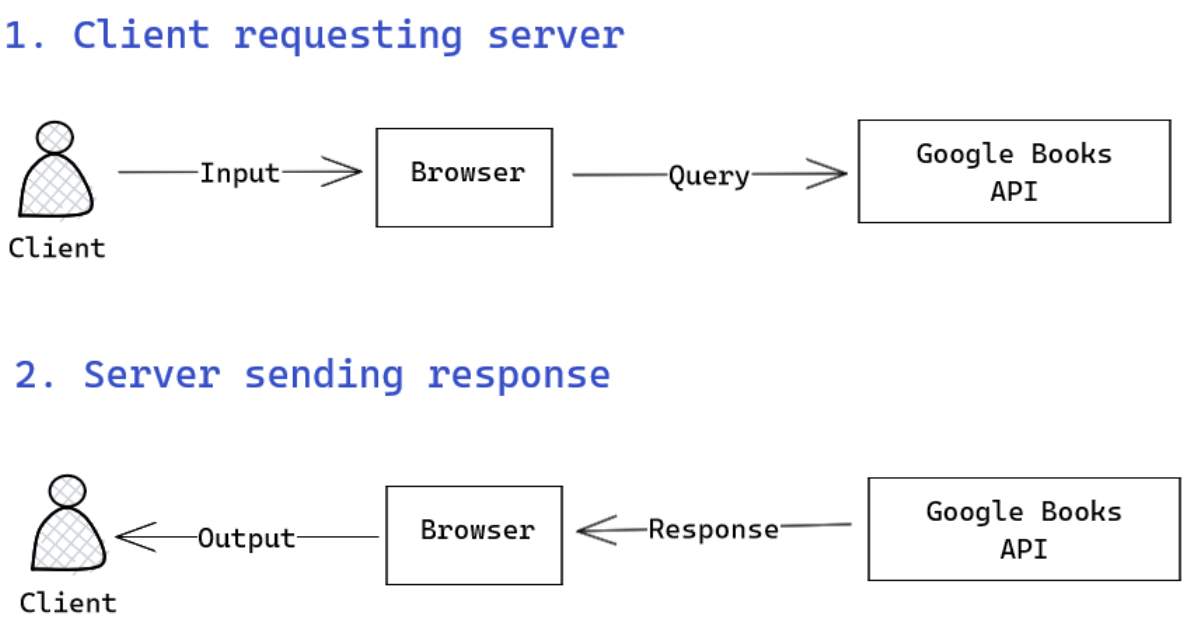

Figure 2. Process flow

3.2 Development Process

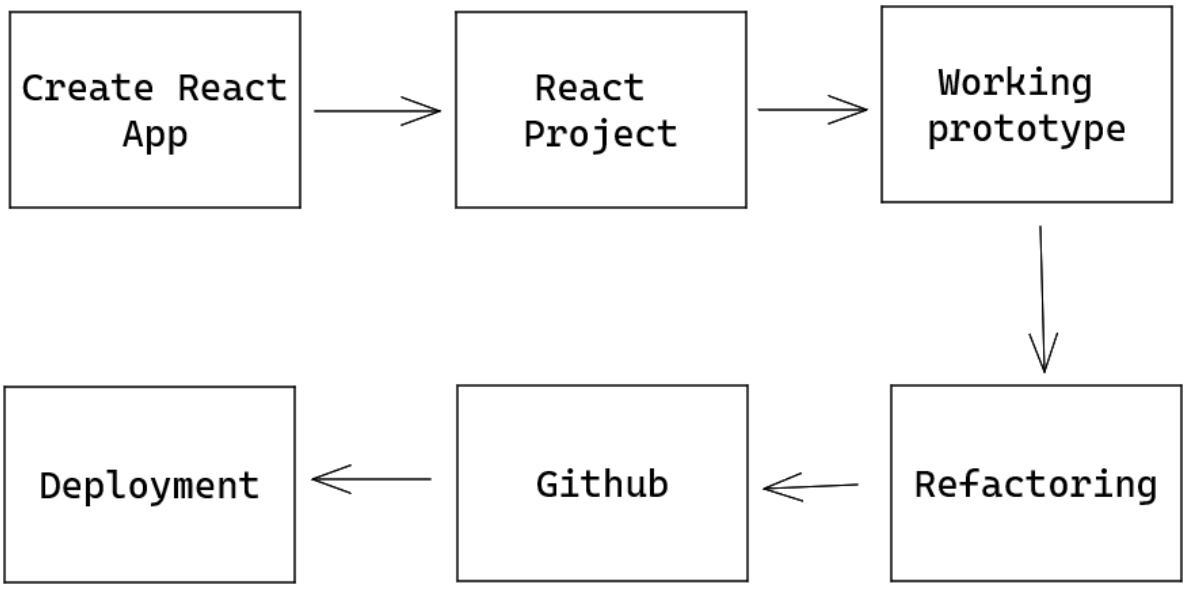

Figure 3. Development flow

3.3 Error handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

4. Performance and Accessibility

The Book Finder Application should be having high performance i.e., faster page loads and less payload in API calls. And it should be following good practices for accessibility.

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 utilisation

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

4.4 Deployment

For this application we choose Netlify for hosting our application because of its ease of use and its comparatively cheaper when compared to its alternatives

Figure 4. Netlify Hosting provider

5. Conclusion

The Book Finder Application will provide a better way to find the information of books they want to read in a distraction free environment providing a nice user interface.