Architecture Document

BOOK FINDER APPLICATION

By Bobby Sadhwani

Document Version Control

Date Issued	Version	Description	Author
07-01-2023	1.0	Initial HLD	Bobby Sadhwani
08-01-2023	1.5	Final Draft	Bobby Sadhwani
08-01-2023	2.0	Final Version	Bobby Sadhwani

Contents

Abstract

The book finder application us a website where users can find any eBook and they can search for the books and view the 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 Architecture Document?

The purpose of this Architecture Document is to provide a high-level overview of the technological architecture we are using in this Book Finder Application.

1.2 Scope

The Architecture documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The Architecture Document 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

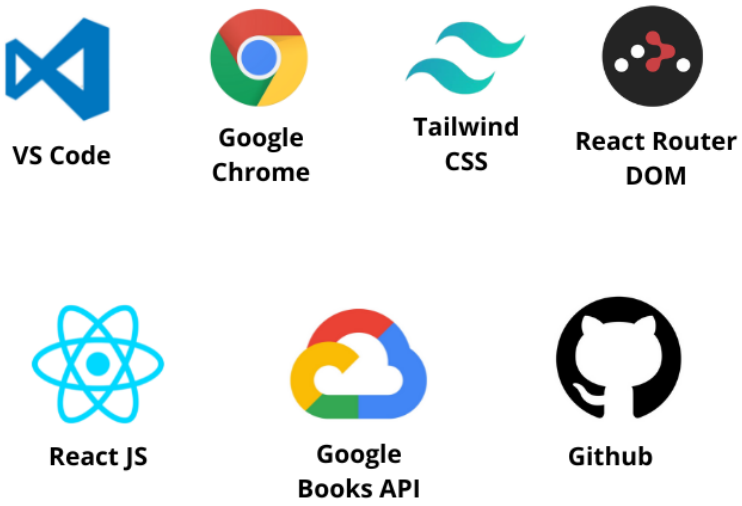

Figure 1. Technologies Used

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 Frontend Architecture

Frontend is the main part of our application which is primarily consumed by the clients to view the meta data of books here we made a thoughtful choice by choosing the below technologies.

- React JS
- Tailwind CSS
- React Router DOM
- Google Books API

3.3 Source Control Architecture

Source control is the key to develop a scalable and efficient project. We choose the Github as a service because it is most popular and free solution as well.

3.4 Deployment Architecture

To take our application into production we need a service which is fast and reliable and has a good uptime server. We found that Netlify offers this with nice user interface so we went with this option.

4. 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.