**Full Stack Development with MERN**

**Frontend Development Report**

| Date | 12.07.24 |
| --- | --- |
| Team ID | SWTID1719925692 |
| Project Name | Bookstore |
| Maximum Marks |  |

**Project Title**: Book Corner

**Date**: 12-07-2024

**Prepared by**:

1.Shaneel Reddy

2 . Vinuthna

3 . Mahitha

4 . Allwin

**Objective**

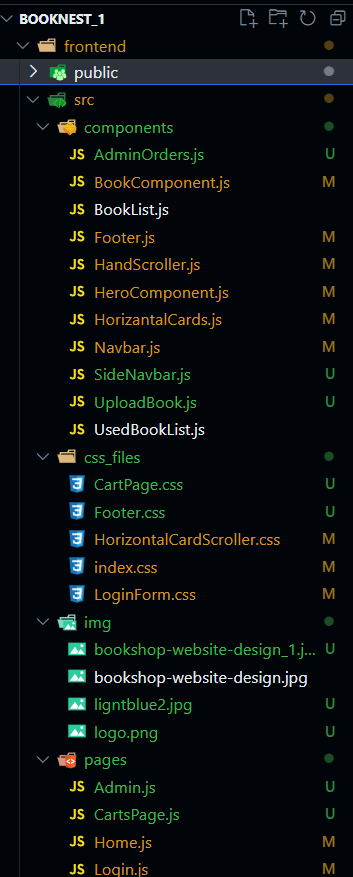
The objective of this frontend component is to facilitate users in listing their books for sale on the BookSell Store. Additionally, it promotes a membership offer that allows users to sell unlimited books per month, enhancing user engagement and sales activity on the platform.

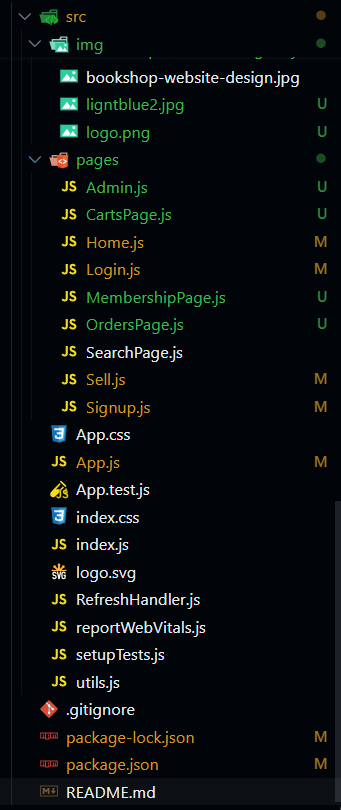
The user can also add to cart their favourite books and place a order

**Technologies Used**

* **Frontend Framework:** React.js
* **UI Framework/Libraries:** Bootstrap
* **API Libraries:** Axios , Google Books Api

**Project Structure**





**Key Components**

1. **Frontend Framework:** React.js
2. **UI Framework/Libraries:** Bootstrap
3. **API Libraries:** Axios, Google Books Api

* **App.js:** Responsible for routing and main application layout.
* **/components: Contains reusable UI components used across the application.**
  + Navbar.js - code for the navbar which is called in required pages .
  + Footer.js - contains the function for footer called when needed.
  + SideNavbar.js -function called in the admin page to display the sidenav
  + AdminOrders.js -is used to fetched the details of the order details.
  + BookComponent.js - fetches the books from google Api.
  + BookList.js - fetches the bestseller book details .
  + Footer.js - for the footer of the website.
  + HandScroller.js - for displaying best seller books in cards.
  + HeroComponent.js - for the main banner thing in the home page .
  + HorizantalCards.js - for displaying the used book info in the cards.
  + Navbar.js - for the navbar component in the website .
  + SideNavbar.js - for vertical navbar in admin page .
  + UploadBook.js - upload functionality for the admin panel.
  + UsedBookList.js - used to fetch the used books details .
* **/pages: Includes different pages for the web app.**
  + HomePage.js - includes all the components from /components that should be displayed in the home page in order.
  + Signup.js - code for a form to be displayed with name , email and password so that user can signup
  + Admin.js-the page for admin panel
  + CartsPage.js-the page where user can add items to cart and place orders
  + Login.js-the user can give mail ,password and can login into the website.
  + MembershipPage.js-this page contains the details about the membership plans for selling books.
  + OrdersPage.js - this will display all the orders placed by the customer.
  + SearchPage.js - when the user searches the books it will direct them to this page with all the matched book details.
  + Sell.js-in this page the user can sell their books .

**Routing**

Routing is managed using React Router. Key routes include:

* **/login :**  Page for a user to login into the book website
* **/signup :** user should signup and then login into the website
* **/home:** Landing page of the application displaying books for sale.
* **/sell:** Page for users to list their books for sale.
* **/membership:** Page describing the membership offer.
* **/admin:** Admin dashboard for managing the store.
* **/order:** Page for users to view their orders.
* **/cart:** Page for managing the shopping cart.
* **/admin:** Page for managing the books for the admin

**Integration with Backend**

The frontend communicates with the backend APIs hosted on “https://bookcorner-jx21.onrender.com”. Key endpoints include:

* **GET /orders:** Retrieves order data for display on the orders page.
* **GET /getbooks:** Retrieves book information to display in the home page .
* **POST /postbooks:** for adding new .
* **POST /add :** For adding the books into the cart **.**
* **GET /get:** Retrieves the cart details of that users
* **GET /gbooks:** Retrieves the google book api’s book details
* **POST /addorder:** for uploading books into the orders page .
* **GET /getorder:**Retrieves book information to dispaly in the order page .
* **GET /usedbooks :** Retrieves the used book details
* **POST /submit:**for uploading the used books into the database
* **POST /login:** for uploading user information into the database .
* **POST /signup:** for uploading user information into the database
* **GET /users :** Retrieves the user information to display it in the admin panel

**User Interface (UI) Design**

* Implemented using Bootstrap and react to create a consistent and visually appealing interface.

**Third-Party Integrations (If any)**

* GoogleBooksapi is used to fetch the book data