YSD_B03_MERN - ReactJS Mini Project

For the ReactJS mini project you are required to integrate your previously built backend system in the Mid-Term exam, with the Frontend. The title would be the same as the previous one, which is **Online Book Store Management System**. It will be fully based on the **ReactJS** technologies and all the API data should be **visually represented** with React. Please note that all necessary actions and events must be informed to the user through toaster messages and/or notifications.

In the system, there will be two types of users, as mentioned below, who can access it. Page access management will be based on react-router-dom authentication and authorization.

The **user** will be able to buy and checkout books using a cart based system. Additionally, the prices of the books will also vary based on the discounts that are provided **(Optional)**.

The other type of user in the system will be an **administrator** who can add and remove books from the system, and update their data accordingly. They will also be able to manipulate specific user data.

Technologies Implementation:

1. ReactJS(Vite)
2. SCSS
3. React-router-dom
4. React-hook-form
5. Custom Hooks
6. Redux **or** Context-API

Types of user
1. User
2. Admin

Functionalities:

Common Functionality	1. Log in to the system
	2. View all the books
	3. View all the books (with filter, sort, pagination and search)

	4. Give feedback to specific books.
User	1. Sign up for system
	2. View books in cart
	3. View transactions
	4. Add books to cart individually
	5. Remove books from cart individually
	6. Add rating for books
	7. Remove rating for books
	8. Add reviews for books
	9. Edit reviews for books
	10. Remove reviews for books
	11. Avail discounted price for books if discount time is valid **(Optional)**
	12. Add transaction by checking out using cart
	13. Book stock must be reduced after a successful checkout
	14. Every user will have a balance, that will be deducted from once a transaction (checkout) successfully takes place
	15. Balance can be updated by user when its too low
	16. Prevent transaction from taking place if user does not have sufficient balance
Admin	**1. Cart, transaction is not required for admin**
	2. View transactions of all users
	3. View all user data
	4. Edit selected existing user data
	5. Delete user

	6. Edit book data individually
	7. Add new books
	8. Delete existing books
	9. Add discounts for books, which will be available for certain products only for a limited amount of time **(Optional)**
	10. Update discounts for books **(Optional)**

Deliverables:

1. Design Document(Project GUI) with the snapshot of pages (in **doc/pdf** file)
2. Project implementation using ReactJS a. **Project Output** should be shown in **Browser** (Chrome preferred). b. Source Code(GitHub)
Start Date: 11-10-2023 Submission Date: 13-10-2023(Till **8am**) Demonstration Date: 13-10-2023

Evaluation criteria:

Requirement understanding - 10
Expected Output - 10
Code Quality & Component Reusability -10
GUI Design - 5
Demonstration & Live Coding Test - 15