**Scope**:

Create a book store application that provides ability to manage (Get, Create, Update, Delete) Books, Publishers. The bookstore provides users to find the books from catalog, and add the books from catalogue to shopping cart and checkout the shopping cart, place order on the backend application.

**Backend**:

Use Json-server to simulate the RESTFul back end server with easy steps. https://github.com/typicode/json-server

Json server to provides 3 sets of APIs

1. Books API
   1. GET /books (get all books)
   2. GET /books/123232 (get one book by id)
   3. PUT /books/123232 with body (update existing book)
   4. POST /books with body (create a new book)
2. Orders API
   1. GET /orders (get all orders )
   2. GET /orders/232 (get one order by id)
   3. PUT / orders /232 with body (update existing order)
   4. POST / orders with body (create a new order)

Duration: 2 weeks (10 working days)

Milestone 1 scope: 3 days

1. Create React App with TypeScript, add static pages with components like App, Application Header, Footer components, Home page, About page and Contact pages for Book Store.
2. Include React Router, add the navigation for React Application, that links to all the static pages

Milestone 2 scope: 2 days (Work Sat, Sun, if required)

1. Include Redux for the application, implement BookList, BookEdit, BookDetail components, implement the reducers, containers for those Book listing, editing, details pages.
2. Include routing for Books modules
3. Pull the data from REST API, complete the listing of books, Editing book details, create a new book

Milestone 3 scope: 3 days

1. Implement shopping cart. 20% discount applied for all Technical Books, 50% discount for all School Books, 10% discount for all other books
2. Add Limits per category
3. Add books to shopping cart, apply discount based on book category, calculate the grand total
4. Implement Checkout page, post the order to backend to orders API

Milestone 4: 2 days

1. Bug Fixes, Performance considerations
2. Order Listing
3. Auth : Optional. Implement AuthReducer, simulate redux login/logout