Assignment # 5 (Weightage in final grade ➔ 10%)
Product Ordering Dashboard

1. Overview

Goal of this assignment is to extend the previous Product Ordering System; which can be used to place orders based on the items available. It should be developed using Complete structure of MERN.

It is expected that proper transaction management will be in place when reading/writing to the database. Details regarding transactions can be found in the lecture notes of Week #10 & 11 on meskanas. Code samples are also available in the Sample code repository (https://github.com/iqmacewan/CodeSamples).

2. Objectives

- ✓ All features specified in Assignment # 4 should be functional and available
- ✓ User should be able to Sort products and Orders
- ✓ User should be able to Sort based on any column
- ✓ Filtering options for Category and Price range in dashboard
- ✓ Ability to cancel orders where the delivery date is more than 5 days away
- ✓ Persist UI state using **Redux** and Local Storage

3. Requirements

Backend (Express & MongoDB)

Modify the existing API's and add the new required.

Products API

1. **GET /products?sort={column}&order={asc/desc}**
   - Sort proudcts by name price or stock

2. **GET /products?category={category}&price_gte={min}&price_lte={max}**
   - Filter Products based on category and price range

Orders API

2. **PUT /orders/cancel/:id**
   - Cancel an order only if the delivery date is more than 5 days from today.

- If cancelled, restore the product's stock.

Frontend **(React UI)**

❖ **Home (Dashboard)**

This dashboard page should display a list of potential products to be purchased. There should be at least 15 Products and few of them should have zero stock. Those items which have zero stock; user cannot place order against and order button should be disabled for such products.

Dashboard Page (Product List)

Product Name	Category	Price	Stock	Action
Laptop	Electronics	$1000	10	Order
Phone	Electronics	$500	5	Order
Chair	Furniture	$150	20	Order

- Click on column headers to toggle ascending/descending
- Filtering section above the table
  - Category Drop Down
  - Price range inputs: Min Price, Max Price
  - Filter should be applied as soon as the value changes in category drop down or price range

❖ **Orders**

Place Orders Page

Order ID	Product	Quantity	Delivery Date	Status	Action
1001	Laptop	2	2025-04-01	Confirmed	Cancel
1002	Phone	1	2025-03-30	Confirmed	Cancel
1003	Chair	3	2025-04-12	Confirmed	Cancel

- Sorting will be applied when user clicks on column headers
- Cancel button will be visible only if delivery date is more than 5 days away from today.
- Status column updates to "Canceled" after cancellation

- Stock value should be restored in the products cancellation after cancelling an order
- Product Details: Name, Price, Available Stock
- Quantity Selector: (Dropdown/Input field)
- Delivery Date Selector: (Date Picker)
- Email Address Selector: (Input field)
- Confirm Order Button

Once the order is confirmed then the user is redirected to the **Orders Page** and the stock in the **Products collection** should be updated.

Orders Page (Orders List)

This page will show all the order users have placed. Please note that Email should be displayed in the table below as well.

Order ID	Product	Quantity	Delivery Date	Status	Action
1001	Laptop	2	2025-04-01	Confirmed	Cancel ❌
1002	Phone	1	2025-03-30	Confirmed	Cancel ❌
1003	Chair	3	2025-04-12	Confirmed	Cancel ❌

Users should be able to navigate back to the dashboard page

4. Submission Guidelines

- Share the link of your repository in the submission notes.

5. Evaluation Criteria

Your assignment will be evaluated based on:

- **Correctness:** Do your functions produce the expected outputs for various test cases and edge cases?
- **Code Quality:** Is your code organized, readable, and well-documented