React Assignment – Ecommerce Website (Extended Version)

Objective

Enhance the existing Ecommerce website by integrating advanced features such as cart management, order tracking, and state management using Redux. The goal is to simulate a complete ecommerce flow — from browsing products to placing and tracking orders.

Existing Functionality (From Previous Assignment)

The current application includes:

- **Landing Page:** Displays products category-wise with search and filter options.
- **Product Details Page:** Shows product details using dynamic routing.
- **Header:** Contains App Logo, Cart, Notifications, and Profile icons.
- **API Source:** Fake Store API

New Features & Functionalities

1. Add to Cart Functionality

- Enable users to add products to their cart from product cards or product details pages.
- Update cart item count dynamically in the header.
- Persist cart data using Redux state (bonus: optional localStorage persistence).

2. Cart Screen

- Display all added items with:
  - Product name, image, and price.
  - Quantity controls (+/-).
  - Individual item total and grand total.
- Include:
  - **Promo Code Section:** Apply discounts (use a mock promo for testing).
  - **Checkout Button:** Navigates to checkout flow.
  - **Address Form:** Capture delivery details during checkout.

3. Order Management

- On checkout:
  - Create a mock “placed order” entry.
  - Automatically update order status to **“Delivered”** after **6 hours**.
- Add an **Orders Screen** under the **Profile Menu**:
  - Display all previous orders with their status (Pending / Delivered / Cancelled).
  - Show delivery address and total amount.

4. Profile Menu

- Dropdown or side menu under the Profile icon containing:
  - **Profile** (just placeholder).
  - **My Orders** – navigates to Orders Screen.
  - **Logout** (just placeholder).

5. Header Updates

- Display **cart item count** badge over the cart icon.
- Retain other icons (Notifications, Profile).

6. State Management

- Implement **Redux Toolkit** for:
  - Cart Management.
  - Orders.

Backend – Ecommerce Website

Objective

Develop a backend for the Ecommerce website using **Spring Boot** and **PostgreSQL** that supports product management, promo code validation, and order processing.

Requirements

- Store a few sample **products** and their details in the database.
- Store **promo codes** with discount details and validity.
- Create necessary **APIs** to:
    - Fetch products (all, by ID, and by category).
    - Validate promo codes.
    - Create order with checkout details
    - Fetch orders
- During checkout, store order details along with address and applied promo code.
- Automatically mark an order as **Delivered** after **6 hours** from placement.
- Use proper **data validation**, **error handling**, and **layered architecture** (Controller, Service, Repository).
- Seed initial data for products and promo codes on startup.
