

Case Study: Product Management System (PMS)

1. Project Overview

Objective:

Build a **web-based Product Management System** where an admin can manage products (CRUD operations) and users can browse products, add them to cart, and checkout.

Technology Stack:

- Frontend: HTML5, CSS3, Bootstrap 5
- Backend: JSON Server or any REST API simulation
- Storage: LocalStorage for logged-in session
- Optional: Images for product catalog

User Roles:

1. **Admin** – Can manage products
2. **User** – Can browse products, add to cart, checkout

2. Pages and Features

A. Welcome Page

- Purpose: Landing page for PMS
- Features:
 - Welcome message and brand/logo
 - Buttons to **Login** and **Signup**
 - Navigation links (optional) to information pages

B. Signup Page

- Purpose: Allow new users to register
- Features:
 - Form fields: Full Name, Email, Password, Role (User/Admin)
 - Validation for required fields
 - Store user data in backend (JSON Server)
 - After successful signup → redirect to **Login Page**

C. Login Page

- Purpose: Authenticate users
- Features:
 - Form fields: Email, Password
 - Authentication via API (validate email/password)
 - Redirect based on role:
 - **Admin** → **Admin Dashboard**
 - **User** → **User Dashboard**
 - Save logged-in user in **LocalStorage**
 - Logout option accessible

D. Admin Dashboard

- Purpose: Manage product catalog
- Features:
 - Navbar: Dashboard, Add Product, Product List, Logout
 - **Add Product** page:
 - Form: Product Name, Description, Price, Image Upload
 - POST product to backend
 - **Product List** page:
 - Display products in cards or table
 - CRUD operations:
 - **Edit**: Pre-fill form, update API
 - **Delete**: Remove product from API
 - **View**: Optional detailed view of product
 - Logout: Clear session, redirect to Login

E. User Dashboard

- Purpose: Browse products and manage cart

- Features:
 - Navbar: Product List, Cart (show item count), Checkout, Logout
 - **Product List:**
 - Display products in card format
 - Show product name, description, price, image
 - **Add to Cart** button:
 - Increment count in cart (display in navbar)
 - Store cart in backend or localStorage
 - Logout: Clear session, redirect to Login

F. Cart / Checkout Page

- Purpose: Display products added to cart and calculate total
- Features:
 - Navbar consistent with user dashboard
 - Table of products in cart:
 - Columns: Product Name, Price
 - Total price calculation
 - Optional: Remove product from cart
 - Place order / checkout confirmation

G. Logout Functionality

- Purpose: Terminate user session
- Features:
 - Clear user session from localStorage
 - Redirect to Login or Welcome page
 - Ensure navigation links change dynamically based on login status

3. Navigation

- **Consistency:**
All pages include a **navbar** reflecting available actions for the current user role.

- **Admin Navbar:** Dashboard, Add Product, Product List, Logout
- **User Navbar:** Product List, Cart, Checkout, Logout
- Navigation should dynamically update based on login state

4. Backend / API Structure

- **Users:** (JSON format)

```
{
  "id": 1,
  "name": "John Doe",
  "email": "john@example.com",
  "password": "hashed_password",
  "role": "admin"
}
```

- **Products:** (JSON format)

```
{
  "id": 1,
  "name": "Smartphone",
  "description": "Latest model with advanced features",
  "price": 699,
  "image": "images/smartphone.png"
}
```

- **Cart:** Optional API or localStorage per user

5. Flow Summary

1. **Welcome → Signup / Login**
2. **Login → Redirect**
 - Admin → Admin Dashboard
 - User → User Dashboard
3. **Admin:** Manage products (CRUD)
4. **User:** Browse products → Add to cart → Checkout
5. **Logout** → Clear session → Welcome / Login