

Product Management System

Project description:

In this project, you will implement a product management system. There are two types of users in the system, one is a regular user, the other is an admin user. Each user type has different authorizations. Regular users can only select a product and add it to cart. Admin users can do everything regular users do, they can also edit/create/remove a product.

In real life, for example a B2C platform like Amazon, Walmart, Target. You can imagine the admin users are the product vendors, they can maintain the system to update their available products in stock. You can imagine the regular users are the regular customers that buy products. Company wants to save money and the system is designed to be shareable with two types of customers - vendors as internal customers and regular external customers.

Mock design: <https://www.figma.com/file/brgvADTppPXJdYkaOR5AmW/Management-Chuwa?node-id=819%3A521>

Phase I: Sign in/ Sign up flow

We need to authenticate what types of customers are logged in.

Requirement details:

1. three pages (sign in/sign up/ update password) using one reusable component, each page renders based on the only reusable component
2. enough validations for the input (either Frontend validation or Backend validation)
3. Log in/out api
4. sign up api
5. responsive design to support different devices screen

Bonus:

1. The whole project header and footer

Phase II: APIs & Product flow

Requirement details:

1. You should have the following apis working (Think about what's the modeled error as api response):
 - a. create a new product api
 - b. list products api
 - c. edit product api
2. Frontend product flow:
 - a. products list page - home page
 - b. create a new product page / edit an existing product page (reuse the same component if you can and make enough validation)
 - c. product detail page

Phase III: APIs & Cart flow

FINAL DEMO Integration Test requirements: (the last class demo)

Basic workflow

1. Authentication workflow
 - a. sign up flow with validation
 - b. sign in flow with validation
 - c. update password flow with validation
 - d. when user doesn't log in or login in as regular user, he can't see Add product button and Edit/Delete product button
 - e. when user log in as admin, he can see every feature
 - f. when page refreshes, all the user info can preserve (real api or mock using localStorage)
 - g. responsive design
2. Product workflow
 - a. after user creates a new product with validation, it should be shown immediately on product list page
 - b. After the user clicks on edit with validation, the page should pre-populate the production details in the form inputs.
 - c. Create/Edit product page should use the same component
 - d. User can add a product into cart and see the quantity on both product list page and product detail page
 - e. Pagination on product list page
 - f. Responsive design
 - g. test create/delete product with pagination
 - h. When a user logged in, its previous cart info will automatically show on the page (real api or mock using localStorage)
3. Cart workflow
 - a. user can edit product on cart info
 - b. cart info price calculation based on promotion code with validation
 - c. cart info should be consistent with product pages
 - d. When a user logged in, its previous cart info should be present (real api or mock using localStorage)
 - e. responsive design
4. Error handling
 - a. use error boundary to handle unexpected error from the root component
 - b. responsive design

Additional req: Search function on prod list page

Do what you think is right / efficient.

Do what you think makes sense to be a good customer experience.