BÁO CÁO ASSIGNMENT MÔN FER202



Thực hiện: Lê Văn Thiện  
MSSV: DE180492

# Mục lục

# 1. Giới thiệu dự án 2. Cấu trúc thư mục và các thành phần chính 3. Phân tích và giải thích mã nguồn 4. Lý do lựa chọn công nghệ 5. Kết luận

# 1. Giới thiệu dự án

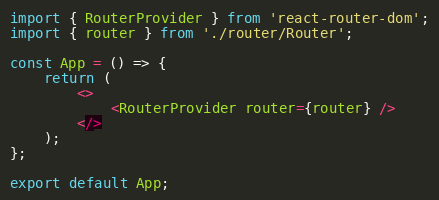
Dự án FERTECH là một nền tảng web độc đáo tập trung vào việc quản lý và theo dõi laptop một cách hiệu quả trong hệ thống quản lý kho. Ứng dụng cung cấp giao diện trực quan và dễ sử dụng để hiển thị danh sách sản phẩm, giúp người dùng dễ dàng nắm bắt tình trạng tồn kho.

# 2. Cấu trúc thư mục và các thành phần chính

Dự án được xây dựng dựa trên ReactJS và sử dụng json-server làm cơ sở dữ liệu giả lập. Các thành phần chính của dự án bao gồm:  
- Thư mục public/: chứa các tài nguyên tĩnh như ảnh sản phẩm, favicon, file HTML gốc.  
- Thư mục src/: chứa mã nguồn chính của ReactJS, gồm các file như App.js, index.js, App.css,...  
- File db.json: chứa dữ liệu mẫu về các sản phẩm laptop.  
- Các file cấu hình như package.json, .gitignore, README.md,...

# 3. Phân tích và giải thích mã nguồn

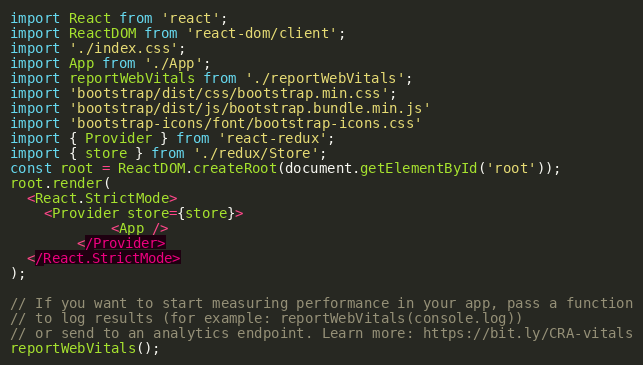
# 3.1. File App.js



File App.js là thành phần chính (main component) của ứng dụng ReactJS. Nó sử dụng RouterProvider từ thư viện react-router-dom để xử lý định tuyến (routing). Cấu trúc tệp như sau:

import { RouterProvider } from 'react-router-dom';  
import { router } from './router/Router';  
  
const App = () => {  
 return (  
 <>  
 <RouterProvider router={router} />  
 </>  
 );  
};  
  
export default App;

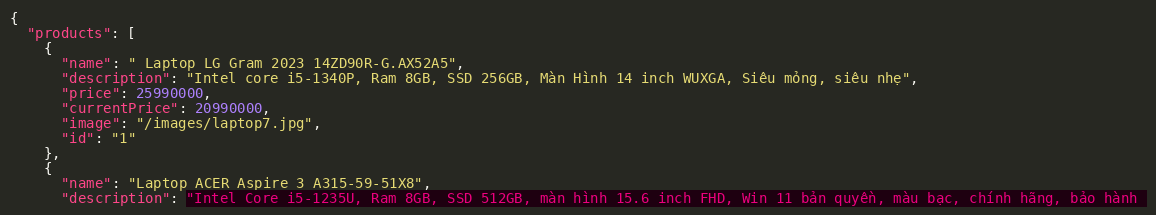
# 3.2. File index.js



Đây là file khởi tạo chính cho ứng dụng ReactJS. Nó kết nối ứng dụng với Redux store, và import các file style từ Bootstrap và Bootstrap Icons:

import React from 'react';  
import ReactDOM from 'react-dom/client';  
import './index.css';  
import App from './App';  
import reportWebVitals from './reportWebVitals';  
import 'bootstrap/dist/css/bootstrap.min.css';  
import 'bootstrap/dist/js/bootstrap.bundle.min.js';  
import 'bootstrap-icons/font/bootstrap-icons.css';  
import { Provider } from 'react-redux';  
import { store } from './redux/Store';  
  
const root = ReactDOM.createRoot(document.getElementById('root'));  
root.render(  
 <React.StrictMode>  
 <Provider store={store}>  
 <App />  
 </Provider>  
 </React.StrictMode>  
);  
  
reportWebVitals();

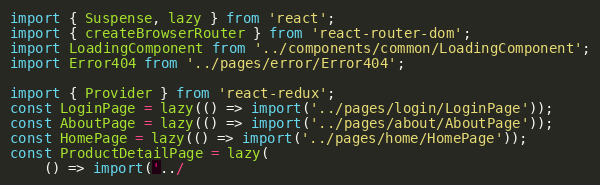
# 3.3. File db.json



Đây là file dữ liệu giả lập dùng cho json-server. Nó mô phỏng danh sách các sản phẩm laptop trong kho, mỗi sản phẩm gồm: tên, mô tả, giá gốc, giá khuyến mãi, đường dẫn ảnh và ID.

{  
 "products": [  
 {  
 "name": "Laptop LG Gram 2023...",  
 "description": "...",  
 "price": 25990000,  
 "currentPrice": 20990000,  
 "image": "/images/laptop7.jpg",  
 "id": "1"  
 },  
 ...  
 ]  
}

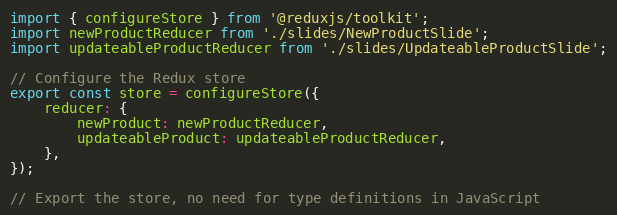
# 3.4. File router/Router.js



Đây là nơi định nghĩa các tuyến đường (route) cho toàn bộ ứng dụng, sử dụng thư viện react-router-dom. File này sử dụng lazy-loading để tải các trang khi cần thiết, giúp tối ưu hiệu năng. Cấu trúc cơ bản như sau:

import { Suspense, lazy } from 'react';  
import { createBrowserRouter } from 'react-router-dom';  
import LoadingComponent from '../components/common/LoadingComponent';  
...  
const LoginPage = lazy(() => import('../pages/login/LoginPage'));  
...  
export const router = createBrowserRouter([  
 {  
 path: '/',  
 element: (  
 <Suspense fallback={<LoadingComponent />}>  
 <HomePage />  
 </Suspense>  
 ),  
 },  
 ...  
]);

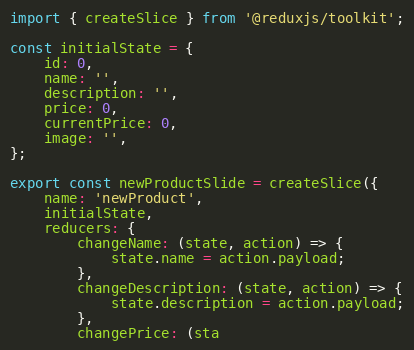
# 3.5. File redux/Store.js



File này thiết lập Redux store cho toàn bộ ứng dụng bằng cách sử dụng configureStore từ @reduxjs/toolkit. Nó khai báo hai reducer là newProduct và updateableProduct để quản lý trạng thái tương ứng:

import { configureStore } from '@reduxjs/toolkit';  
import newProductReducer from './slides/NewProductSlide';  
import updateableProductReducer from './slides/UpdateableProductSlide';  
  
export const store = configureStore({  
 reducer: {  
 newProduct: newProductReducer,  
 updateableProduct: updateableProductReducer,  
 },  
});

# 3.6. File redux/slides/NewProductSlide.js



File này quản lý trạng thái của sản phẩm mới được thêm vào hệ thống. Sử dụng createSlice từ @reduxjs/toolkit để tạo reducer và action tương ứng. State ban đầu gồm các thuộc tính như id, name, description, price, currentPrice, image. Một số action điển hình:

- changeName, changeDescription: cập nhật tên và mô tả sản phẩm  
- changePrice, changeCurrentPrice: cập nhật giá gốc và giá khuyến mãi (kiểm tra > 0)  
- changeImage: cập nhật đường dẫn ảnh  
- reset: đưa state về giá trị mặc định

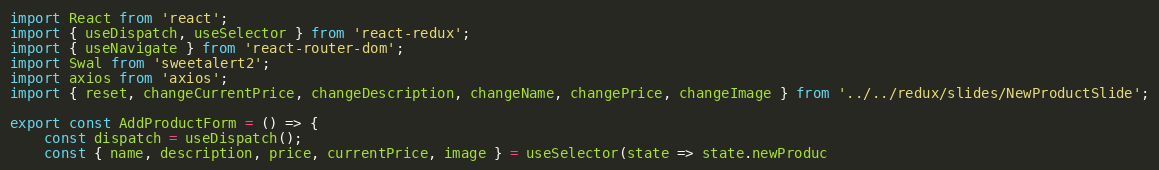
# 3.7. File redux/slides/UpdateableProductSlide.js



Tương tự như NewProductSlide, file này dùng để quản lý thông tin sản phẩm đang được chỉnh sửa. Ngoài các action giống bên trên, còn có thêm changeId để cập nhật ID sản phẩm hiện tại. reset sẽ xóa toàn bộ thông tin của sản phẩm cần cập nhật.

# 3.8. Các thành phần trong components/product

# 3.8.1. AddProductForm.js



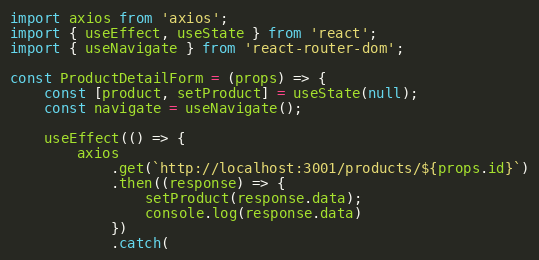
Form dùng để thêm sản phẩm mới. Sử dụng Redux để lấy và cập nhật state từ newProduct. Có kiểm tra hợp lệ (validate) cho tên, mô tả, giá gốc và giá ưu đãi. Sau khi thêm sản phẩm thành công, ứng dụng sẽ chuyển hướng về danh sách và hiện thông báo bằng SweetAlert.

# 3.8.2. ProductCard.js



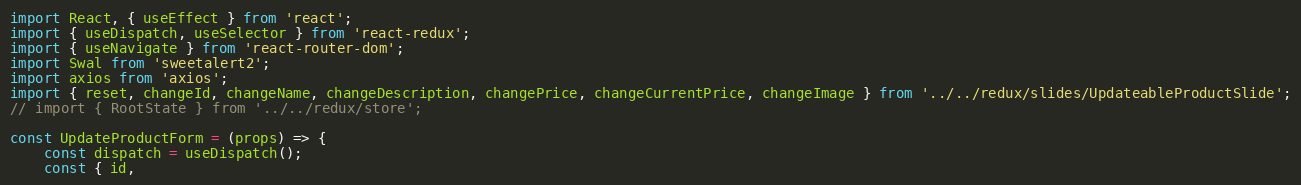
Hiển thị sản phẩm dạng card bao gồm ảnh, tên, mô tả, giá. Có hiệu ứng hover phóng to nhẹ. Dùng CSS để rút gọn nội dung dài. Giá tiền được định dạng theo kiểu Việt Nam.

# 3.8.3. ProductDetailForm.js



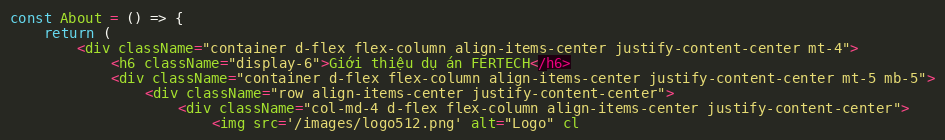
Hiển thị chi tiết sản phẩm thông qua ID truyền vào từ props. Lấy dữ liệu qua axios. Gồm tính năng định dạng giá và tính phần trăm giảm giá. Giao diện có thể được dùng ở modal hoặc trang riêng.

# 3.8.4. UpdateProductForm.js



Tương tự AddProductForm, nhưng dùng để chỉnh sửa sản phẩm hiện có. Dữ liệu được load từ server dựa vào ID, lưu vào Redux (updateableProduct). Sau khi cập nhật thành công cũng có thông báo và chuyển hướng như form thêm mới.

# 3.9. File components/about/About.js



Đây là thành phần giao diện dùng để giới thiệu tổng quan về dự án FERTECH. Nội dung hiển thị gồm tiêu đề, đoạn mô tả về mục tiêu của dự án và danh sách các công nghệ được sử dụng. Giao diện được chia bố cục hợp lý với ảnh logo và đoạn văn mô tả ở hai cột (sử dụng Bootstrap).

Các công nghệ được liệt kê trong phần giới thiệu bao gồm: React, Redux, React Router, Bootstrap và json-server. Thành phần này đóng vai trò như một trang 'About Us' truyền thống trong các ứng dụng web.

# 4. Lý do lựa chọn công nghệ

Dự án FERTECH được xây dựng dựa trên nền tảng các công nghệ hiện đại, linh hoạt và phổ biến trong phát triển web hiện nay. Việc lựa chọn các công nghệ này xuất phát từ nhu cầu xây dựng một hệ thống quản lý kho thân thiện với người dùng, dễ mở rộng, dễ bảo trì:

- ReactJS: thư viện JavaScript mạnh mẽ giúp xây dựng giao diện người dùng mượt mà, phản hồi nhanh.  
- Redux Toolkit: hỗ trợ quản lý trạng thái tập trung, giúp ứng dụng ổn định khi mở rộng.  
- React Router: hỗ trợ điều hướng nhiều trang trong ứng dụng SPA.  
- Bootstrap: thư viện CSS phổ biến giúp tạo giao diện đẹp và chuẩn responsive.  
- json-server: công cụ tạo REST API giả lập giúp mô phỏng nhanh hệ thống backend trong quá trình phát triển.

# 5. Kết luận

Dự án FERTECH là một hệ thống quản lý kho sản phẩm laptop có giao diện hiện đại, dễ sử dụng và đầy đủ tính năng như thêm, chỉnh sửa, xem chi tiết và xoá sản phẩm. Thông qua việc sử dụng ReactJS kết hợp với Redux và json-server, hệ thống đã mô phỏng được các thao tác quản lý sản phẩm hiệu quả. Báo cáo này đã trình bày cấu trúc, phân tích mã nguồn, cũng như lý do lựa chọn công nghệ nhằm giúp ban lãnh đạo hiểu rõ tiềm năng của dự án.