# Mô tả Dự Án: FruitShop Lite

## Giới thiệu

FruitShop Lite là một ứng dụng web thương mại điện tử bán trái cây tươi và các sản phẩm liên quan. Dự án tập trung vào việc xây dựng một hệ thống cơ bản nhưng đầy đủ các tính năng cần thiết, có phân quyền người dùng (Guest, User, Admin) và khả năng mở rộng.

## Các Tính Năng Chính

### 1. Guest (Người dùng chưa đăng nhập)

- Xem danh sách sản phẩm.

- Xem chi tiết sản phẩm.

- Đăng ký và đăng nhập để sử dụng các tính năng khác.

### 2. User (Người dùng đã đăng nhập)

- Tất cả tính năng của Guest.

- Thêm sản phẩm vào giỏ hàng (được lưu trên server).

- Thanh toán và tạo đơn hàng.

- Xem lịch sử đơn hàng.

### 3. Admin (Quản trị viên)

- Quản lý sản phẩm (thêm, sửa, xóa).

- Xem danh sách đơn hàng của tất cả người dùng.

- Cập nhật trạng thái đơn hàng (pending, delivered).

## Cách Hoạt Động

1. Phân quyền Role:

- Guest: Không cần đăng nhập.

- User: Đăng nhập bằng tài khoản với role user.

- Admin: Đăng nhập bằng tài khoản với role admin.

2. Header thay đổi theo Role:

- Guest: Hiển thị menu cơ bản (Trang chủ, Sản Phẩm, Đăng nhập/Đăng ký).

- User: Hiển thị thêm Giỏ hàng và Lịch sử đơn hàng.

- Admin: Hiển thị thêm Quản lý sản phẩm và Quản lý đơn hàng.

3. Xác thực JWT:

- Người dùng đăng nhập nhận token JWT, dùng token để truy cập các API.

- Middleware xác minh token trên backend để phân quyền truy cập.

## Công Nghệ Sử Dụng

1. Frontend: Vue.js - Quản lý giao diện, navigation, và trạng thái (Pinia/Vuex).

2. Backend: Express.js - Xử lý logic nghiệp vụ và cung cấp API RESTful.

3. Database: MySQL - Quản lý dữ liệu người dùng, sản phẩm, đơn hàng.

4. Authentication: JWT - Bảo vệ các API và phân quyền theo role.

## Kiến Trúc Dự Án

FruitShopLite/  
├── backend/  
│ ├── src/  
│ │ ├── controllers/ # Logic nghiệp vụ (Auth, Product, Order)  
│ │ ├── models/ # Định nghĩa bảng MySQL  
│ │ ├── routes/ # Định nghĩa các API endpoint  
│ │ ├── middlewares/ # Xử lý xác thực và phân quyền  
│ └── app.js # Entry point chính của backend  
├── frontend/  
│ ├── src/  
│ │ ├── components/ # Component giao diện (Header, Footer, ...)  
│ │ ├── pages/ # Các trang chính (Home, Product, Admin)  
│ │ ├── store/ # Quản lý trạng thái ứng dụng  
│ │ └── App.vue # Gốc ứng dụng Vue.js  
└── README.md

## Quy trình hoạt động

1. Người dùng truy cập trang web:

- Nếu chưa đăng nhập (Guest), chỉ có thể xem sản phẩm.

2. Đăng nhập:

- Người dùng nhập thông tin đăng nhập.

- Backend xác thực và trả về token JWT chứa role (user/admin).

3. Sử dụng tính năng theo Role:

- Guest: Không cần token, truy cập được trang công khai.

- User: Cần token hợp lệ, truy cập tính năng khách hàng.

- Admin: Token hợp lệ và role là admin, truy cập trang quản trị.

## Điểm Nổi Bật

- Đơn giản và dễ hiểu: Hệ thống được tối ưu chỉ với các tính năng cốt lõi.

- Phân quyền rõ ràng: Guest, User, và Admin chỉ thấy và truy cập được các tính năng phù hợp.

- Dễ mở rộng: Có thể thêm chatbot, recommendation system hoặc các tính năng nâng cao khác sau này.

## Kế Hoạch Mở Rộng

- Chatbot: Tư vấn sản phẩm và hỗ trợ khách hàng.

- Recommendation System: Gợi ý sản phẩm dựa trên hành vi mua sắm.

- Thanh toán trực tuyến: Tích hợp cổng thanh toán như PayPal, Stripe.