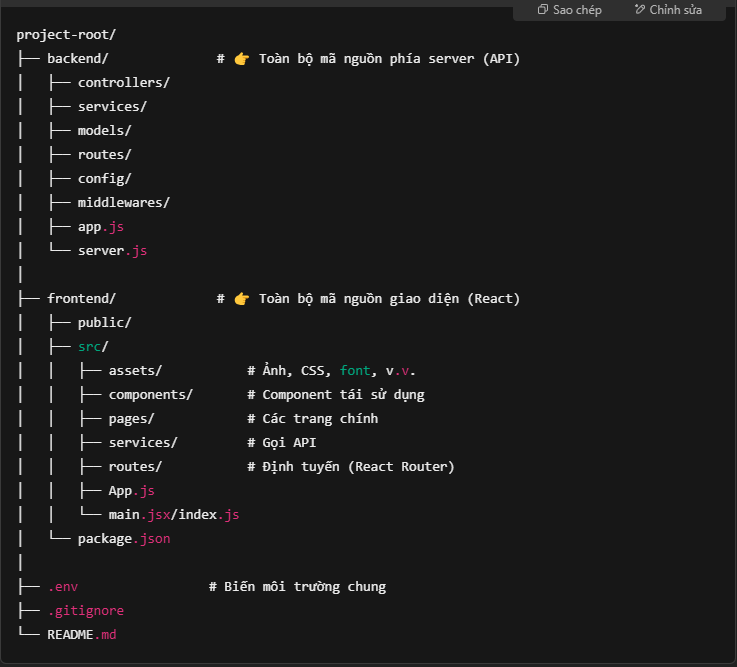
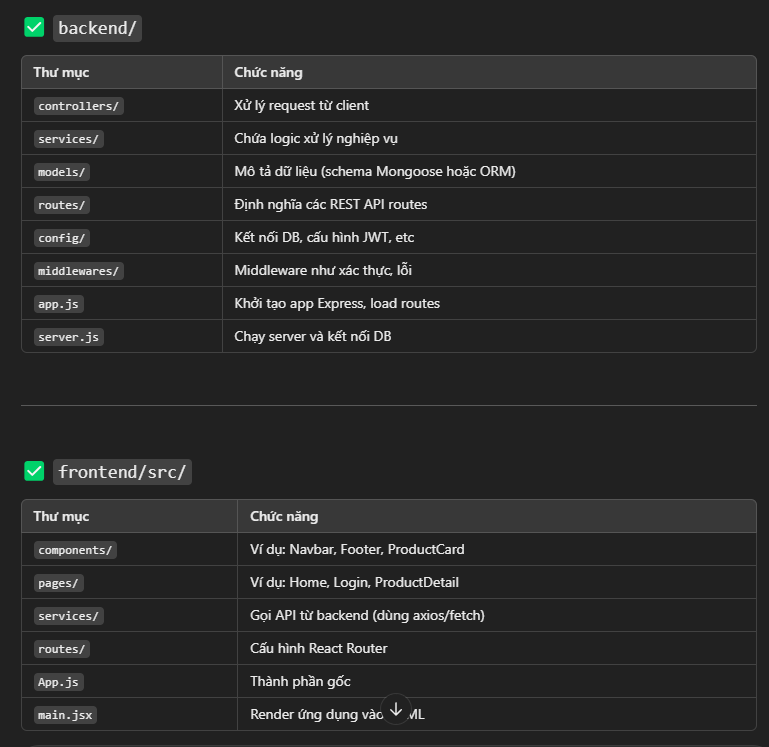
**Mô hình MVC REST API (Backend) và SPA bằng React (Frontend):**





Cách vận hành của mô hình MVC + REST API + React SPA hoạt động theo quy trình rõ ràng, tách biệt frontend và backend như sau:

🧭 Tổng quan cách vận hành

css

Sao chépChỉnh sửa

[User] 👈👉 [Frontend React] 👈👉 [REST API - Backend Express] 👈👉 [Database]

🔁 Luồng xử lý yêu cầu (request flow)

🔹 1. Người dùng tương tác với frontend (React)

Ví dụ: Click vào “Xem sản phẩm”, điền form đăng nhập, thêm vào giỏ hàng…

🔹 2. Frontend gửi request API đến Backend

* Dùng fetch() hoặc axios để gọi API REST từ backend.

js

Sao chépChỉnh sửa

// Ví dụ:

axios.get('http://localhost:3000/api/products')

🔹 3. Backend xử lý theo mô hình MVC

* Router nhận request, chuyển sang Controller.
* Controller gọi Service để xử lý logic.
* Service tương tác với Model (DB) để lấy hoặc lưu dữ liệu.

🔹 4. Model truy vấn database

* Dùng ORM như Sequelize, Mongoose hoặc query SQL thô.

🔹 5. Dữ liệu trả về frontend

* Dữ liệu từ DB → Service xử lý → Controller trả response JSON → React nhận được và hiển thị.

💬 Ví dụ cụ thể – lấy danh sách sản phẩm

🧑‍💻 Người dùng vào trang “Sản phẩm”:

1. React gọi API:  
   GET http://localhost:3000/api/products
2. Backend xử lý:

bash

Sao chépChỉnh sửa

routes/productRoutes.js

-> productController.getAllProducts

-> productService.getAll()

-> ProductModel.findAll()

1. Trả về dữ liệu JSON:

json

Sao chépChỉnh sửa

[

{ "id": 1, "name": "Áo thun", "price": 150000 },

{ "id": 2, "name": "Giày thể thao", "price": 350000 }

]

1. React hiển thị lên UI:

jsx

Sao chépChỉnh sửa

<ProductCard name="Áo thun" price={150000} />

🔒 Xử lý đăng nhập / bảo mật (Auth)

1. React gửi thông tin đăng nhập → Backend kiểm tra → tạo JWT Token.
2. React lưu token vào localStorage.
3. Gọi API sau đó luôn đính kèm token vào header:

http

Sao chépChỉnh sửa

Authorization: Bearer <token>

1. Backend dùng middleware để xác thực.

✅ Ưu điểm của mô hình này:

* Tách biệt frontend và backend → dễ bảo trì, dễ chia team.
* Frontend chỉ lo giao diện, backend chỉ lo logic + dữ liệu.
* Backend có thể tái sử dụng cho mobile app, admin app.
* Dễ scale thành mô hình lớn hơn (microservices, serverless…).