**📂 Cấu trúc project React**

**foodfast-ui/**

**│── Dockerfile**

**│── package.json**

**│── vite.config.js**

**│── public/**

**│── src/**

**│ ├── main.jsx**

**│ ├── App.jsx**

**│ ├── api/**

**│ │ ├── axiosClient.js**

**│ │ ├── userApi.js**

**│ │ ├── productApi.js**

**│ │ ├── orderApi.js**

**│ │ └── paymentApi.js**

**│ ├── components/**

**│ │ ├── Navbar.jsx**

**│ │ └── ProductCard.jsx**

**│ ├── pages/**

**│ │ ├── Home.jsx**

**│ │ ├── Login.jsx**

**│ │ ├── Register.jsx**

**│ │ ├── Cart.jsx**

**│ │ ├── Orders.jsx**

**│ │ └── Payment.jsx**

**│ └── store/**

**│ └── cartStore.js (Zustand hoặc Redux)**

**📦 Đề án: FoodFast Delivery (Microservices Architecture)**

**1. Kiến trúc tổng thể**

**Hệ thống được chia thành 4 dịch vụ backend + 1 frontend web:**

**[ React Web App (FoodFast UI) ]**

**↓**

**[API Gateway]**

**↓**

**┌────────────────────────────┐**

**│ User Service → user-mongo│**

**│ Product Service→ product-mongo│**

**│ Order Service → order-mongo │**

**│ Payment Service→ payment-mongo│**

**└────────────────────────────┘**

* **Frontend: React (hiển thị UI cho người dùng, gọi API qua API Gateway).**
* **Backend: 4 dịch vụ Node.js (Express + MongoDB).**
* **Database: mỗi service có Mongo riêng (cách ly dữ liệu).**
* **Triển khai: Docker Compose (local) → có thể nâng cấp Kubernetes (production).**
* **API Gateway: Nginx/Kong/Traefik (gom API, quản lý auth, rate limiting).**

**2. Các Service**

**🧑 User Service**

* **Quản lý người dùng: đăng ký, đăng nhập, thông tin cá nhân.**
* **DB: users collection.**
* **API:**
  + **POST /users → tạo user**
  + **GET /users/:id → lấy thông tin user**
  + **POST /auth/login → đăng nhập**

**🍔 Product Service**

* **Quản lý sản phẩm (món ăn).**
* **DB: products collection.**
* **API:**
  + **POST /products → thêm sản phẩm**
  + **GET /products → danh sách sản phẩm**
  + **GET /products/:id → chi tiết sản phẩm**

**📦 Order Service**

* **Quản lý đơn hàng.**
* **Liên kết với User Service (người đặt) + Product Service (món ăn).**
* **DB: orders collection.**
* **API:**
  + **POST /orders → tạo đơn hàng (check user + product tồn tại)**
  + **GET /orders/:id → chi tiết đơn hàng**

**💳 Payment Service**

* **Xử lý thanh toán.**
* **Có thể async bằng message queue (RabbitMQ/Kafka) nếu cần.**
* **DB: payments collection.**
* **API:**
  + **POST /payments → thực hiện thanh toán (liên kết Order).**
  + **GET /payments/:id → trạng thái thanh toán.**

**3. Docker Compose (tóm tắt)**

**version: '3.8'**

**services:**

**user-service:**

**build: ./user-service**

**ports: ["4001:4001"]**

**environment:**

**- MONGO\_URL=mongodb://user-mongo:27017/users**

**depends\_on: [user-mongo]**

**product-service:**

**build: ./product-service**

**ports: ["4002:4002"]**

**environment:**

**- MONGO\_URL=mongodb://product-mongo:27017/products**

**depends\_on: [product-mongo]**

**order-service:**

**build: ./order-service**

**ports: ["4003:4003"]**

**environment:**

**- MONGO\_URL=mongodb://order-mongo:27017/orders**

**depends\_on: [order-mongo]**

**payment-service:**

**build: ./payment-service**

**ports: ["4004:4004"]**

**environment:**

**- MONGO\_URL=mongodb://payment-mongo:27017/payments**

**depends\_on: [payment-mongo]**

**# MongoDB instances**

**user-mongo:**

**image: mongo:5**

**volumes: [ "user\_data:/data/db" ]**

**product-mongo:**

**image: mongo:5**

**volumes: [ "product\_data:/data/db" ]**

**order-mongo:**

**image: mongo:5**

**volumes: [ "order\_data:/data/db" ]**

**payment-mongo:**

**image: mongo:5**

**volumes: [ "payment\_data:/data/db" ]**

**volumes:**

**user\_data:**

**product\_data:**

**order\_data:**

**payment\_data:**

**4. Frontend (React)**

**Công nghệ:**

* **React + Vite (nhẹ, nhanh).**
* **Axios để gọi API.**
* **React Router cho navigation.**
* **Redux/Zustand để quản lý state (cart, user login).**

**Các trang chính:**

* **Trang chủ: hiển thị danh sách món ăn (Product Service).**
* **Đăng ký/Đăng nhập: kết nối User Service.**
* **Giỏ hàng / Order: tạo đơn hàng (Order Service).**
* **Thanh toán: thực hiện giao dịch (Payment Service).**
* **Quản trị (Admin): thêm/sửa sản phẩm (Product Service).**

**Ví dụ gọi API trong React:**

**import axios from "axios";**

**const API\_URL = "http://localhost:8080"; // API Gateway**

**export async function getProducts() {**

**const res = await axios.get(`${API\_URL}/products`);**

**return res.data;**

**}**

**5. Cải tiến & tối ưu**

* **API Gateway: gom toàn bộ API, dễ quản lý.**
* **Bảo mật: dùng JWT cho auth, bcrypt cho password hash.**
* **Monitoring: Prometheus + Grafana, Winston/Morgan log.**
* **CI/CD: GitHub Actions → build & deploy Docker.**
* **Scale: chuyển sang Kubernetes khi cần.**

**⚙️ Dockerfile (React UI)**

**FROM node:18-alpine**

**WORKDIR /app**

**COPY package.json package-lock.json ./**

**RUN npm install**

**COPY . .**

**EXPOSE 5173**

**CMD ["npm", "run", "dev", "--", "--host"]**

**📦 package.json (React + Vite + Axios + Zustand)**

**{**

**"name": "foodfast-ui",**

**"version": "1.0.0",**

**"scripts": {**

**"dev": "vite",**

**"build": "vite build",**

**"preview": "vite preview"**

**},**

**"dependencies": {**

**"axios": "^1.6.2",**

**"react": "^18.2.0",**

**"react-dom": "^18.2.0",**

**"react-router-dom": "^6.22.2",**

**"zustand": "^4.5.0"**

**},**

**"devDependencies": {**

**"@vitejs/plugin-react": "^4.0.3",**

**"vite": "^5.0.0"**

**}**

**}**

**🛠 src/api/axiosClient.js**

**import axios from "axios";**

**const axiosClient = axios.create({**

**baseURL: "http://localhost:8080", // API Gateway**

**headers: {**

**"Content-Type": "application/json",**

**},**

**});**

**export default axiosClient;**

**🛠 src/api/productApi.js**

**import axiosClient from "./axiosClient";**

**const productApi = {**

**getAll: () => axiosClient.get("/products"),**

**getById: (id) => axiosClient.get(`/products/${id}`),**

**};**

**export default productApi;**

**(Tương tự tạo userApi.js, orderApi.js, paymentApi.js).**

**🖼 src/components/Navbar.jsx**

**import { Link } from "react-router-dom";**

**export default function Navbar() {**

**return (**

**<nav className="p-4 bg-green-600 text-white flex justify-between">**

**<Link to="/">FoodFast</Link>**

**<div className="space-x-4">**

**<Link to="/cart">Giỏ hàng</Link>**

**<Link to="/orders">Đơn hàng</Link>**

**<Link to="/login">Đăng nhập</Link>**

**</div>**

**</nav>**

**);**

**}**

**🏠 src/pages/Home.jsx**

**import { useEffect, useState } from "react";**

**import productApi from "../api/productApi";**

**import ProductCard from "../components/ProductCard";**

**export default function Home() {**

**const [products, setProducts] = useState([]);**

**useEffect(() => {**

**productApi.getAll().then((res) => setProducts(res.data));**

**}, []);**

**return (**

**<div className="p-4 grid grid-cols-3 gap-4">**

**{products.map((p) => (**

**<ProductCard key={p.\_id} product={p} />**

**))}**

**</div>**

**);**

**}**

**🛒 src/components/ProductCard.jsx**

**import useCartStore from "../store/cartStore";**

**export default function ProductCard({ product }) {**

**const addToCart = useCartStore((s) => s.add);**

**return (**

**<div className="border p-3 rounded shadow">**

**<h3 className="font-bold">{product.name}</h3>**

**<p>{product.price} VND</p>**

**<button**

**onClick={() => addToCart(product)}**

**className="bg-green-500 text-white px-3 py-1 rounded"**

**>**

**Thêm vào giỏ**

**</button>**

**</div>**

**);**

**}**

**🛍 src/store/cartStore.js**

**import { create } from "zustand";**

**const useCartStore = create((set) => ({**

**cart: [],**

**add: (item) =>**

**set((state) => ({**

**cart: [...state.cart, item],**

**})),**

**clear: () => set({ cart: [] }),**

**}));**

**export default useCartStore;**

**🚀 src/main.jsx**

**import React from "react";**

**import ReactDOM from "react-dom/client";**

**import { BrowserRouter, Routes, Route } from "react-router-dom";**

**import App from "./App";**

**import Home from "./pages/Home";**

**import Login from "./pages/Login";**

**import Register from "./pages/Register";**

**import Cart from "./pages/Cart";**

**import Orders from "./pages/Orders";**

**import Payment from "./pages/Payment";**

**ReactDOM.createRoot(document.getElementById("root")).render(**

**<BrowserRouter>**

**<Routes>**

**<Route path="/" element={<App />}>**

**<Route index element={<Home />} />**

**<Route path="login" element={<Login />} />**

**<Route path="register" element={<Register />} />**

**<Route path="cart" element={<Cart />} />**

**<Route path="orders" element={<Orders />} />**

**<Route path="payment" element={<Payment />} />**

**</Route>**

**</Routes>**

**</BrowserRouter>**

**);**

**🚀 src/App.jsx**

**import { Outlet } from "react-router-dom";**

**import Navbar from "./components/Navbar";**

**export default function App() {**

**return (**

**<div>**

**<Navbar />**

**<Outlet />**

**</div>**

**);**

**}**

**Hướng dẫn chạy:**

* **Chạy React frontend bằng Docker (docker build -t foodfast-ui . && docker run -p 5173:5173 foodfast-ui).**
* **Gọi API backend qua API Gateway (http://localhost:8080/...).**
* **Có sẵn state quản lý giỏ hàng (Zustand).**
* **Có thể mở rộng dễ dàng (thêm đăng nhập, đơn hàng, thanh toán).**