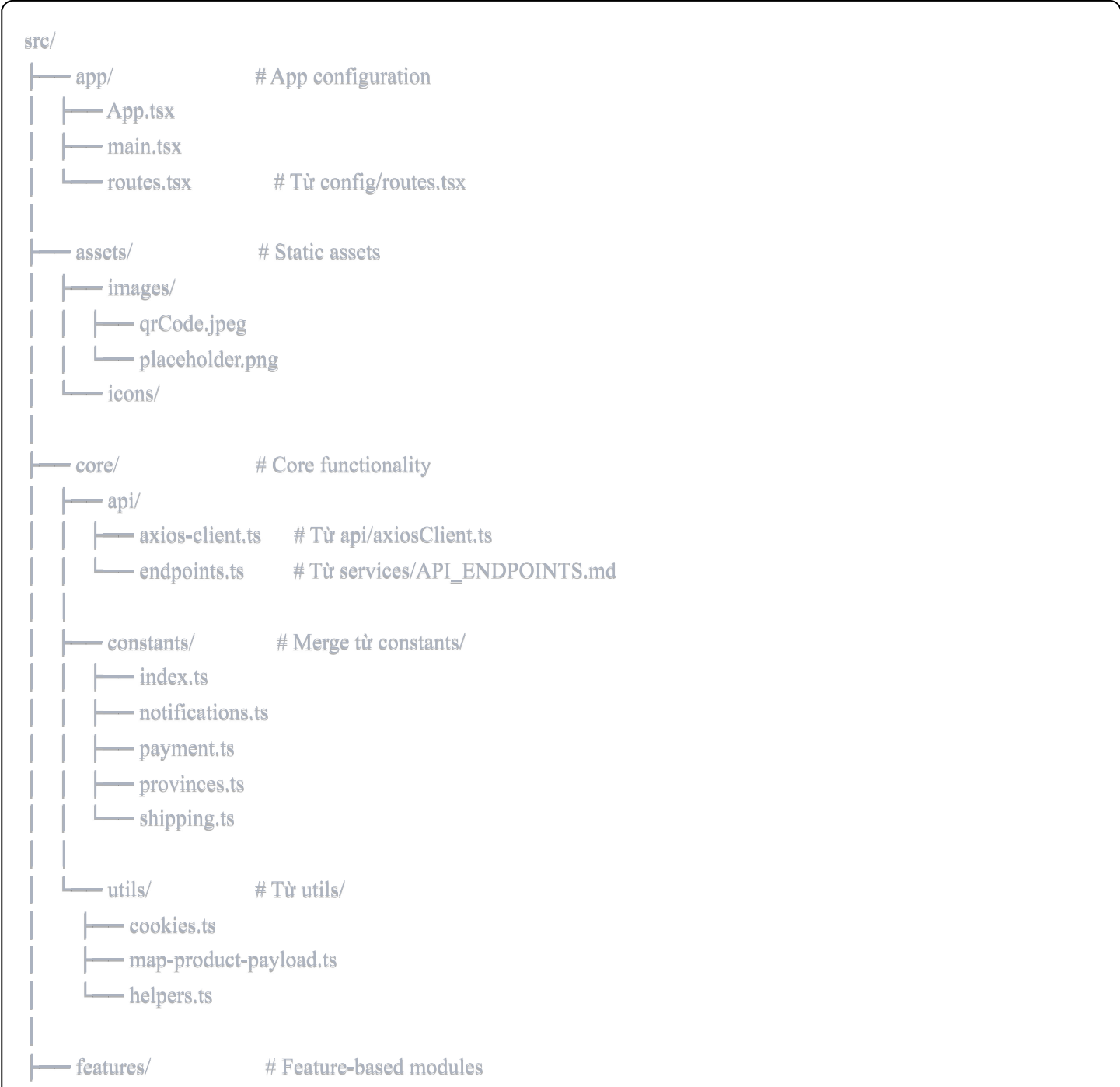


# Kế hoạch Refactor Cấu trúc Frontend

## Mục tiêu

- Tổ chức code rõ ràng theo chức năng
- Giảm trùng lặp và conflict
- Dễ scale và maintain
- Tuân thủ best practices

## Cấu trúc Đề xuất



- auth/
  - components/
    - LoginForm.tsx
    - LogoutModal.tsx
    - ProtectedRoute.tsx
    - PublicRoute.tsx
  - hooks/
    - use-auth.ts
  - services/
    - auth-api.ts
    - role-api.ts
  - types/
    - auth.types.ts # Tır types/account.types.ts
  - pages/
    - LoginPage.tsx
    - index.ts

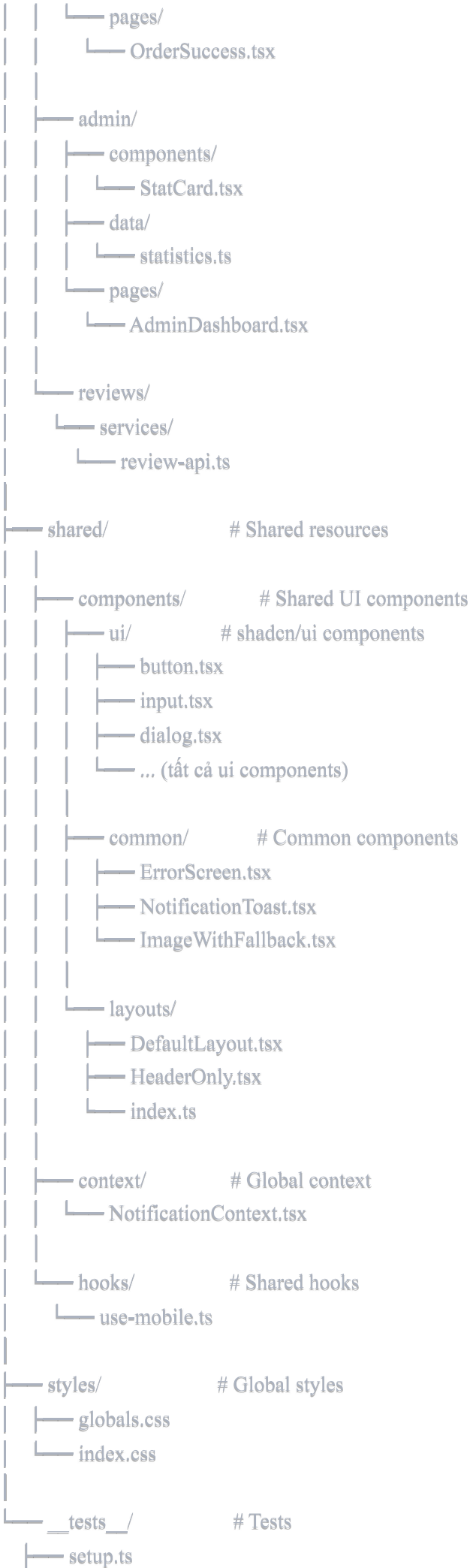
- cart/
  - components/
    - CartPage.tsx
    - CartReview.tsx
    - CartItem.tsx
  - context/
    - CartContext.tsx
  - hooks/
    - use-cart.ts
    - use-cart-actions.ts
    - use-cart-calculator.ts
    - use-cart-identity.ts
  - services/
    - cart-api.ts
  - types/
    - cart.types.ts
  - pages/
    - CartPage.tsx

- checkout/
  - components/
    - ShippingForm.tsx
    - PaymentMethodV2.tsx
    - OrderSummary.tsx
    - StepIndicator.tsx
  - context/
    - CheckoutContext.tsx
  - hooks/

- └─ use-payment.ts
- └─ use-order.ts
- └─ services/
  - └─ payment-api.ts
  - └─ delivery-api.ts
- └─ types/
  - └─ checkout.types.ts
- └─ pages/
  - └─ CheckoutPage.tsx
  - └─ CheckoutFlow.tsx
  - └─ OrderSuccess.tsx
  - └─ PaymentFailedPage.tsx
  - └─ index.ts

- └─ products/
  - └─ components/
    - └─ ProductCard.tsx
    - └─ ProductGrid.tsx
    - └─ ProductFilters.tsx
    - └─ ProductForm.tsx
    - └─ ProductItem.tsx
  - └─ hooks/
    - └─ use-product.ts
    - └─ use-goods.ts
  - └─ services/
    - └─ product-api.ts
    - └─ goods-api.ts
  - └─ types/
    - └─ product.types.ts
  - └─ data/
    - └─ mock-products.ts
    - └─ products.ts
  - └─ pages/
    - └─ ProductDetails.tsx
    - └─ ProductManager.tsx
    - └─ index.ts

- └─ orders/
  - └─ components/
    - └─ OrderSummary.tsx
  - └─ services/
    - └─ order-api.ts
  - └─ types/
    - └─ order.types.ts
  - └─ data/
    - └─ orders.ts



```
|— checkout-back-navigation.test.ts
|— logout-functionality.test.ts
```

## Chi tiết Thay đổi

### 1. Loại bỏ Trùng lặp

Vấn đề:

- `CartPage.tsx` xuất hiện ở cả `components/` và `pages/Cart/`
- `ShippingForm.tsx`, `PaymentMethod.tsx` trùng lặp
- `OrderSummary.tsx` và `OrderSuccess.tsx` trùng lặp

Giải pháp:

- ✓ GIỮ: `features/cart/pages/CartPage.tsx`
- ✗ XÓA: `components/CartPage.tsx`
- ✓ GIỮ: `features/checkout/components/ShippingForm.tsx`
- ✗ XÓA: `pages/Checkout/ShippingForm.tsx`
- ✓ GIỮ: `features/checkout/pages/OrderSuccess.tsx`
- ✗ XÓA: `components/OrderSuccess.tsx`, `pages/OrderSuccess/`

### 2. Tổ chức Services

Trước:

```
services/
|— cartApi.ts      # ✗ Trùng với carts/cartApi.ts
|— paymentApi.ts   # ✗ Trùng với payment/paymentApi.ts
|— carts/cartApi.ts
|— payment/paymentApi.ts
|— ...
```

Sau:

features/

```
|— cart/services/cart-api.ts
|— checkout/services/
|   |— payment-api.ts
|   |— delivery-api.ts
|— products/services/
|   |— product-api.ts
|   |— goods-api.ts
```

### 3. Đổi tên File theo Naming Convention

camelCase (❌) → kebab-case (✅)

axiosClient.ts → axios-client.ts

CartContext.tsx → cart-context.tsx

useCart.ts → use-cart.ts

cartApi.ts → cart-api.ts

mockProducts.ts → mock-products.ts

### 4. Tách Shared Components

**UI Components (shadcn/ui):**

shared/components/ui/

```
|— button.tsx
|— dialog.tsx
|— input.tsx
|— ... (giữ nguyên)
```

**Common Components:**

shared/components/common/

```
|— ErrorScreen.tsx
|— NotificationToast.tsx
|— ImageWithFallback.tsx
```

**Layouts:**

shared/components/layouts/

```
|— DefaultLayout.tsx
|— HeaderOnly.tsx
```

## 5. Tổ chức Hooks

### Feature-specific hooks:

```
features/cart/hooks/  
├── use-cart.ts  
├── use-cart-actions.ts  
├── use-cart-calculator.ts  
└── use-cart-identity.ts
```

### Shared hooks:

```
shared/hooks/  
└── use-mobile.ts
```

## 6. Data & Mock

```
features/products/data/  
├── mock-products.ts  # Mock data  
└── products.ts      # Real data logic  
  
features/orders/data/  
└── orders.ts  
  
features/admin/data/  
└── statistics.ts
```

---

## Action Items

### Phase 1: Cleanup & Restructure

- ☐ Xóa các file trùng lặp
- ☐ Tạo cấu trúc thư mục mới
- ☐ Di chuyển files theo feature

### Phase 2: Refactor Imports

- ☐ Cập nhật tất cả import paths
- ☐ Tạo barrel exports (index.ts) cho mỗi feature
- ☐ Cập nhật tsconfig paths

### Phase 3: Naming Convention

- ☐ Đổi tên files sang kebab-case
- ☐ Đổi tên folders sang kebab-case
- ☐ Cập nhật imports

### Phase 4: Testing

- ☐ Di chuyển tests vào `tests/`
- ☐ Cập nhật test imports
- ☐ Chạy test suite

## 📁 Lợi ích

- ✅ **Clear Separation of Concerns:** Mỗi feature độc lập
  - ✅ **Easy Navigation:** Tìm file dễ dàng theo feature
  - ✅ **Scalable:** Thêm feature mới không ảnh hưởng code cũ
  - ✅ **Reusability:** Shared components rõ ràng
- Type Safety:** Types gắn với code sử dụng   ✅ **Testing:** Tests tổ chức theo feature

## 📄 Quy tắc Coding

### Import Order

```
typescript

// 1. External libraries
import React from 'react';
import { useNavigate } from 'react-router-dom';

// 2. Internal core
import { axiosClient } from '@core/api';
import { ROUTES } from '@core/constants';

// 3. Shared
import { Button } from '@shared/components/ui';
import { DefaultLayout } from '@shared/components/layouts';

// 4. Feature
import { useCart } from '../hooks/use-cart';
import type { CartItem } from '../types/cart.types';

// 5. Relative imports
import { CartItem } from './CartItem';
```



## File Naming

- Components: `PascalCase.tsx` → `ProductCard.tsx`
- Utilities: `kebab-case.ts` → `format-price.ts`
- Hooks: `use-kebab-case.ts` → `use-cart.ts`
- Types: `kebab-case.types.ts` → `product.types.ts`

## Export Pattern

typescript

//  *Named exports cho utilities*

```
export const formatPrice = () => {};
```

```
export const calculateTotal = () => {};
```

//  *Default export cho components*

```
export default function ProductCard() {}
```

//  *Barrel exports trong index.ts*

```
export { ProductCard } from './ProductCard';
```

```
export { ProductGrid } from './ProductGrid';
```

---

## Next Steps

Bạn muốn mình:

1. Tạo script migration tự động?
2. Tạo detailed checklist cho từng file?
3. Giải thích chi tiết về một phần cụ thể?
4. Tạo tsconfig paths mapping?