src/

│

├── api/ # API services

│ ├── patientApi.js

│ ├── appointmentApi.js

│ ├── prescriptionApi.js

│ └── authApi.js

│

├── assets/ # Static files

│ ├── images/

│ ├── fonts/

│ └── styles/

│ ├── antd.less # Ant Design theme override

│ └── global.less # Global styles

│

├── components/ # Reusable components

│ ├── common/ # Common UI components

│ │ ├── CustomTable/

│ │ ├── LoadingOverlay/

│ │ └── ErrorBoundary/

│ │

│ └── features/ # Feature-specific components

│ ├── Patient/

│ │ ├── PatientForm/

│ │ └── PatientInfoCard/

│ │

│ └── Appointment/

│ ├── CalendarView/

│ └── TimeSlotPicker/

│

├── config/ # App configuration

│ ├── routes.js

│ └── constants.js

│

├── contexts/ # React Contexts

│ ├── AuthContext.js

│ └── ClinicContext.js

│

├── hooks/ # Custom hooks

│ ├── useForm.js

│ └── usePagination.js

│

├── layouts/ # Page layouts

│ ├── MainLayout/ # Layout chính (có sidebar)

│ └── AuthLayout/ # Layout cho auth pages

│

├── pages/ # Page components

│ ├── Dashboard/

│ ├── Patients/

│ │ ├── PatientList/

│ │ └── PatientDetail/

│ ├── Appointments/

│ │ ├── AppointmentCalendar/

│ │ └── CreateAppointment/

│ ├── Prescriptions/

│ └── Settings/

│

├── store/ # State management (Redux/Zustand)

│ ├── slices/

│ │ ├── patientSlice.js

│ │ └── appointmentSlice.js

│ └── store.js

│

├── utils/ # Utilities & helpers

│ ├── validators/ # Validation schemas

│ ├── formatters/ # Data formatting

│ └── auth.js

│

├── App.jsx

└── main.jsx

**Giải thích chi tiết từng phần**

**1. Thư mục api**

* **Chức năng**: Xử lý tất cả logic gọi API
* **Ví dụ**:

javascript

Copy

// patientApi.js

export const fetchPatients = async (params) => {

return axios.get('/api/patients', { params });

};

**2. Thư mục assets**

* **Chức năng**: Chứa tài nguyên tĩnh và file style
* **Ant Design Customization**:

less

Copy

/\* antd.less \*/

@primary-color: #1890ff; // Override màu chủ đạo

@border-radius-base: 4px;

**3. Thư mục components**

* **Tổ chức**:
  + **common**: Components dùng chung toàn app
  + **features**: Components đặc thù theo nghiệp vụ
* **Ví dụ Ant Design Integration**:

jsx

Copy

// CustomTable.jsx

import { Table, Button } from 'antd';

const CustomTable = ({ columns, data }) => {

return (

<Table

columns={columns}

dataSource={data}

pagination={{ pageSize: 10 }}

/>

);

};

**4. Thư mục pages**

* **Cấu trúc điển hình**:

jsx

Copy

// pages/Patients/PatientList.jsx

import { PageHeader, Space } from 'antd';

import CustomTable from '@components/common/CustomTable';

const PatientList = () => {

return (

<PageHeader title="Danh sách bệnh nhân">

<Space direction="vertical">

<SearchBar />

<CustomTable columns={patientColumns} data={patients} />

</Space>

</PageHeader>

);

};

**5. State Management với Ant Design**

* **Kết hợp Form**:

jsx

Copy

// components/features/Patient/PatientForm.jsx

import { Form, Input, DatePicker } from 'antd';

const PatientForm = ({ initialValues }) => {

const [form] = Form.useForm();

return (

<Form form={form} layout="vertical">

<Form.Item label="Họ tên" name="name" rules={[...]}>

<Input />

</Form.Item>

<Form.Item label="Ngày sinh" name="dob">

<DatePicker />

</Form.Item>

</Form>

);

};

**6. Layout với Ant Design**

* **Main Layout**:

jsx

Copy

// layouts/MainLayout.jsx

import { Layout, Menu } from 'antd';

const { Header, Sider, Content } = Layout;

export default ({ children }) => {

return (

<Layout>

<Sider theme="light">

<Menu items={[...]} />

</Sider>

<Layout>

<Header style={{ background: '#fff' }}>...</Header>

<Content style={{ padding: 24 }}>{children}</Content>

</Layout>

</Layout>

);

};

**Best Practices với Ant Design**

1. **Theme Customization**
   * Override biến LESS trong assets/styles/antd.less
   * Import file style trong main.jsx:

javascript

Copy

import './assets/styles/antd.less';

1. **Responsive Design**

jsx

Copy

import { Grid } from 'antd';

const { useBreakpoint } = Grid;

const Component = () => {

const screens = useBreakpoint();

return (

<div>{screens.md ? 'Desktop View' : 'Mobile View'}</div>

);

};

1. **Form Handling**
   * Sử dụng Form kết hợp Form.Item và rules validation
   * Tích hợp với yup/zod cho validation phức tạp
2. **Global Components**
   * Tạo higher-order components cho các chức năng chung:

jsx

Copy

const withAuth = (WrappedComponent) => {

return (props) => {

const { user } = useAuth();

return user ? <WrappedComponent {...props} /> : <Navigate to="/login" />;

};

};

**Cài đặt cần thiết**

bash

Copy

npm install antd @ant-design/icons axios react-router-dom redux @reduxjs/toolkit react-redux

Cấu trúc này giúp:

* Dễ dàng mở rộng tính năng mới
* Tái sử dụng components
* Quản lý state tập trung
* Custom theme Ant Design linh hoạt
* Tách biệt rõ ràng các tầng logic