frontend/

├── public/ # 静态资源（不经过打包，直接复制到 dist）

│ └── favicon.ico

│

├── src/ # 核心代码

│ ├── assets/ # 图片、字体、全局样式等静态资源

│ │ ├── images/

│ │ └── styles/

│ │ └── global.css

│

│ ├── components/ # 可复用的通用组件

│ │ ├── Navbar.jsx

│ │ ├── Sidebar.jsx

│ │ └── Button.jsx

│

│ ├── pages/ # 页面级组件（按路由划分）

│ │ ├── Dashboard/

│ │ │ └── index.jsx

│ │ ├── Users/

│ │ │ └── index.jsx

│ │ ├── Roles/

│ │ │ └── index.jsx

│ │ └── Login.jsx

│

│ ├── layouts/ # 页面布局组件（如带导航栏、侧边栏的主框架）

│ │ └── MainLayout.jsx

│

│ ├── routes/ # 路由配置

│ │ └── index.jsx

│

│ ├── services/ # 与后端 API 交互

│ │ ├── api.js # axios 封装

│ │ ├── userService.js

│ │ └── roleService.js

│

│ ├── store/ # 全局状态管理 (Redux / Zustand / Context)

│ │ └── userStore.js

│

│ ├── utils/ # 工具函数

│ │ └── formatDate.js

│

│ ├── App.jsx # 根组件

│ └── main.jsx # 入口文件

│

├── .env # 环境变量（API 地址等）

├── package.json

└── vite.config.js

 **components** → 通用小组件（按钮、弹窗、表格…）

 **pages** → 页面，按功能模块组织（用户管理、角色管理、登录…）

 **layouts** → 页面布局（带导航栏/侧边栏的主框架）

 **services** → 统一管理接口请求（方便换后端地址、加 token 等）

 **store** → 全局状态管理（比如保存当前登录用户信息）

 **utils** → 公用工具函数（时间格式化、权限判断等）