**React.js Full Dashboard Project Guide: AutoZone**

### 🏠 Project Overview

**Title:** AutoZone - Vehicle Service Management Dashboard

**Description:** A role-based dashboard system similar to a luxury car brand (e.g., Benz), which includes:

* **User Dashboard**: View cars, spare parts, and book services
* **Admin Dashboard**: Manage users, cars, parts, and services
* **Company Dashboard**: Add/manage their own cars, parts, and services

### 🛠️ Tech Stack

**Frontend:**

* React.js + Vite
* TypeScript
* Tailwind CSS / MUI
* React Router
* Axios
* Zustand / Redux Toolkit
* React Hook Form + Yup

**Backend (Open Source or Mock APIs):**

* [mockapi.io](https://mockapi.io/)
* [JSON Server](https://github.com/typicode/json-server)
* Node.js Express APIs (optional)

### 📊 Dashboard Modules

#### User Dashboard

* Browse cars, spare parts, services
* Search, filter, cart,and sort listings
* Book services and track status

#### Admin Dashboard

* View analytics: users, companies, bookings
* Approve/reject company submissions
* CRUD operations on all entities

#### Company Dashboard

* Add/edit/delete car listings
* Manage spare parts and services
* Track booking requests

### 📈 Features Table

| Feature | Role | Description |
| --- | --- | --- |
| Cars Listing | All | View cars with details |
| Service Booking | User | Form-based service booking |
| Admin Controls | Admin | Approve, reject, manage data |
| Add Spare Parts | Company | Input form with validation |
| Role-Based Access | All | Render dashboards based on role |
| Responsive UI | All | Mobile-friendly with Tailwind |

### 🔗 Sample db.json (JSON Server)

{  
 "cars": [  
 {  
 "id": 1,  
 "brand": "Mercedes-Benz",  
 "model": "E-Class",  
 "price": 7000000,  
 "image": "/cars/eclass.jpg",  
 "companyId": 101  
 }  
 ],  
 "parts": [  
 {  
 "id": 1,  
 "name": "Brake Pads",  
 "price": 12000,  
 "carModel": "E-Class",  
 "companyId": 101  
 }  
 ],  
 "services": [  
 {  
 "id": 1,  
 "name": "Full Engine Service",  
 "price": 25000,  
 "companyId": 101  
 }  
 ],  
 "users": [  
 {  
 "id": 1,  
 "name": "Mamatha",  
 "role": "user",  
 "email": "mamatha@gmail.com"  
 },  
 {  
 "id": 2,  
 "name": "Admin",  
 "role": "admin",  
 "email": "admin@autozone.com"  
 },  
 {  
 "id": 3,  
 "name": "Benz Motors",  
 "role": "company",  
 "email": "benz@company.com"  
 }  
 ],  
 "bookings": []  
}

Start backend with:

npm install -g json-server  
json-server --watch db.json --port 3001

### 🚀 React Project Starter (with Tailwind, Vite, TypeScript)

#### 1. Create Project

npm create vite@latest autozone-dashboard -- --template react-ts  
cd autozone-dashboard  
npm install  
npm install axios react-router-dom zustand clsx  
npm install -D tailwindcss postcss autoprefixer  
npx tailwindcss init -p

#### 2. Tailwind Config (tailwind.config.js)

content: ["./index.html", "./src/\*\*/\*.{js,ts,jsx,tsx}"]

#### 3. Folder Structure

autozone-dashboard/  
├── public/  
├── src/  
│ ├── assets/  
│ ├── components/  
│ │ ├── common/  
│ │ └── dashboard/  
│ ├── pages/  
│ │ ├── auth/  
│ │ ├── admin/  
│ │ ├── user/  
│ │ └── company/  
│ ├── routes/  
│ ├── services/  
│ ├── store/  
│ ├── hooks/  
│ ├── utils/  
│ ├── App.tsx  
│ └── main.tsx

#### 4. Basic Routing Setup

**App.tsx**

import { BrowserRouter as Router, Routes, Route } from 'react-router-dom'  
import Login from './pages/auth/Login'  
import UserDashboard from './pages/user/UserDashboard'  
import AdminDashboard from './pages/admin/AdminDashboard'  
import CompanyDashboard from './pages/company/CompanyDashboard'  
  
function App() {  
 return (  
 <Router>  
 <Routes>  
 <Route path="/" element={<Login />} />  
 <Route path="/user" element={<UserDashboard />} />  
 <Route path="/admin" element={<AdminDashboard />} />  
 <Route path="/company" element={<CompanyDashboard />} />  
 </Routes>  
 </Router>  
 )  
}  
  
export default App

### 🎓 Interview Topics Practiced

| Topic | Covered |
| --- | --- |
| React Routing | ✅ |
| State Management | ✅ Zustand/Redux |
| Axios Integration | ✅ |
| Forms & Validation | ✅ React Hook Form + Yup |
| Conditional UI | ✅ Role-based rendering |
| Code Structure | ✅ Modular, scalable |
| Responsive Design | ✅ |
| Authentication Logic | ✅ Mock roles, access control |

Let me know if you’d like:

* Login logic with mock auth
* Dashboard layout with sidebar/topbar
* Role-protected routes
* CRUD form example

**🔗 Sample Open Source Backends**

1. [typicode/json-server](https://github.com/typicode/json-server) — Quick REST API with a db.json
2. [SaidMounaim/auto-service-api](https://github.com/SaidMounaim/auto-service-api) — Auto service backend
3. [devmeganet/vehicle-management-backend](https://github.com/devmeganet/vehicle-management-backend) — Vehicles, user auth, bookings
4. [open-vehicles/OVMS](https://github.com/openvehicles/Open-Vehicle-Monitoring-System) — Real-world telematics (advanced)