## 🏗️ High-Level Architecture

### Tenancy Strategy

* **Single Database, Shared Schema**
  + Tenants are distinguished by tenant\_id on shared tables.
  + Easy to scale, best for MVP/SaaS projects.

## 📋 Project Setup Plan

### ✅ Phase 0: Foundation

* Laravel app installed
* Inertia.js with Vue 3 via Breeze
* Authentication working

### ✅ Phase 1: Tenancy Core

#### 1. Tenant Identification

* Middleware: detect tenant by domain, subdomain, or session.
* tenants table: id, name, domain, plan, etc.

#### 2. Middleware + Service Provider

* IdentifyTenant middleware sets current tenant.
* Service Provider shares tenant context with Inertia views.

#### 3. Tenant-Aware Models

* Add tenant\_id to models.
* Use global scopes to auto-filter by tenant.

### ✅ Phase 2: Authentication + Roles

#### 4. User Roles/Permissions

* Use spatie/laravel-permission package.
* Define roles: Owner, Admin, Member per tenant.

#### 5. Team-based Access (Optional)

* Tenants as Teams.
* Users assigned to multiple tenants.

### ✅ Phase 3: Main Features (Modular)

#### Core Tenant Features

* 📁 Projects/Workspaces: CRUD, tenant-scoped.
* 👥 User Management: invite, assign roles.
* 📊 Dashboard: tenant-specific data.
* ⚙️ Settings: profile, API keys, tenant config.
* 🧾 Billing: subscription plans (Stripe).
* 🔔 Notifications: tenant-level.
* 📦 Custom Resources: tasks, posts, etc.

### ✅ Phase 4: Admin Portal

#### Super Admin Features

* Manage tenants.
* Subscription plans.
* Global analytics.
* Impersonate tenant users.

### ✅ Phase 5: Infrastructure & Optimization

* Environment-aware configs per tenant.
* Caching by tenant.
* Queue workers per tenant (if required).
* Telescope/Horizon for monitoring.

### ✅ Phase 6: Frontend (Vue 3)

* Responsive layouts per tenant.
* Sidebar, top nav, dark mode.
* Form validation via Inertia + FormRequests.
* Vue state with Pinia or Composition API.

## 🧱 Suggested Stack & Packages

| Tool / Package | Purpose |
| --- | --- |
| stancl/tenancy (optional) | Advanced tenancy support |
| spatie/laravel-permission | Roles & permissions |
| tightenco/ziggy | Route support for Inertia/Vue |
| laravel/sanctum | SPA auth / API token support |
| laravel/cashier | Stripe billing |

## 🔄 Dev Tips

* Middleware: IdentifyTenant, CheckRole, EnsureTenantAccess
* Extract reusable Inertia layouts
* Use Inertia shared props for tenant name, role, and settings
* Clean folder structure: Modules/, Domains/, Tenants/, etc.

Ready to start coding from here. Next step suggestion: IdentifyTenant middleware.