Laravel 12 + React Starter Kit Full Tutorial

1. Project Structure Overview

your-project/

- app/

  - Http/

    - Controllers/

      - AdminController.php

      - SuperAdminController.php

      - CategoryController.php

      - SubcategoryController.php

      - ProductController.php

      - ContactMessageController.php

      - PostController.php

    - Middleware/

      - RoleMiddleware.php

  - Models/

    - User.php

    - Post.php

    - Category.php

    - Subcategory.php

    - Product.php

    - ContactMessage.php

- database/

  - migrations/

  - seeders/

- resources/

  - js/

    - Pages/

    - Components/

    - Layouts/

  - views/

- routes/

  - web.php

  - api.php

- storage/

- artisan

- composer.json

- package.json

- vite.config.js

- .env

2. Setting up Laravel 12 with Inertia.js and React

- Install Laravel

- Install Inertia.js server-side adapter

- Install React front-end

- Set up basic routes and Inertia pages

3. Authentication Setup

- Install Laravel Breeze or Laravel Jetstream (Inertia version)

- Modify registration/login logic

- Create roles: Super Admin, Admin

4. Middleware for Role Management

- Create RoleMiddleware.php

- Protect routes using role-based authorization

5. CRUD for Admin Management

- Super Admin can create Admins

- Admins cannot create other admins

- AdminController.php

6. Category and Subcategory CRUD

- CategoryController.php
- SubcategoryController.php
- Create, Read, Update, Delete
- Pagination, Search, Sorting

7. Product CRUD with Image Upload

- ProductController.php
- Upload product images
- Validation
- Search products by category, subcategory

8. Contact Us Form

- ContactMessageController.php
- Frontend ContactForm.vue
- Backend stores messages

9. Post CRUD (Optional Blogging System)

- Create PostController.php
- Allow Admins to manage blog posts

10. Pagination, Search, Sorting Frontend (React)

- Pagination.vue component
- SearchBar.vue component
- SortButtons.vue component

11. File Upload Handling

- Handle file uploads via FormData

- Store in /storage/app/public/posts/

- Create symbolic link: php artisan storage:link

12. Final Touches

- Dashboard.vue for Admin

- Navbar.vue and Sidebar.vue

- Responsive layout with TailwindCSS

13. Deployment

- Build assets: npm run build

- Set environment: APP_ENV=production

- Deploy to server (optional SSL, optional Horizon for queues)