

Software Requirements Specification (SRS)

Business Website Platform (Laravel 12 + Vue 3 + Vuetify 3)
Version 1.0

1. INTRODUCTION

1.1 Purpose

This SRS outlines the complete technical and functional specifications of a modular business website platform designed to support dynamic content management, product catalogs, service catalogs, RBAC-based admin panel, and an SEO-friendly public website.

1.2 System Overview

The system features:

- Dynamic CMS pages
- Product & service catalog with multi-tab structure
- Full admin panel built with Vue 3 + Vuetify
- Laravel 12 REST API ecosystem
- Visitor & Login analytics
- Deployment-ready architecture (subdirectory/shared hosting support)

1.3 Intended Users

- System Administrators
- Content Managers
- Developers
- End Users (website visitors)

1.4 Technology Stack

Backend: Laravel 12, Sanctum, PHP 8.2+

Frontend: Vue 3 (SPA), Vuetify 3

Database: MySQL / PostgreSQL

Build Tools: Vite

Plugins: SweetAlert2, Axios, Progress Bar

2. SYSTEM ARCHITECTURE

2.1 High-Level Architecture

The system follows a decoupled architecture:

Frontend (Vue SPA) \leftrightarrow REST API \leftrightarrow Laravel Backend \leftrightarrow Database

2.2 Modular Architecture

Modules include:

- CMS
- Products
- Services
- Leads
- RBAC
- Logs
- Settings
- Optional: Portfolio, FAQ, Careers, Branches

2.3 Deployment Architecture

Supports:

- Local dev mode

- Shared hosting
- Subdirectory deployment
- Cloud or VPS

3. FUNCTIONAL REQUIREMENTS

3.1 Admin Panel

Includes authenticated access with RBAC.

3.1.1 Authentication

- Login
- Logout
- Fetch logged-in user
- Sanctum token-based auth

3.1.2 Product Module – 11-Tab Detailed Specification

1. Basic Info

- Title (required)
- Slug (unique)
- SKU, short & long descriptions

2. Media

- Thumbnail
- Gallery images

3. Pricing

- price
- price_range
- show_price (boolean)

4. Categories & Tags

- category_ids (pivot)
- tag_names (auto-created)

5. Specifications

- Dynamic key-value JSON list

6. Features

- Key technical or marketing bullet points

7. Downloads

- Datasheets, manuals, PDFs

8. FAQs

- Question-answer structured data

9. Warranty & Service

- Coverage duration
- Terms & conditions

10. SEO

- Meta title

- Meta description
- Meta keywords
- OG Image

11. Settings

- published
- featured
- stock
- order prioritization

3.1.3 Services Module

CRUD with images, content, SEO metadata.

3.1.4 CMS Page Module

Dynamic content pages with meta tags.

3.1.5 RBAC (Roles & Permissions)

- Create roles
- Assign permissions
- Enforce access-control
- Users CRUD

3.1.6 Leads Module

Stores public inquiries; CSV export.

3.1.7 Logging

- Visitor logs
- Login logs

3.2 Public Website Requirements

3.2.1 Dynamic Homepage

3.2.2 Product Listing with filters, search, sorting

3.2.3 Product Details with gallery, specs, FAQs, warranty

3.2.4 Service Pages

3.2.5 Contact Form (public → leads module)

4. API REQUIREMENTS

4.1 Admin API (/api/v1/)

- Full CRUD for: pages, products, categories, tags, leads, users, roles, settings
- File upload endpoints
- Logs API
- Authentication endpoints

4.2 Public API (/api/openapi/)

- Homepage
- Pages
- Services
- Products
- Categories
- Settings
- Contact form submission

5. NON-FUNCTIONAL REQUIREMENTS

5.1 Performance

- Vite-optimized assets
- Query optimization
- Cached config/routes/views

5.2 Security

- Sanctum-based auth
- CSRF protection
- Role-based access control
- .htaccess protected sensitive paths

5.3 Usability

- Intuitive admin UI (Vuetify)
- Clean layout
- Responsive public interface

5.4 Scalability

- Modular structure
- Specification JSON extensibility

5.5 Reliability

- Graceful error handling
- Log tracking

6. DEPLOYMENT REQUIREMENTS

6.1 Build Commands

```
npm run build
```

```
php artisan config:cache, route:cache, view:cache
```

6.2 Shared Hosting Requirements

- Correct APP_URL, ASSET_URL
- Move public/ files if needed
- Permissions for storage, cache

6.3 Subdirectory Deployment

- Works under /folder/subfolder/
- htaccess rewrite rules
- Asset path correctness via ASSET_URL

6.4 Production Preparation

- Optimize Laravel
- Disable APP_DEBUG

7. TROUBLESHOOTING

- Forbidden errors – htaccess misconfiguration
- Vite 404 – ASSET_URL incorrect

- Routes 404 – route cache issues
- Storage link issues – symbolic link restrictions
- Database errors – invalid credentials or permission restrictions

8. SYSTEM CONSTRAINTS

- PHP 8.2+
- Node.js 18+
- Shared hosting limitations apply
- Database must support JSON

9. FUTURE SCOPE

- Full eCommerce system
- Multi-language support
- Advanced CMS blocks
- API versioning
- Rate-limiting and analytics dashboards