Business Website Platform

A comprehensive, generic business website platform built according to SRS specifications. This platform can represent almost any type of business with configurable modules.

🛠 Technical Specifications

Backend Stack

Core Framework

- **Laravel Framework**: ^12.0

- **PHP**: ^8.2

- **Composer**: Dependency management

Authentication & Security

- **Laravel Sanctum**: ^4.2 - API token-based authentication

- **Laravel UI**: ^4.6 - Authentication scaffolding

Database & ORM

- **Laravel Eloquent ORM**: Built-in ORM for database operations

- **MySQL/PostgreSQL/SQLite**: Supported database systems

- **Yajra Laravel DataTables**: ^12.3 - Advanced data tables with server-side processing

- **Yajra Laravel OCI8**: ^12.0 - Oracle database support

File Processing & Media

- **Intervention Image**: ^3.11 - Image manipulation library

- **Intervention Image Laravel**: ^1.5 - Laravel integration for image processing

- **Barryvdh Laravel Snappy**: ^1.0 - PDF generation from HTML

Data Export & Import

- **Maatwebsite Excel**: ^3.1 - Excel file import/export functionality

API & Integration

- **Guzzle HTTP**: ^7.9 - HTTP client for API requests

- **Inertia.js Laravel**: ^2.0 - Modern monolith approach (SPA-like experience)

- **Tightenco Ziggy**: ^2.4 - Route helper for JavaScript

Utilities

- **SimpleSoftwareIO Simple QRCode**: ^4.2 - QR code generation

- **Irazasyed Telegram Bot SDK**: ^3.15 - Telegram bot integration

- **DirectoryTree LDAPRecord Laravel**: ^3.4 - LDAP authentication support

Development Tools

- **Laravel Tinker**: ^2.10.1 - REPL for Laravel

- **Laravel Pint**: ^1.18 - Code style fixer

- **Laravel Sail**: ^1.41 - Docker development environment

- **Laravel Pail**: ^1.2.2 - Real-time log viewer

- **OpcodesIO Log Viewer**: ^3.17 - Advanced log viewing interface

Testing

- **PHPUnit**: ^11.5.3 - Unit testing framework

- **Mockery**: ^1.6 - Mocking library for testing

- **Nunomaduro Collision**: ^8.6 - Error handler for console/CLI

- **FakerPHP Faker**: ^1.23 - Fake data generator for testing

Frontend Stack

Core Framework

- **Vue.js**: ^3.2.37 - Progressive JavaScript framework

- **Vue Router**: ^4.3.0 - Official router for Vue.js

- **Vite**: ^6.3.5 - Next-generation frontend build tool

UI Framework

- **Vuetify**: ^3.5.14 - Material Design component framework

- **Material Design Icons**: ^7.4.47 - Icon library (@mdi/font)

State Management

- **Pinia**: ^3.0.4 - Official state management for Vue 3

- **Vuex**: ^4.1.0 - State management pattern library

Forms & Validation

- **VForm**: ^2.1.2 - Form validation library for Vue.js

Rich Text Editing

- **Quill**: ^2.0.3 - Rich text WYSIWYG editor

- **Vue3 Editor**: ^0.1.1 - Vue 3 wrapper for Quill

Data Visualization

- **Chart.js**: ^4.5.1 - Simple yet flexible JavaScript charting library

- **Vue ChartJS**: ^5.3.3 - Vue.js wrapper for Chart.js

UI Enhancements

- **Vue SweetAlert2**: ^5.0.5 - Beautiful, responsive, customizable popup boxes

- **Vue3 Progress Bar**: ^1.0.3 - Progress bar component for Vue 3

Utilities

- **Axios**: ^1.6.4 - Promise-based HTTP client

- **Moment.js**: ^2.30.1 - Date manipulation library

- **Bootstrap**: ^5.3.6 - CSS framework (for some legacy components)

- **Popper.js**: ^2.11.6 - Tooltip and popover positioning engine

- **Fontsource Prompt**: ^5.0.13 - Self-hosted Prompt font

Build Tools

- **Vite**: ^6.3.5 - Fast build tool and dev server

- **Sass**: ^1.89.1 - CSS preprocessor

- **Laravel Vite Plugin**: ^1.0 - Laravel integration for Vite

- **@vitejs/plugin-vue**: ^5.2.4 - Vue.js plugin for Vite

Development Environment

Required Software

- **PHP**: 8.2 or higher

- **Composer**: PHP dependency manager

- **Node.js**: 18+ (for npm)

- **npm**: Package manager for JavaScript

- **MySQL/PostgreSQL/SQLite**: Database system

Build Commands

```bash

## Install PHP dependencies

composer install

## Install JavaScript dependencies

npm install

## Development build (with hot reload)

npm run dev

## Production build

npm run build

```

Architecture

Backend Architecture

- **MVC Pattern**: Model-View-Controller architecture

- **RESTful API**: RESTful API design for admin panel

- **Service Layer**: Business logic separation

- **Repository Pattern**: Data access abstraction (where applicable)

Frontend Architecture

- **Component-Based**: Vue.js component architecture

- **SPA (Single Page Application)**: Client-side routing with Vue Router

- **API-Driven**: Frontend communicates with backend via REST API

- **State Management**: Centralized state with Pinia/Vuex

Security Features

- **CSRF Protection**: Laravel's built-in CSRF protection

- **XSS Protection**: Automatic output escaping

- **SQL Injection Protection**: Eloquent ORM parameter binding

- **Authentication**: Laravel Sanctum token-based authentication

- **Authorization**: Role-Based Access Control (RBAC) with permissions

Performance Optimizations

Backend

- **Query Optimization**: Eloquent eager loading

- **Caching**: Laravel's caching system

- **Database Indexing**: Optimized database queries

- **File Compression**: Image optimization with Intervention Image

Frontend

- **Code Splitting**: Lazy loading of routes and components

- **Asset Optimization**: Vite's built-in optimization

- **Image Lazy Loading**: Lazy loading for images

- **Tree Shaking**: Unused code elimination

Browser Support

- **Modern Browsers**: Chrome, Firefox, Safari, Edge (latest versions)

- **Mobile Browsers**: iOS Safari, Chrome Mobile

- **Responsive Design**: Mobile-first approach with Vuetify

🚀 Features

Core Features

Authentication & Security

- **Laravel Sanctum Authentication**: Secure API token-based authentication for admin panel

- **Role-Based Access Control (RBAC)**: Granular permission system with roles and permissions

- **User Management**: Complete user administration with role assignment

- **Permission Management**: Fine-grained permission control for all features

- **Login Logging**: Track all login attempts (successful and failed) with IP addresses and user agents

- **Visitor Logging**: Comprehensive visitor tracking with device, browser, OS, and bot detection

Settings Management

- **General Settings**: Site name, tagline, contact information, address

- **Home Page Settings**: Modular section management with tabbed interface:

  - Hero section with customizable title, subtitle, and CTA buttons

  - Statistics section with up to 4 customizable stat cards

  - Trusted By section with dynamic client logo management

  - Services section (WHAT WE DO) with dynamic service cards

  - Why Choose Us section with features management

  - Testimonials section with testimonial cards

  - Featured Products section

  - CTA (Call-to-Action) section with primary and secondary buttons

  - Section visibility toggles for each section

- **Contact Page Settings**: Contact form configuration and page content

- **Branding Settings**: Logo, favicon, and color scheme customization

- **Social Media Settings**: Social media links and integration

- **SEO Settings**: Default meta tags, descriptions, and keywords

- **Email/SMTP Settings**: Email configuration for notifications

Leads & Communication

- **Lead Management**: Complete lead tracking system with:

  - Lead status management (New, Contacted, Qualified, Converted, Lost)

  - Lead type categorization (Contact, Quote, Support)

  - Read/Unread status tracking

  - Lead assignment to users

  - Notes and comments

  - CSV export functionality

- **Newsletter Subscriptions**: Newsletter management with:

  - Subscription status tracking

  - Email verification

  - CSV export functionality

  - Bulk operations

About Page Management

- **Singleton About Page**: Single about page management with:

  - Company story and description

  - Mission and vision

  - Team information

  - Company values

Admin Panel Features

Dashboard & Analytics

- **AI-Powered Analytics Dashboard**: Comprehensive dashboard with:

  - **Statistics Cards**:

    - Total Visitors with trend indicators

    - Human Visits with percentage breakdown

    - New Leads with attention alerts

    - Total Products with services count

  - **Interactive Charts** (Chart.js):

    - Visitor Trends: Line chart showing total and human visits over time

    - Device Distribution: Doughnut chart for desktop, mobile, tablet breakdown

    - Browser Distribution: Bar chart showing top browsers

    - Login Activity: Pie chart for successful vs failed logins

    - Leads by Status: Bar chart showing lead status distribution

    - Top Visited Pages: Horizontal bar chart for most visited URLs

  - **Time Range Selection**: Filter data by 7 days, 30 days, 90 days, or 1 year

  - **AI-Powered Insights**: Automated analysis providing:

    - Warnings for high bot traffic, login failures, new leads requiring attention

    - Success messages for strong engagement
    - Information about content portfolio
  - **Recent Activity**: Latest leads table with unread indicators
  - **Real-time Data**: All statistics update dynamically from database

Services Management
- **Service CRUD**: Complete service management with:
  - Title, slug, and descriptions
  - Icon selection (Material Design Icons)
  - Image upload with preview
  - Price range display
  - Features and benefits management
  - SEO optimization
  - Published/Draft status
  - Display order management
- **Service Details View**: Read-only view of complete service information
- **Rich Text Editor**: WYSIWYG editor for service descriptions

Products Management
- **Comprehensive Product Management**: 11-tab form system:
  1. **Basic Info**: Title, slug, SKU, short description, full description
  2. **Media**: Thumbnail and gallery images with preview and upload
  3. **Pricing**: Price, price range, show/hide price toggle
  4. **Categories & Tags**: Multi-select with auto-creation support
  5. **Specifications**: Dynamic key-value pairs for technical specs
  6. **Features**: Key features list with descriptions

  7. **Downloads**: File uploads (PDFs, datasheets, manuals) with preview

  8. **FAQs**: Question-answer pairs for customer support

  9. **Warranty & Service**: Warranty information and service details

  10. **SEO**: Meta title, description, keywords, OG image

  11. **Settings**: Published status, featured flag, stock management, display order

- **Product Details View**: Complete read-only view of all product information

- **Image Management**:

  - Thumbnail upload with preview

  - Gallery images with drag-and-drop reordering

  - Image deletion

  - Automatic path normalization

- **File Management**:

  - Downloadable files (PDFs, ZIPs, etc.)

  - File type detection

  - File size display

  - Preview before upload

Categories Management

- **Hierarchical Categories**:

  - Parent-child relationships

  - Category type filtering (product, service, etc.)

  - Category icons and images

  - Published/Draft status

  - Display order

  - SEO optimization

Tags Management

- **Tag System**:
  - Tag creation and management
  - Tag type filtering
  - Slug auto-generation
  - Tag usage tracking

Blog Management

- **Blog Post Management**:
  - Complete CRUD operations for blog posts
  - Rich text editor for content
  - Featured image upload with preview
  - Excerpt management
  - Author assignment (automatic from authenticated user)
  - Category assignment (multiple categories per post)
  - Tag assignment (multiple tags per post)
  - Publication date scheduling
  - Draft/Published status
  - SEO optimization (meta title, description, keywords, OG image)
  - View counter tracking
  - Search and filter functionality
  - Sorting by title, date, views, published status
  - Pagination support
- **Blog Category Management**:
  - Category CRUD operations
  - Category type filtering (post type)

  - Hierarchical category support

  - Category description and images

  - Published/Draft status

  - Display order management

  - SEO optimization

Career Management

- **Career Posting Management**:

  - Complete CRUD operations for career listings

  - Job title, slug, and department management

  - Location and employment type (full-time, part-time, contract, internship, freelance)

  - Rich text editor for job description, responsibilities, requirements, and benefits

  - Application deadline management with automatic expiration checking

  - Published/Draft status control

  - Display order management

  - Search and filter functionality (by department, location, employment type, published status)

  - Sorting by title, department, location, order, deadline, created date

  - Pagination support

  - Application count tracking per career

- **Job Application Management**:

  - Complete application tracking system

  - Application status management (new, reviewed, shortlisted, rejected, hired)

  - Applicant information (name, email, phone with Bangladesh validation)

  - Cover letter management

  - Resume file upload and download (PDF, DOC, DOCX)

  - Application notes and comments

  - Search and filter functionality (by status, career, applicant name/email)

  - Sorting by name, email, status, application date

  - Application details view with notes editing

  - Statistics dashboard (total, by status, by career)

  - Resume file management with automatic cleanup on deletion

User Management

- **User Administration**:

  - User creation and editing

  - Role assignment

  - Avatar upload

  - Account status management

  - Password reset functionality

Role & Permission Management

- **Role Management**:

  - Role creation with permissions

  - Permission assignment

  - System role protection

  - Permission grouping

- **Permission Management**:

  - Permission creation

  - Slug auto-generation

  - Permission grouping by feature

  - Flat and grouped view modes

Logs Management

- **Login Logs**:

  - Track all login attempts

  - Filter by status (success/failed)

  - Search by email, IP, user agent

  - Statistics dashboard

  - Time-series data for trends

- **Visitor Logs**:

  - Comprehensive visitor tracking

  - Device type detection (Desktop, Mobile, Tablet)

  - Browser and OS detection

  - Bot detection and filtering

  - IP address tracking

  - URL tracking

  - Referer tracking

  - Statistics with device/browser/OS breakdown

  - Top visited pages

  - Time-series data for trends

  - Bulk delete functionality

Public Website Features

Homepage

- **Dynamic Homepage**: Fully configurable homepage with:

  - Hero section with customizable content

  - Statistics section with animated counters
  - Trusted By section with client logos
  - Services showcase section
  - Why Choose Us section with features
  - Testimonials carousel
  - Featured Products section
  - Call-to-Action section
  - All sections can be toggled on/off from admin

Products Display
- **Product Listing Page**:
  - Category-based filtering with icon buttons
  - Real-time search across titles, descriptions, SKU, and specifications
  - Multiple sorting options (newest, price, name, featured)
  - Product cards with images, quick specs, and prices
  - Featured product badges
  - **Product Comparison Tool**: Compare up to 3 products side-by-side with:
    - Price comparison
    - Key specifications comparison
    - Technical differences
    - Recommended use-cases
    - Quick access to product details
  - Responsive grid layout
  - Sticky filter bar on scroll

- **Product Detail Page**:

  - Hero section with product overview

  - Product gallery with:

    - Main image with zoom functionality

    - Thumbnail navigation

    - Image zoom dialog for detailed viewing

  - Key features section with icons

  - Quick specifications sidebar

  - Tabbed content:

    - **Overview**: Detailed description and benefits

    - **Technical Specs**: Complete specifications table

    - **Features**: Detailed feature list

    - **Downloads**: Datasheets, manuals, documentation (PDF, ZIP, etc.)

    - **FAQs**: Expandable FAQ section

    - **Warranty & Service**: Warranty coverage and support information

  - Related products section

  - Trust badges (Warranty, Delivery, Support)

Services Display

- **Services Listing**: Grid view of all published services

- **Service Detail Page**: Complete service information with features and benefits

Blog System

- **Blog Listing Page**:

  - Modern hero section with animated gradient effects

  - Real-time search across post titles, excerpts, and content

  - Category-based filtering with sidebar navigation

  - Multiple sorting options (Latest, Oldest, Most Views, Title A-Z)
  - Pagination support with configurable items per page
  - Responsive card-based layout
  - Loading states and empty state handling
  - Category badges and tag display
  - Author information and view counters
  - Featured image support
- **Blog Detail Page**:
  - Hero section with featured image overlay
  - Full post content with rich HTML rendering
  - Breadcrumb navigation
  - Author information card
  - Social sharing (Facebook, Twitter, LinkedIn, Copy link)
  - Tags and categories display
  - Automatic view counter increment
  - Related posts placeholder
  - Styled content typography (headings, lists, blockquotes, code blocks)
  - Responsive image display
- **Blog Features**:
  - SEO-friendly URLs (slug-based)
  - Meta tags and Open Graph support
  - Category management
  - Tag system
  - Author attribution
  - Publication date management
  - Draft/Published status

  - View tracking

Career System

- **Career Listing Page**:

  - Modern hero section with animated gradient effects

  - Real-time search across job titles, departments, locations, and descriptions

  - Advanced filtering:

    - Department filter (dynamically populated from available careers)

    - Location filter (dynamically populated from available careers)

    - Employment type filter (full-time, part-time, contract, internship, freelance)

  - Active careers only (automatically filters out expired deadlines)

  - Responsive card-based layout with hover effects

  - Career cards displaying:

    - Job title and department badge

    - Location and employment type

    - Application deadline (if set)

    - Truncated job description

    - Quick view button

  - Pagination support

  - Loading states and empty state handling

  - Clear filters functionality

- **Career Detail Page**:

  - Hero section with job title and key information chips

  - Comprehensive job information display:

    - Full job description with rich HTML rendering

    - Detailed responsibilities section

    - Job requirements section

    - Benefits and perks section

  - **Job Application Form** (sidebar):

    - Full name field (required)

    - Email field with validation (required)

    - Phone field with Bangladesh phone number validation (optional)

    - Cover letter textarea

    - Resume upload (PDF, DOC, DOCX, max 5MB)

    - Application deadline display and validation

    - Active/inactive status checking

    - Form validation with user-friendly error messages

    - Success/error notifications using SweetAlert2

    - Automatic form reset on successful submission

  - Back to careers navigation

  - Responsive layout (details on left, form on right)

- **Career Features**:

  - SEO-friendly URLs (slug-based)

  - Automatic deadline expiration checking

  - Bangladesh phone number validation (supports local: 01707080401 and international: +8801707080401 formats)

  - Resume file management with secure uploads

  - Application tracking and status management

  - Only published and active careers shown to public

  - Application count per career

Contact & Communication

- **Contact Form**:

  - Name, email, phone, subject, message fields

  - Automatic lead creation

  - Email notifications

- **Newsletter Subscription**: Footer subscription form with email verification

SEO & Performance

- **SEO Optimization**:

  - Meta tags per page

  - Open Graph tags

  - Structured data

  - Sitemap generation

- **Performance**:

  - Image optimization

  - Lazy loading

  - Responsive images

  - Fast page loads

📋 Prerequisites

- PHP 8.2 or higher

- Composer

- Node.js & npm

- MySQL/PostgreSQL/SQLite

🔧 Installation

1. Clone and Install Dependencies

```bash
git clone <repository-url>
cd s_h_micro_control
composer install
npm install
```

2. Environment Setup

```bash
cp .env.example .env
php artisan key:generate
```

Configure your `.env` file:

- Database connection
- `APP_URL`
- Mail settings (for notifications)

3. Database Setup

```bash
php artisan migrate
php artisan db:seed
```

This will create:

- All database tables

- Module definitions (disabled by default)

- Default admin user

4. Build Assets

```bash
npm run dev  # Development

or
npm run build  # Production
```

5. Start Development Server

```bash
php artisan serve

npm run dev
```

Visit:

- Public website: `http://localhost:8000`

- Admin panel: `http://localhost:8000/admin/login`

🔑 Default Admin Credentials

- **Email**: admin@example.com

- **Password**: password

⚠️ **Change these immediately in production!**

📚 API Documentation

Admin API (`/api/v1/`)

All admin endpoints require authentication via Bearer token.

Authentication:

- `POST /api/v1/auth/login` - Login

- `POST /api/v1/auth/logout` - Logout

- `GET /api/v1/auth/user` - Get current user

About Page Management:

- `GET /api/v1/about` - Get about page (requires `manage-pages` permission)

- `POST /api/v1/about` - Create/update about page (requires `manage-pages` permission)

- `PUT /api/v1/about` - Update about page (requires `manage-pages` permission)

Blog Management:

- `GET /api/v1/blog-posts` - List blog posts (with pagination, filtering, sorting, search) (requires `manage-pages` permission)

- `POST /api/v1/blog-posts` - Create blog post (requires `manage-pages` permission)

- `GET /api/v1/blog-posts/{id}` - Get blog post (by ID or slug) (requires `manage-pages` permission)

- `PUT /api/v1/blog-posts/{id}` - Update blog post (requires `manage-pages` permission)

- `DELETE /api/v1/blog-posts/{id}` - Delete blog post (requires `manage-pages` permission)

- `GET /api/v1/blog-categories` - List blog categories (requires `manage-pages` permission)

- `POST /api/v1/blog-categories` - Create blog category (requires `manage-pages` permission)

- `GET /api/v1/blog-categories/{id}` - Get blog category (requires `manage-pages` permission)

- `PUT /api/v1/blog-categories/{id}` - Update blog category (requires `manage-pages` permission)

- `DELETE /api/v1/blog-categories/{id}` - Delete blog category (requires `manage-pages` permission)

Career Management:

- `GET /api/v1/careers` - List careers (with pagination, filtering, sorting, search) (requires `manage-pages` permission)

- `POST /api/v1/careers` - Create career (requires `manage-pages` permission)

- `GET /api/v1/careers/{id}` - Get career (by ID or slug) (requires `manage-pages` permission)

- `PUT /api/v1/careers/{id}` - Update career (requires `manage-pages` permission)

- `DELETE /api/v1/careers/{id}` - Delete career (requires `manage-pages` permission)

Job Application Management:

- `GET /api/v1/job-applications` - List job applications (with pagination, filtering, sorting, search) (requires `view-leads` permission)

- `GET /api/v1/job-applications/{id}` - Get job application (requires `view-leads` permission)

- `PUT /api/v1/job-applications/{id}` - Update job application status/notes (requires `manage-leads` permission)

- `DELETE /api/v1/job-applications/{id}` - Delete job application (requires `manage-leads` permission)

- `GET /api/v1/job-applications/statistics` - Get application statistics (requires `view-leads` permission)

Services Management:

- `GET /api/v1/services` - List services (with pagination, filtering, sorting)

- `POST /api/v1/services` - Create service (requires `manage-services` permission)

- `GET /api/v1/services/{id}` - Get service (by ID or slug)

- `PUT /api/v1/services/{id}` - Update service (requires `manage-services` permission)

- `DELETE /api/v1/services/{id}` - Delete service (requires `manage-services` permission)

Product Management:

- `GET /api/v1/products` - List products (with pagination, filtering, sorting, search)

- `POST /api/v1/products` - Create product (requires `manage-products` permission)

- `GET /api/v1/products/{id}` - Get product (by ID or slug, includes all relationships)

- `PUT /api/v1/products/{id}` - Update product (requires `manage-products` permission)

- `DELETE /api/v1/products/{id}` - Delete product (requires `manage-products` permission)

Category Management:

- `GET /api/v1/categories` - List categories (supports filtering by type, parent_id, published)

- `POST /api/v1/categories` - Create category (requires `manage-products` permission)

- `GET /api/v1/categories/{id}` - Get category (by ID or slug)

- `PUT /api/v1/categories/{id}` - Update category (requires `manage-products` permission)

- `DELETE /api/v1/categories/{id}` - Delete category (requires `manage-products` permission)

Tag Management:

- `GET /api/v1/tags` - List tags (supports filtering by type, search)

- `POST /api/v1/tags` - Create tag (requires `manage-products` permission)

- `GET /api/v1/tags/{id}` - Get tag (by ID or slug)

- `PUT /api/v1/tags/{id}` - Update tag (requires `manage-products` permission)

- `DELETE /api/v1/tags/{id}` - Delete tag (requires `manage-products` permission)

File Upload:

- `POST /api/v1/upload/image` - Upload single image (requires authentication)

- `POST /api/v1/upload/images` - Upload multiple images (requires authentication)

- `POST /api/v1/upload/file` - Upload file (PDF, DOC, ZIP, etc.) (requires authentication)

- `DELETE /api/v1/upload/image` - Delete image (requires authentication)

Leads Management:

- `GET /api/v1/leads` - List leads (with pagination, filtering, sorting, search) (requires `view-leads` permission)

- `GET /api/v1/leads/statistics` - Get leads statistics with time-series data (requires `view-leads` permission)

- `GET /api/v1/leads/unread-count` - Get count of unread leads (requires `view-leads` permission)

- `GET /api/v1/leads/{id}` - Get lead details (requires `view-leads` permission)

- `POST /api/v1/leads/{id}/mark-as-read` - Mark lead as read (requires `view-leads` permission)

- `PUT /api/v1/leads/{id}` - Update lead (requires `manage-leads` permission)

- `DELETE /api/v1/leads/{id}` - Delete lead (requires `manage-leads` permission)

- `GET /api/v1/leads/export/csv` - Export leads to CSV (requires `export-leads` permission)

Newsletter Subscriptions:

- `GET /api/v1/newsletters` - List newsletter subscriptions (requires `view-leads` permission)

- `GET /api/v1/newsletters/{id}` - Get subscription details (requires `view-leads` permission)

- `PUT /api/v1/newsletters/{id}` - Update subscription status (requires `manage-leads` permission)

- `DELETE /api/v1/newsletters/{id}` - Delete subscription (requires `manage-leads` permission)

- `GET /api/v1/newsletters/export/csv` - Export subscriptions to CSV (requires `manage-leads` permission)

User Management:

- `GET /api/v1/users` - List users (requires `manage-users` permission)

- `POST /api/v1/users` - Create user (requires `manage-users` permission)

- `GET /api/v1/users/{id}` - Get user (requires `manage-users` permission)

- `PUT /api/v1/users/{id}` - Update user (requires `manage-users` permission)

- `DELETE /api/v1/users/{id}` - Delete user (requires `manage-users` permission)

Role & Permission Management:

- `GET /api/v1/roles` - List roles (requires `manage-roles` permission)

- `POST /api/v1/roles` - Create role (requires `manage-roles` permission)

- `PUT /api/v1/roles/{id}` - Update role (requires `manage-roles` permission)

- `PUT /api/v1/roles/{id}/permissions` - Sync role permissions (requires `manage-roles` permission)

- `DELETE /api/v1/roles/{id}` - Delete role (requires `manage-roles` permission)

- `GET /api/v1/permissions` - List permissions (requires `manage-roles` permission)

- `GET /api/v1/permissions/groups` - Get permission groups (requires `manage-roles` permission)

- `POST /api/v1/permissions` - Create permission (requires `manage-roles` permission)

- `PUT /api/v1/permissions/{id}` - Update permission (requires `manage-roles` permission)

- `DELETE /api/v1/permissions/{id}` - Delete permission (requires `manage-roles` permission)

Logs & Analytics:

- `GET /api/v1/login-logs` - List login logs (with pagination, filtering, sorting) (requires `view-login-logs` permission)

- `GET /api/v1/login-logs/statistics` - Get login statistics with time-series data (requires `view-login-logs` permission)

- `GET /api/v1/login-logs/{id}` - Get login log details (requires `view-login-logs` permission)

- `DELETE /api/v1/login-logs/{id}` - Delete login log (requires `view-login-logs` permission)

- `GET /api/v1/visitor-logs` - List visitor logs (with pagination, filtering, sorting, search) (requires `view-visitor-logs` permission)

- `GET /api/v1/visitor-logs/statistics` - Get visitor statistics with time-series data (requires `view-visitor-logs` permission)

- `GET /api/v1/visitor-logs/{id}` - Get visitor log details (requires `view-visitor-logs` permission)

- `DELETE /api/v1/visitor-logs/{id}` - Delete visitor log (requires `view-visitor-logs` permission)

- `POST /api/v1/visitor-logs/delete-multiple` - Delete multiple visitor logs (requires `view-visitor-logs` permission)

Settings:

- `GET /api/v1/settings` - Get all settings (requires authentication)

- `PUT /api/v1/settings` - Update settings (requires `manage-settings` permission)

Public API (`/api/openapi/`)

- `GET /api/openapi/home` - Homepage data (includes home page settings and featured content)

- `GET /api/openapi/pages/{slug}` - Get page by slug (for public page viewing)

- `GET /api/openapi/services` - List published services

- `GET /api/openapi/services/{slug}` - Get service by slug

- `GET /api/openapi/products` - List published products (supports category filter, search, sorting)

- `GET /api/openapi/products/{slug}` - Get product by slug (includes categories, tags, specifications, downloads)

- `GET /api/openapi/categories` - List categories (supports type filter, pagination)

- `GET /api/openapi/blog` - List published blog posts (supports search, category filter, tag filter, sorting, pagination)

- `GET /api/openapi/blog/{slug}` - Get blog post by slug (includes author, categories, tags, auto-increments views)

- `GET /api/openapi/blog/categories` - List published blog categories

Career (Public):

- `GET /api/openapi/careers` - List published and active careers (supports search, department, location, employment type filters, active_only flag, sorting, pagination, returns filter options)

- `GET /api/openapi/careers/{slug}` - Get career by slug (includes application count, active status check)

- `POST /api/openapi/careers/apply` - Submit job application (with resume upload, Bangladesh phone validation, deadline checking)

- `GET /api/openapi/about` - Get about page content

- `GET /api/openapi/settings` - Get public settings

- `POST /api/openapi/contact` - Submit contact form (creates lead)

- `POST /api/openapi/newsletter/subscribe` - Subscribe to newsletter

🎨 Module Configuration

Modules are stored in the `modules` table. To enable a module:

Via Code:

```php
Module::where('name', 'services')->update(['enabled' => true]);
```

Via Database:

```sql
UPDATE modules SET enabled = 1 WHERE name = 'services';
```

Available Modules:

- `services` - Services catalog

- `products` - Products catalog

- `portfolio` - Portfolio/Projects

- `blog` - Blog/News

- `faq` - FAQ

- `careers` - Careers & Recruitment

- `booking` - Appointment booking

- `events` - Events & Registrations

- `branches` - Multi-location/Branches

- `ecommerce` - E-commerce (future)

- `multilanguage` - Multi-language support (future)

📁 Project Structure

```
app/
├── Http/
│   ├── Controllers/
│   │   ├── Api/           # Admin API controllers
│   │   │   ├── auth/
│   │   │   ├── about/
│   │   │   ├── blog/
│   │   │   ├── leads/
│   │   │   ├── NewsletterController.php
│   │   │   ├── logs/
│   │   │   ├── products/
│   │   │   ├── settings/
│   │   │   ├── upload/
│   │   │   └── users/
│   │   └── Public/         # Public website controllers
│   │       ├── pages/
│   │       ├── blog/
│   │       ├── NewsletterController.php
│   │       ├── products/
│   │       └── services/
│   └── Middleware/         # Authentication & authorization
├── Models/                 # Eloquent models

database/
├── migrations/             # Database migrations
└── seeders/                # Database seeders

resources/
├── js/
│   ├── components/
│   │   ├── admin/          # Admin panel components
│   │   │   ├── auth/
│   │   │   ├── about/
│   │   │   ├── blog/
│   │   │   ├── leads/
│   │   │   ├── newsletters/
│   │   │   ├── logs/
│   │   │   ├── products/
│   │   │   ├── settings/
│   │   │   └── users/
│   │   └── public/         # Public website components
│   │       ├── pages/
│   │       ├── blog/
│   │       ├── products/
│   │       └── services/
│   ├── mixins/          # Vue mixins
│   ├── plugins/         # Vue plugins (Vuetify, SweetAlert, ProgressBar)
│   ├── routes.js        # Vue Router configuration
│   └── utils/           # Utility functions
└── sass/
    └── app.scss         # Main stylesheet

routes/
├── api.php              # API routes
└── web.php              # Web routes (includes public API)

public/
└── uploads/             # Uploaded files (images, documents)
    ├── products/
    └── ...
```

🔒 Security

- Admin routes protected with Laravel Sanctum

- Password hashing

- CSRF protection

- XSS protection

- SQL injection protection (Eloquent ORM)

- Role-based access control (RBAC)

- Permission-based route protection

- Bot detection and filtering

- IP address tracking for security

📝 Notes

- This is a foundational structure that can be expanded

- All core models and relationships are set up

- Additional features can be added incrementally

- The platform is designed to be modular and configurable

- Product management includes comprehensive features for industrial/tech product websites

- File uploads are stored in `public/uploads/{folder}/` for easy access

- Dashboard includes AI-powered insights and real-time analytics

- All charts use Chart.js for modern, interactive visualizations

🛠️ Development

Adding a New Module

1. Create migration: `php artisan make:migration create_[module]_table`

2. Create model: `php artisan make:model [Module]`

3. Create controller: `php artisan make:controller Api/[Module]Controller --api`

4. Add to ModuleSeeder

5. Create Vue components

6. Update routes

Testing

```bash
php artisan test
```

📄 License

This project is open-sourced software licensed under the [MIT license](https://opensource.org/licenses/MIT).

🤝 Contributing

Contributions are welcome! Please feel free to submit a Pull Request.