

Voila CMS System Documentation

November 20, 2024

1 VOILA CMS SYSTEM DOCUMENTATION

2 SYSTEM OVERVIEW

Voila CMS is a modern, feature-rich Content Management System built with Laravel 12. It provides a complete solution for managing articles, categories, comments, and user interactions. The system features a beautiful responsive frontend for visitors and a powerful admin panel for content management.

3 WHAT THIS SYSTEM DOES

3.1 CONTENT MANAGEMENT

- Create, edit, and publish articles with rich content
- Organize articles into categories
- Upload and manage featured images
- Draft and publish workflow
- Article status management (draft, published, archived)

3.2 USER INTERACTION

- Allow users to browse and read articles
- Enable commenting system with moderation
- Provide contact form for inquiries
- User registration and authentication
- Role-based access control

3.3 ADMINISTRATIVE FUNCTIONS

- Dashboard with system statistics
- Complete content management capabilities
- Comment moderation and approval
- Contact message management

- User and category management
- Bulk operations for efficiency

3.4 CATEGORIZATION

- Create and manage content categories
- Filter articles by categories
- Category-based organization
- Visual category identification with colors

4 KEY FEATURES

5 FRONTEND FEATURES:

- Modern responsive design using Tailwind CSS
- Mobile-first approach for all devices
- Beautiful homepage with latest articles
- Article detail pages with full content display
- Category-based article filtering
- Search and browsing functionality
- Interactive comment system
- Professional contact form
- Social sharing capabilities
- SEO-friendly URL structure
- Clean and intuitive navigation

6 ADMIN PANEL FEATURES:

- Comprehensive dashboard with statistics
- Complete article CRUD operations
- WYSIWYG editor support for content creation
- Featured image upload and management
- Category management with visual elements
- Comment moderation system (approve/reject)
- Contact message management
- Bulk operations for efficiency
- User role management

- Article status controls
- Publishing workflow management
- System analytics and reporting

7 USER MANAGEMENT:

- Multi-role system (Guest, User, Editor, Administrator)
- Secure authentication system
- Role-based access control
- User registration and login
- Password management
- Session security
- Profile management

8 TECHNICAL FEATURES:

- Built with Laravel 12 framework
- Eloquent ORM for database management
- Migration system for version control
- Middleware for security and access control
- Form validation and CSRF protection
- Image upload with security validation
- Responsive design system
- RESTful API structure ready
- Secure file storage
- Error handling and logging

9 ACCESS LEVELS & PERMISSIONS

10 GUEST USERS:

- Browse published articles
- View categories and filter content
- Use contact form
- Register for account

11 REGISTERED USERS:

- All guest permissions plus:
- Comment on articles
- Manage personal account
- Receive notifications

12 EDITORS:

- All user permissions plus:
- Create and edit articles
- Upload and manage images
- Publish/unpublish own articles
- Manage assigned categories

13 ADMINISTRATORS:

- Full system access including:
- Complete article management (all users)
- Category management
- Comment moderation
- Contact message management
- User management
- Dashboard analytics
- System configuration

14 ACCESS URLs AND ROUTES

15 PUBLIC FRONTEND URLs:

- Homepage: /
- Article Details: /article/{slug}
- Category Pages: /category/{slug}
- Contact Form: /contact
- User Login: /login
- User Registration: /register

16 ADMIN PANEL URLs:

- Admin Dashboard: /admin
- Article Management: /admin/articles
- Create Article: /admin/articles/create
- Edit Article: /admin/articles/{id}/edit
- Article Actions: /admin/articles/{id}/toggle-status
- Category Management: /admin/categories
- Create Category: /admin/categories/create
- Edit Category: /admin/categories/{id}/edit
- Comment Moderation: /admin/comments
- View Comment: /admin/comments/{id}
- Approve Comment: /admin/comments/{id}/approve
- Reject Comment: /admin/comments/{id}/reject
- Bulk Actions: /admin/comments/bulk-action
- Contact Messages: /admin/contacts
- View Messages: /admin/contacts/{id}
- Bulk Actions: /admin/contacts/bulk-action
- Mark Reviewed: /admin/contacts/{id}/mark-reviewed

16.1 API ENDPOINTS (Available for future mobile apps)

- GET /api/articles - List all articles
- GET /api/articles/{slug} - Get specific article
- GET /api/categories - List all categories
- GET /api/categories/{slug}/articles - Get articles by category
- POST /api/articles/{slug}/comments - Add comment to article
- POST /api/contact - Submit contact form

17 DATABASE STRUCTURE

18 MAIN TABLES:

- users: User accounts with roles (admin, editor, user)
- articles: Article content with title, content, status, publish date
- categories: Content categories with name, description, color

- article_category: Many-to-many relationship between articles and categories
- comments: User comments with moderation status
- contacts: Contact form submissions

19 KEY FIELDS:

- Articles: title, slug, excerpt, content, featured_image, status, published_at
- Categories: name, slug, description, color, status
- Comments: content, status (pending, approved, rejected), article_id
- Contacts: name, email, phone, subject, message, status
- Users: name, email, password, role

20 SECURITY FEATURES

- CSRF Protection on all forms
- SQL Injection prevention through Eloquent ORM
- XSS Prevention through input sanitization
- Secure authentication system
- Role-based authorization middleware
- File upload validation and security
- Session security
- Password hashing
- Form validation
- Secure routing with middleware

21 SYSTEM REQUIREMENTS

- PHP 8.2 or higher
- Composer (PHP dependency manager)
- Node.js 18+ and NPM
- Web server (Apache/Nginx)
- Database (MySQL, PostgreSQL, or SQLite)
- Modern web browser for users

22 INSTALLATION STEPS

1. Clone or download the project
2. Run: composer install
3. Run: npm install
4. Configure database settings in .env
5. Run: php artisan key:generate
6. Run: php artisan migrate --seed
7. Run: php artisan storage:link
8. Run: php artisan serve
9. Access at <http://localhost:8000>

23 DEFAULT ADMIN ACCOUNTS

23.1 After installation, use these credentials

24 ADMIN ACCOUNT:

- Email: admin@example.com
- Password: password
- Role: Administrator

25 EDITOR ACCOUNT:

- Email: editor@example.com
- Password: password
- Role: Editor

26 USER ACCOUNT:

- Email: user@example.com
- Password: password
- Role: User

27 SUPPORT & MAINTENANCE

28 TROUBLESHOOTING:

- Check error logs in storage/logs/laravel.log
- Verify database connections
- Ensure proper file permissions
- Clear caches when needed

29 REGULAR MAINTENANCE:

- Monitor system performance
- Review and moderate comments
- Update content regularly
- Backup database and files
- Keep system updated

30 CUSTOMIZATION OPTIONS:

- Modify colors and branding in CSS files
- Add custom helper functions
- Extend models and controllers
- Create new views and templates
- Add new features through migrations

31 TECHNOLOGIES USED

31.1 Backend

- Laravel 12 (PHP Framework)
- Eloquent ORM
- MySQL
- PHP 8.3+

31.2 Frontend

- Tailwind CSS
- JavaScript
- HTML5
- CSS3

31.3 Development

- Composer
- Artisan Console
- Laravel Mix/Vite

31.4 File Management

- Laravel Storage
- File Upload Handling
- Image Processing

END OF DOCUMENTATION

Document Version: 1.0

Last Updated: November 20, 2024

System Version: Voila CMS v1.0

Framework: Laravel 12

For technical support or questions, contact your system administrator or refer to the project documentation and user manual.