# TerraFlow CMS User Manual

Complete Guide to Content Management System

terraFlow Development Team

August 4, 2025

# Introduction

## What is TerraFlow CMS?

TerraFlow CMS is a modern, enterprise-grade Content Management System built on the robust Symfony 6.4 framework. Designed with flexibility and extensibility in mind, it provides a comprehensive solution for managing websites, content, users, and multimedia files. The system features a powerful plugin architecture that allows developers and content managers to extend functionality without modifying the core system.

This CMS is particularly well-suited for organizations that need a scalable, secure, and feature-rich content management platform with advanced user management capabilities.

### Key Features

- **Modern Architecture**: Built on Symfony 6.4 framework with PHP 8.1+ support, following best practices and modern development standards
- **Advanced User Management**: Comprehensive role-based access control system with granular permissions and roles
- **Intelligent Content Management**: Hierarchical page and category system with custom routing, SEO optimization
- **Professional Media Management**: Advanced file upload system with automatic image compression, WebP conversion, and organized storage
- **Extensible Plugin System**: Modular architecture allowing for custom functionality through a well-defined plugin interface
- **Multilingual Support**: Built-in internationalization with German and English translations, extensible for additional languages
- **Modern Admin Interface**: Responsive, intuitive administration dashboard with real-time statistics and quick actions
- **SEO Optimization**: Advanced SEO features including custom meta tags, URL routing, sitemap generation, and structured data
- **Performance Optimization**: Multi-layer caching system, image optimization, and performance monitoring tools
- **Enterprise Security**: Role-based permissions, secure authentication, CSRF protection, and input validation
- **Console Tools**: Comprehensive command-line interface for system administration and maintenance
- **Theme System**: Flexible theming architecture with Twig templating engine

- **API Ready**: RESTful API endpoints for headless implementations and third-party integrations

#### System Requirements

#### Minimum Requirements

- **PHP**: 8.1 or higher with required extensions
- **Database**: MySQL 5.7+ or MariaDB 10.3+ (managed via Doctrine ORM)
- **Web Server**: Apache 2.4+ or Nginx 1.18+
- **PHP Extensions**: ctype, gd, iconv, zip, intl, json, mbstring, xml
- **Memory**: Minimum 256MB RAM (512MB recommended for production)
- **Storage**: Minimum 100MB free space (more for media files)

#### Recommended Production Environment

- **PHP**: 8.2+ with OPcache enabled
- **Database**: MySQL 8.0+ or MariaDB 10.6+
- **Memory**: 1GB+ RAM
- **Storage**: SSD storage for better performance
- **Security**: SSL/TLS certificate for HTTPS
- **Monitoring**: Error tracking and performance monitoring tools

#### Development Environment

- **Composer**: Latest version for dependency management
- **Node.js**: For asset compilation (if using Webpack Encore)
- **Git**: Version control integration
- **IDE**: PHP-aware IDE with Symfony support (recommended: PhpStorm, VS Code)

### Installation

#### Prerequisites

Ensure you have Composer installed and your server meets the system requirements.

#### Installation Steps

1. **Download TerraFlow CMS**:

```
curl -L -o terraflow-cms.zip "https://itsandremov.de/terraflow-cms/latest"
unzip terraflow-cms.zip
cd terraflow-cms
```

2. **Install dependencies**:

```
composer install
```

3. **Configure environment**:

   - Copy `.env.example` to `.env`
   - Configure database connection
   - Set up your domain and environment settings

4. **Set up database**:

```
php bin/console doctrine:database:create
php bin/console doctrine:migrations:migrate
```

5. **Create admin user**:

```
php bin/console terraflow:admin:create
```

6. **Set up web server**: Point your web server document root to the **`public`**/ directory.

## User Management

### Understanding Users and Roles

TerraFlow CMS uses a comprehensive role-based access control system that allows fine-grained permission management.

#### User Entity Structure

Each user in the system has the following properties: - **Username**: Unique identifier for login - **Email**: User's email address - **Password**: Securely hashed using Symfony's password hasher - **Profile Image**: Optional user avatar - **Roles**: Collection of assigned roles determining permissions

#### Role Management

Roles define what actions users can perform in the system. The system includes:

- **Administrative Roles**: Full system access

- **Content Manager Roles**: Content creation and management
- **User Roles**: Basic access permissions
- **Custom Roles**: Configurable roles with specific permissions

### Creating Users

#### Via Admin Interface

1. Navigate to **Administration** → **User Management**
2. Click **"Create User"**
3. Fill in the required information:
    - Username (unique)
    - Email address
    - Password
    - Assign roles
    - Upload profile image (optional)
4. Click **"Save"** to create the user

#### Via Console Command

```
1 php bin/console terraflow:admin:create
```

This command creates an administrative user with full system access.

### Managing User Permissions

Permissions in TerraFlow CMS are defined by the `Permissions` class and include:

- `ALLOW_DASHBOARD_ACCESS`: Dashboard access
- `ALLOW_USER_VIEW`: View users
- `ALLOW_USER_CREATE`: Create new users
- `ALLOW_USER_EDIT`: Edit existing users
- `ALLOW_USER_DELETE`: Delete users
- `ALLOW_PAGE_VIEW`: View pages
- `ALLOW_PAGE_CREATE`: Create pages
- `ALLOW_PAGE_EDIT`: Edit pages
- `ALLOW_PAGE_DELETE`: Delete pages
- `ALLOW_MEDIA_UPLOAD`: Upload media files
- `ALLOW_MEDIA_VIEW`: View media library
- `ALLOW_MEDIA_DELETE`: Delete media files

- `ALLOW_PLUGINS_INSTALL`: Install plugins
- `ALLOW_PLUGINS_MANAGE`: Manage plugins

#### User Profile Management

Users can manage their own profiles through the admin interface:

1. Navigate to **Profile** in the admin menu
2. Update personal information
3. Change password
4. Upload/change profile image

## Content Management

### Pages

Pages are the core content units in TerraFlow CMS. Each page represents a web page on your site.

#### Page Structure

Pages contain the following elements:

- **Title**: Display title of the page
- **Content**: Main page content (HTML supported)
- **Route**: URL path for the page
- **Category**: Optional category assignment
- **Meta Tags**: SEO meta information
- **Custom Stylesheets**: Page-specific CSS
- **Custom JavaScript**: Page-specific JS (header and footer)
- **Navigation Settings**: Control navigation display
- **Permission Requirements**: Access control
- **Landing Page**: Designate as home page

#### Creating Pages

1. Navigate to **Administration** → **Pages Management**
2. Click **“Create Page”**

3. Configure page settings:
    - **Basic Information**: Title, content, route
    - **SEO Settings**: Meta tags, descriptions
    - **Advanced Settings**: Custom CSS/JS, permissions
    - **Navigation**: Display in navigation, show title
4. Save the page

#### Page Routing

TerraFlow CMS uses intelligent routing: - Custom routes can be defined per page - Landing pages use root URL (`/`) - Default pattern: `/content/{page-id}` - URL encoding is automatically handled

#### Page Permissions

Pages can have permission requirements: - Set `permissionRequired` to restrict access - Users must have the specified permission to view - Integrates with the role-based permission system

### Categories

Categories provide hierarchical organization for your content.

#### Category Features

- **Hierarchical Structure**: Parent-child relationships
- **Priority Ordering**: Custom sort order
- **Active/Inactive States**: Control visibility
- **Permission Control**: Access restrictions
- **Custom Views**: Template customization
- **Navigation Integration**: Automatic menu generation

#### Managing Categories

1. Navigate to **Administration** → **Categories**
2. Create new categories or edit existing ones
3. Configure:

- **Title**: Category name
- **Parent Category**: For hierarchy
- **Priority**: Sort order
- **Permissions**: Access control
- **Custom Views**: Template settings

#### Category Hierarchy

Categories support unlimited nesting levels:

```
Root Category
|-- Subcategory 1
|   |-- Sub-subcategory A
|   `-- Sub-subcategory B
`-- Subcategory 2
    `-- Sub-subcategory C
```

## Media Management

### File Upload System

TerraFlow CMS includes a comprehensive media management system for handling files, images, and other assets.

#### Supported File Types

The system accepts various file types based on server configuration: - **Images**: JPG, PNG, GIF, WebP, SVG - **Documents**: PDF, DOC, DOCX, TXT - **Archives**: ZIP (for plugin uploads) - **Other**: Based on MIME type configuration

#### Uploading Media

1. Navigate to **Administration** → **Media**
2. Click **“Upload Media”**
3. Select files or drag and drop
4. Provide descriptive names
5. Upload files

#### Media Processing

The system includes advanced media processing capabilities:

- **Image Compression**: Automatic optimization
- **WebP Conversion**: Modern format support
- **Thumbnail Generation**: Automatic resizing
- **Metadata Extraction**: File information storage

#### Using Media in Content

Uploaded media can be referenced in: - Page content (via URLs) - Category descriptions - Template customizations - Plugin configurations

### File Organization

Media files are stored in: - **Upload Directory**: `public/uploads/` - **Media Subdirectory**: `public/uploads/media/` - **Favicon Directory**: `public/uploads/favicons/`

File naming follows a secure pattern:

```
{md5-hash-of-unique-id}.{original-extension}
```

## Plugin System

### Plugin Architecture

TerraFlow CMS features a powerful plugin system that allows extending functionality without modifying core code.

#### Plugin Structure

Each plugin follows a standardized structure:

```
plugin/
`-- YourPlugin/
    |-- YourPlugin.php (Main plugin class)
    |-- Controller/ (Plugin controllers)
    |-- Resources/
    |   |-- config/
    |   |   `-- routes.yaml
    |   |-- views/ (Twig templates)
```

```
|   `-- translations/
`-- README.md
```

### Plugin Interface

All plugins must implement `PluginInterface` and extend `AbstractPlugin`:

```php
class YourPlugin extends AbstractPlugin
{
    public function getPluginInfo(): PluginInfo
    {
        return new PluginInfo(
            'YourPlugin',
            '1.0.0',
            'Your Name',
            'Plugin description'
        );
    }
}
```

### Plugin Management

#### Via Admin Interface

1. Navigate to **Administration** → **Plugins**
2. View available plugins
3. Install/Uninstall plugins
4. Activate/Deactivate plugins
5. Configure plugin settings

#### Via Console Commands **List all plugins**:

```bash
php bin/console terraflow:plugin:list
```

**Install a plugin**:

```bash
php bin/console terraflow:plugin:install PluginName
```

**Activate a plugin**:

```bash
php bin/console terraflow:plugin:activate PluginName
```

**Deactivate a plugin**:

```bash
php bin/console terraflow:plugin:deactivate PluginName
```

#### Plugin Installation

1. **Upload Method**: ZIP file upload through admin interface
2. **Manual Method**: Extract to `plugin/` directory
3. **Console Method**: Use installation commands

#### Plugin Development

#### Creating a Plugin

1. Create plugin directory: `plugin/YourPlugin/`
2. Create main plugin class extending `AbstractPlugin`
3. Implement required methods
4. Define plugin routes (optional)
5. Create templates (optional)
6. Add translations (optional)

**Plugin Hooks** Plugins can hook into system events: - **Installation/Uninstallation**: Setup/cleanup operations - **Activation/Deactivation**: Enable/disable functionality - **Route Loading**: Custom routing - **Template Loading**: Custom views - **Translation Loading**: Multilingual support

## Administration Interface

### Dashboard Overview

The admin dashboard provides a central control panel for managing your TerraFlow CMS installation.

#### Dashboard Features

- **System Statistics**: User, page, category, and media counts
- **Quick Actions**: Fast access to common operations
- **System Modules**: Access to administrative tools
- **Real-time Information**: Current time and system status

#### Navigation Structure

The admin interface is organized into logical sections:

1. **Dashboard**: Overview and quick access
2. **Administration**: User and role management
3. **Content Management**: Pages and categories
4. **Media Management**: File uploads and organization
5. **System Modules**: Advanced tools and settings
6. **Plugins**: Extension management
7. **Settings**: System configuration

## System Settings

### Global Configuration

Access system settings via **Administration** → **Settings**:

- **System Name**: Site title and branding
- **Timezone**: Default timezone setting
- **Theme Selection**: Choose active theme
- **Navigation Settings**: Menu configuration
- **SEO Settings**: Global meta tags
- **Security Settings**: Login and registration options

### Advanced Settings

- **Automatic Backup**: Enable/disable backups
- **Email Notifications**: System notifications
- **Maintenance Mode**: Site maintenance control
- **Custom CSS/JavaScript**: Global code injection
- **Favicon**: Site icon configuration

## User Interface

### Modern Design

The admin interface features: - **Responsive Design**: Works on all devices - **Dark/Light Themes**: User preference support - **Intuitive Navigation**: Clear menu structure - **Search Functionality**: Quick content finding - **Keyboard Shortcuts**: Efficient operation (Ctrl+S for save)

#### Accessibility

- **Screen Reader Support**: ARIA labels and descriptions
- **Keyboard Navigation**: Full keyboard accessibility
- **High Contrast**: Color scheme options
- **Font Scaling**: Responsive typography

## Console Commands

TerraFlow CMS includes numerous console commands for system administration and maintenance.

### System Management Commands

#### Cache Management

**Clear cache**:

```
php bin/console terraflow:cache:manage clear
```

**Warm up cache**:

```
php bin/console terraflow:cache:manage warmup
```

**Show cache statistics**:

```
php bin/console terraflow:cache:manage stats
```

**Invalidate specific cache tags**:

```
php bin/console terraflow:cache:manage invalidate --tags=pages,content
```

#### Database Operations

**Export database**:

```
php bin/console terraflow:database:export
```

**Import database**:

```
php bin/console terraflow:database:import backup.sql
```

**Reset database**:

```
php bin/console terraflow:database:reset
```

#### Content Management

**Activate all pages**:

```
php bin/console terraflow:pages:activate-all
```

**Activate all categories**:

```
php bin/console terraflow:categories:activate-all
```

#### Media Processing

**Compress images**:

```
php bin/console terraflow:media:compress
```

**Convert images to WebP**:

```
php bin/console terraflow:media:convert-webp
```

#### System Maintenance

**Enable maintenance mode**:

```
php bin/console terraflow:maintenance:start
```

**Disable maintenance mode**:

```
php bin/console terraflow:maintenance:stop
```

#### License Management

**Check license status**:

```
php bin/console terraflow:license --status
```

**Validate license**:

```
php bin/console terraflow:license --validate
```

### Plugin Commands

#### Plugin Management

**List all plugins**:

```
php bin/console terraflow:plugin:list
```

**Show only installed plugins**:

```
php bin/console terraflow:plugin:list --installed
```

**Show only active plugins**:

```
php bin/console terraflow:plugin:list --active
```

**Reload plugins**:

```
php bin/console terraflow:plugins:reload
```

#### Plugin Operations

**Install plugin**:

```
php bin/console terraflow:plugin:install PluginName
```

**Uninstall plugin**:

```
php bin/console terraflow:plugin:uninstall PluginName
```

**Activate plugin**:

```
php bin/console terraflow:plugin:activate PluginName
```

**Deactivate plugin**:

```
php bin/console terraflow:plugin:deactivate PluginName
```

## Advanced Features

### SEO Optimization

#### Page-Level SEO

Each page can be optimized with: - **Meta Title**: Custom page titles - **Meta Description**: Page descriptions - **Meta Keywords**: Relevant keywords - **Custom Meta Tags**: Additional SEO tags - **URL Structure**: SEO-friendly URLs

#### Global SEO Settings

System-wide SEO configuration: - **Global Meta Tags**: Site-wide tags - **Default Descriptions**: Fallback content - **Structured Data**: Schema.org markup - **Sitemap Generation**: Automatic XML sitemaps - **Robots.txt**: Search engine directives

### Performance Optimization

#### Caching System

TerraFlow CMS includes multiple caching layers:

1. **Application Cache**: Symfony cache component
2. **Settings Cache**: System configuration caching
3. **Plugin Cache**: Plugin state and configuration
4. **Template Cache**: Twig template compilation

#### Cache Configuration

Cache can be managed via: - **Admin Interface**: Cache management tools - **Console Commands**: Command-line operations - **Automatic Invalidation**: Smart cache updates

#### Image Optimization

- **Automatic Compression**: Reduce file sizes
- **WebP Conversion**: Modern format support
- **Lazy Loading**: Improve page load times
- **Responsive Images**: Multiple size variants

### Security Features

#### Authentication

- **Secure Password Hashing**: Using Symfony's hasher
- **Remember Me**: Optional persistent login
- **Session Management**: Secure session handling
- **Login Attempts**: Brute force protection

#### Authorization

- **Role-Based Access**: Granular permissions
- **Permission Inheritance**: Hierarchical roles
- **Page-Level Security**: Content access control
- **API Security**: Protected endpoints

#### Data Protection

- **Input Validation**: XSS protection
- **SQL Injection Prevention**: Doctrine ORM
- **CSRF Protection**: Form security
- **File Upload Security**: MIME type validation

### Internationalization

#### Supported Languages

- **German (de)**: Complete translation
- **English (en)**: Complete translation
- **Extensible**: Additional languages via plugins

#### Translation Management

Translations are managed through: - **YAML Files**: Standard translation format - **Admin Interface**: Translation management tools - **Plugin Translations**: Independent language support

#### Language Switching

- **Automatic Detection**: Browser preference
- **User Selection**: Manual language choice
- **URL-Based**: Language in URL structure

# Customization

## Themes

### Theme System

TerraFlow CMS supports multiple themes: - **Default Theme**: Built-in TerraFlowCMS theme - **Plugin Themes**: Themes provided by plugins - **Custom Themes**: User-created themes

### Creating Custom Themes

1. Create theme directory structure
2. Implement base templates
3. Define theme configuration
4. Register theme in system

### Theme Structure

```
templates/
`-- your-theme/
    |-- base.html.twig
    |-- page.html.twig
    |-- category.html.twig
    `-- elements/
        `-- header/
            `-- nav-item.html.twig
```

## Template Customization

### Twig Templates

TerraFlow CMS uses Twig templating engine: - **Template Inheritance**: Extend base templates - **Block System**: Override specific sections - **Custom Functions**: terraFlow-specific helpers - **Asset Management**: CSS/JS integration

### Available Templates

- **Frontend Templates**: Public-facing pages
- **Admin Templates**: Administration interface
- **Email Templates**: System notifications
- **Error Templates**: Error pages (404, 500, etc.)

### CSS and JavaScript

#### Global Customization

Add custom styles and scripts: - **Global CSS**: System-wide stylesheets - **Global JavaScript**: Site-wide functionality - **Page-Specific**: Per-page customization - **Plugin Assets**: Extension-provided assets

#### Asset Management

- **Asset Mapper**: Symfony asset management
- **Webpack Encore**: Advanced asset processing
- **CDN Integration**: External asset delivery
- **Compression**: Minification and optimization

## Troubleshooting

### Common Issues

#### Installation Problems

**Database Connection Errors**: - Verify database credentials in `.env` - Ensure database server is running - Check firewall settings

**Permission Errors**: - Set correct file permissions on `var/` and `public/uploads/` - Ensure web server can write to cache directory

**Memory Errors**: - Increase PHP memory limit - Optimize database queries - Enable caching

#### Runtime Issues

**Plugin Errors**: - Check plugin compatibility - Verify plugin installation - Review error logs

**Template Errors**: - Check template syntax - Verify template paths - Clear template cache

**Performance Issues**: - Enable caching - Optimize database - Compress images

## Debugging

### Error Logging

TerraFlow CMS logs errors to: - **Application Logs**: `var/log/` - **Web Server Logs**: Apache/Nginx logs - **Database Logs**: MySQL/MariaDB logs

### Debug Mode

Enable debug mode for development:

```
APP_ENV=dev
APP_DEBUG=true
```

### Profiler

Symfony Web Profiler provides: - **Request Information**: Timing and memory usage - **Database Queries**: Query performance - **Template Rendering**: Twig performance - **Cache Operations**: Cache hit/miss ratios

## Getting Help

### Documentation

- **This Manual**: Comprehensive user guide
- **API Documentation**: Technical reference
- **Plugin Documentation**: Extension guides

### Support Channels

- **Issue Tracker**: Bug reports and feature requests
- **Community Forum**: User discussions
- **Email Support**: Direct assistance

### Contributing

- **Bug Reports**: Help improve the system
- **Feature Suggestions**: Propose enhancements

- **Code Contributions**: Submit pull requests
- **Documentation**: Improve guides and manuals

# Appendices

## A. Configuration Reference

### Environment Variables

```
# Database Configuration
DATABASE_URL=mysql://user:password@127.0.0.1:3306/terraflow

# License Configuration
TERRAFLOW_LICENSE_KEY=your-license-key

# Security Configuration
CLOUDFLARE_CAPTCHA_SECRET=your-captcha-secret

# Application Environment
APP_ENV=prod
APP_DEBUG=false
APP_SECRET=your-app-secret
```

### Service Configuration

Key service parameters: - `media_directory`: Media upload location - `favicons_directory`: Favicon storage - `terraflow_license_key`: License configuration - `captcha_secret`: Security configuration

## B. Database Schema

### Core Entities

1. **User**: System users and authentication
2. **Role**: User roles and permissions
3. **Permission**: System permissions
4. **Page**: Content pages
5. **Category**: Content organization
6. **Media**: File management
7. **Settings**: System configuration
8. **Plugin**: Plugin management
9. **Translation**: Internationalization

10. **Redirect**: URL redirection
11. **ContactRequest**: Contact form submissions

#### Entity Relationships

- Users have many Roles (many-to-many)
- Pages belong to Categories (many-to-one)
- Categories have hierarchical relationships (self-referencing)
- Pages can have permission requirements
- Media files are managed independently

### C. API Reference

#### REST API Endpoints

TerraFlow CMS provides REST API access: - **Authentication**: Token-based access - **Content Management**: CRUD operations - **Media Management**: File operations - **User Management**: Account operations

#### API Security

- **Authentication Required**: All endpoints protected
- **Permission-Based**: Role-based access control
- **Rate Limiting**: Abuse prevention
- **CORS Support**: Cross-origin requests

### D. Performance Guidelines

#### Optimization Recommendations

1. **Enable Caching**: Use all available cache layers
2. **Optimize Images**: Compress and convert to WebP
3. **Database Tuning**: Optimize queries and indexes
4. **Server Configuration**: Use PHP OPcache
5. **CDN Integration**: Serve static assets from CDN

**Monitoring**

- **Performance Metrics**: Response times and throughput
- **Resource Usage**: CPU and memory monitoring
- **Error Rates**: Track system errors
- **User Analytics**: Usage patterns and behavior

---

*This manual covers TerraFlow CMS version 1.1.0. For the latest information and updates, please refer to the official documentation and release notes.*