

# Kesgrave Town Council CMS - Complete Package

## Package Contents

This package contains the complete Kesgrave Town Council Content Management System with all functionality implemented:

### Main CMS File:

- `cms_final_complete.py` - The complete, fully functional CMS

### Key Features Implemented:

- **Councillor Management** - Complete with photos, ward tags, social media
- **Ward Tag System** - Color-coded tag management
- **Content Management** - Categories, subcategories, and content pages
- **Event Management** - Enhanced with gallery, multiple categories, downloads
- **File Upload System** - Images, documents, and media management
- **Rich Text Editor** - Full WYSIWYG content editing
- **Dashboard** - Statistics and quick actions

## Setup Instructions

### Requirements:

```
pip install flask flask-sqlalchemy flask-login werkzeug
```

### To Run:

1. Save `cms_final_complete.py` to your computer
2. Open terminal/command prompt in that folder
3. Run: `python cms_final_complete.py`
4. Open browser to: `http://localhost:8038`
5. Login with: **admin / admin**

## Database:

- SQLite database will be created automatically on first run
- Protected categories and sample data will be initialized
- Upload directories will be created automatically

## What You Can Test/Populate:

### Councillors Section:

- Add councillor profiles with photos
- Assign ward tags with colors
- Add social media links
- Manage qualifications and contact info

### Ward Tags:

- Create color-coded ward tags
- Assign tags to councillors
- Manage tag descriptions

### Events:

- Create events with categories
- Add event galleries with multiple images
- Set up booking and contact information
- Add related links and downloads

### Event Categories:

- Manage event categories with colors and icons
- Organize events by type

### Content Management:

- **Categories:** News, Council Information, Meetings, Financial Information, Reporting Problems (protected)
- **Subcategories:** Especially for Meetings (Annual Town Meetings, Community and Recreation, etc.)
- **Content Pages:** Full rich text content with galleries, related links, downloads
- **Date Management:** Creation, approval, review dates

# Continuation Prompt for Future Tasks

If you need to continue working on this CMS in a new task/session, use this prompt:

---

## CONTINUATION PROMPT:

"I have a complete Kesgrave Town Council CMS that needs further development. The CMS includes:

- Complete Councillor management with ward tags
- Enhanced Event management with galleries and categories
- Comprehensive Content management with categories/subcategories
- Rich text editing and file upload capabilities
- Dashboard with statistics

The main CMS file is `cms_final_complete.py` which runs on Flask with SQLAlchemy. It includes:

### Database Models:

- Councillor, Tag, CouncillorTag (councillor management)
- ContentCategory, ContentSubcategory, ContentPage, ContentGallery, ContentRelatedLink, ContentRelatedDownload (content management)
- Event, EventCategory, EventGallery, EventCategoryAssignment, EventRelatedLink, EventRelatedDownload (event management)

### Key Features:

- Protected categories: News, Council Information, Meetings, Financial Information, Reporting Problems
- Protected subcategories for Meetings: Annual Town Meetings, Community and Recreation, Finance and Governance, Planning and Development
- Rich text editor with Quill.js
- File upload system with proper validation
- UK date formatting throughout
- Responsive Bootstrap 5 interface

**Current Status:** All core functionality is complete and working. The CMS runs on port 8038 with admin/admin login.

Please set up a working sandbox environment with this CMS and help me with [DESCRIBE YOUR SPECIFIC NEEDS HERE]."

---

# File Structure

```
kesgrave_cms_complete/
├── cms_final_complete.py      # Main CMS application
├── uploads/                  # Upload directories (auto-created)
│   ├── councillors/         # Councillor photos
│   ├── content/             # Content files
│   │   ├── gallery/         # Content gallery images
│   │   └── downloads/       # Content downloads
│   ├── events/              # Event files
│   │   ├── gallery/         # Event gallery images
│   │   └── downloads/       # Event downloads
└── README.md                # This file
```

## Technical Details

- **Framework:** Flask with SQLAlchemy ORM
- **Database:** SQLite (auto-created)
- **Frontend:** Bootstrap 5 + Font Awesome
- **Rich Text:** Quill.js editor
- **File Uploads:** Werkzeug secure filename handling
- **Authentication:** Flask-Login with simple admin/admin
- **Date Format:** UK format (DD/MM/YYYY)

## Next Steps

1. **Populate Data:** Use the CMS interface to add your councillors, events, and content
2. **Customize:** Modify colors, styling, or add new features as needed
3. **Deploy:** When ready, deploy to a production server with proper security
4. **Backup:** Regularly backup your SQLite database file

## Support

If you need to continue development or encounter issues, use the continuation prompt above in a new task to get immediate assistance with the existing codebase.