



Business Requirements Document for Project “IT-RAYS website Page Builder”

| | |
|-----------------|--|
| Presented to | IT-RAYS |
| Project Name | IT-RAYS Page Builder Business Requirements |
| Document Number | IRTRD001 |
| Document Type | Technical Requirements |
| Date | 14-July-2025 |
| Version | 1.0.0 |
| Prepared By | Mohamed Montaser (CEO) |

1. Summary

This document outlines the business requirements for the development of a comprehensive Content Management System (CMS). The primary objective is to provide a user-friendly, powerful, and flexible platform for content creation, management, and publishing. A key differentiator will be its integrated, Elementor-like drag-and-drop Page Builder. This CMS aims to streamline content workflows, enhance content customization capabilities, and support various content types including dynamic pages, posts, and a centralized media library.

2. Project Goals & Objectives

- **Empower Content Creators:** Provide intuitive tools for users to create and manage diverse content without requiring technical expertise.
- **Enhance Content Customization:** Offer extensive design and layout flexibility for web pages through a powerful page builder.
- **Streamline Publishing Workflows:** Facilitate efficient content creation, review, and publishing processes.
- **Centralize Asset Management:** Provide a single, organized repository for all digital media assets.
- **Ensure Scalability & Performance:** Build a system capable of handling growing content volumes and user traffic.
- **Maintain Security:** Protect content and user data through robust authentication and authorization mechanisms.

3. Core CMS Features: Business Requirements

3.1. Main CMS Dashboard

- **Purpose:** To provide administrators and content managers with a centralized, at-a-glance overview of the CMS activity, key metrics, and quick access to main functionalities.
- **Business Requirements:**
 - **BR-D-001:** The system SHALL provide a customizable dashboard accessible upon successful login.
 - **BR-D-002:** The dashboard SHALL display an overview of recent activities (e.g., recently published pages/posts, pending drafts).
 - **BR-D-003:** The dashboard SHALL provide quick links or widgets to frequently accessed modules (e.g., "Add New Page", "Add New Post", "Media Library").
 - **BR-D-004:** The dashboard SHALL display summary statistics (e.g., total pages, total posts, number of users).
 - **BR-D-005:** The dashboard SHALL be intuitive and easy to navigate for users with varying technical proficiencies.
 - **BR-D-006:** the system SHALL support Multi-languages (En & Ar) as initial languages.

3.2. Pages Module

- **Purpose:** To enable users to create, manage, and publish dynamic web pages with highly customizable layouts using the integrated Page Builder.
- **Business Requirements:**
 - **BR-P-001:** The system SHALL allow authenticated users (with appropriate roles) to create new pages.

- **BR-P-002:** Users SHALL be able to define a unique, SEO-friendly slug for each page.
- **BR-P-003:** The system SHALL store page content generated by using a rich text editor (e.g., TinyMCE, TipTap)..
- **BR-P-004:** The system SHALL store page builder content as a structured JSON array, generated by the Page Builder, allowing for dynamic rendering.
- **BR-P-005:** Users SHALL be able to edit existing pages by launching the integrated editor or Page Builder.
- **BR-P-006:** Users SHALL be able to view a list of all pages, including their status (Draft, Published, Archived), slug, and creation/last updated dates.
- **BR-P-007:** Users SHALL be able to change the status of a page (e.g., from Draft to Published, or Published to Archived).
- **BR-P-008:** The system SHALL provide functionality to delete pages.
- **BR-P-009 (Future):** The system SHALL support the creation and application of pre-built page templates.
- **BR-P-010 (Future):** The system SHOULD allow users to revert to previous versions of a page.

3.3. Posts Module

- **Purpose:** To enable users to create, manage, and publish dynamic, time-sensitive content such as blog articles or news updates.
- **Business Requirements:**
 - **BR-PO-001:** The system SHALL allow authenticated users (with appropriate roles) to create new posts.
 - **BR-PO-002:** Users SHALL be able to input post content using a rich text editor (e.g., TinyMCE, TipTap).

- **BR-PO-003:** Users SHALL be able to assign posts to one or more categories and tags for organization.
- **BR-PO-004:** Users SHALL be able to set a featured image for each post, selected from the Media Library.
- **BR-PO-005:** The system SHALL support various post statuses (Draft, Pending Review, Published, Scheduled).
- **BR-PO-006:** Users SHALL be able to assign an author to a post.
- **BR-PO-007:** The system SHALL provide functionality to view, edit, and delete existing posts.
- **BR-PO-008 (Future):** The system SHOULD allow for basic comment management for posts.

3.4. Media Library Module

- **Purpose:** To provide a central repository for all digital assets, allowing for easy upload, organization, and integration into content. (Like WordPress Media Library)
- **Business Requirements:**
 - **BR-ML-001:** The system SHALL allow users to upload various media file types (e.g., images: JPG, PNG, GIF, SVG; videos: MP4; documents: PDF).
 - **BR-ML-002:** Users SHALL be able to view all uploaded media in a grid or list format.
 - **BR-ML-003:** Users SHALL be able to search and filter media files based on name, type.
 - **BR-ML-004:** Users SHALL be able to edit metadata for each media item (e.g., title, caption, alt text for images).
 - **BR-ML-005:** The system SHALL provide a mechanism to select and insert media files directly into the Page Builder and Rich Text Editor for Posts.

- **BR-ML-006:** Users SHALL be able to delete media items.
- **BR-ML-007 (Future):** The system SHOULD provide basic image manipulation features (e.g., cropping, resizing, rotation).
- **BR-ML-008 (Future):** The system SHOULD support folder structures or advanced tagging for better organization.

3.5. Users Module

- **Purpose:** To manage user accounts and their access to the CMS, ensuring only authorized individuals can perform specific actions.
- **Business Requirements:**
 - **BR-U-001:** The system SHALL allow administrators to create, view, edit, and delete user accounts.
 - **BR-U-002:** Each user account SHALL be associated with at least one role.
 - **BR-U-003:** Users SHALL be able to manage their own profile information (e.g., name, email, password).
 - **BR-U-004:** The system SHALL implement a secure authentication mechanism (e.g., token-based via Laravel Sanctum) for user login.
 - **BR-U-005:** The system SHALL provide a secure password reset functionality.

3.6. Roles Module

- **Purpose:** To define and manage distinct roles within the CMS, each with specific permissions, enabling granular access control.
- **Business Requirements:**
 - **BR-R-001:** The system SHALL allow administrators to create, view, edit, and delete custom roles.
 - **BR-R-002:** Each role SHALL have a configurable set of permissions that dictate access to specific CMS functionalities (e.g., "can create pages", "can publish posts", "can manage users").

- **BR-R-003:** The system SHALL enforce role-based access control across all CMS modules and actions.
- **BR-R-004:** The system SHALL come with predefined default roles (e.g., Administrator, Editor, Author, Contributor).

3.7. Settings Module

- **Purpose:** To provide a centralized interface for managing global configurations and preferences that affect the entire CMS and the public-facing website. (Like WordPress Settings)
- **Business Requirements:**
 - **BR-S-001:** The system SHALL allow administrators to configure general site settings (e.g., Site Title, Timezone, Language).
 - **BR-S-002:** The system SHALL allow administrators to configure reading settings (e.g., default homepage, default posts page, number of posts per page).
 - **BR-S-003:** The system SHALL allow administrators to configure permalink structures for pages and posts.
 - **BR-S-004:** All settings changes SHALL be persistent and immediately affect the CMS and public site where applicable.
 - **BR-S-005 (Future):** The system SHOULD provide fields for integrating with external services (e.g., Google Analytics tracking ID, social media links).
 - **BR-S-006 (Future):** The system SHOULD allow for injecting custom CSS/JS globally.

3.8. Page Builder Module (Elementor-like)

- **Purpose:** To provide a visual, drag-and-drop interface for designing and customizing web page layouts and content.
- **Business Requirements:**

- **BR-PB-001:** The Page Builder SHALL provide a canvas area where users can drag and drop elements to build page layouts.
- **BR-PB-002:** The Page Builder SHALL support a comprehensive library of elements (e.g., Text Blocks, Images, Buttons, Videos, Forms, Sections, Columns, Dividers).
- **BR-PB-003:** Users SHALL be able to intuitively drag elements onto the canvas and rearrange them (sorting, reordering).
- **BR-PB-004:** Users SHALL be able to select any element on the canvas and access a dedicated styling/properties panel to customize its appearance and content.
- **BR-PB-005:** The styling panel SHALL offer extensive customization options for each element, including but not limited to:
 - Typography (font family, size, weight, line height, color).
 - Colors (text, background, borders).
 - Spacing (padding, margin).
 - Borders (width, style, radius).
 - Backgrounds (color, image, gradient, video).
 - Responsive settings (visibility, sizing adjustments per breakpoint).
 - Alignment (text, element).
 - Links (internal, external).
- **BR-PB-006:** The Page Builder SHALL provide responsive view modes (e.g., Desktop, Tablet, Mobile) to preview how the page looks on different devices.
- **BR-PB-007:** Users SHALL be able to save their page designs, which will be stored as a JSON array in the database.
- **BR-PB-008:** The Page Builder SHALL allow users to load existing page content from the database for editing.

- **BR-PB-009:** The Page Builder SHALL integrate with the Media Library for easy insertion of images and videos.
- **BR-PB-010:** The Page Builder SHALL support undo/redo functionality for editing actions.
- **BR-PB-011:** Users SHALL be able to save a created page as a template for future reuse.
- **BR-PB-012:** Users SHALL be able to load and apply pre-built templates to start a new page.

4. General Non-Functional Requirements

- **Performance:** The CMS and public-facing pages must load quickly (target < 2 seconds for core pages). API response times should be optimized.
- **Security:**
 - All user data and content must be protected.
 - Robust authentication and authorization mechanisms are essential.
 - Protection against common web vulnerabilities (XSS, SQL Injection, CSRF).
- **Usability:** The CMS interface must be intuitive, easy to learn, and efficient for content creators and administrators.
- **Scalability:** The architecture should support future growth in content volume and user base.
- **Maintainability:** Codebase should be clean, well-documented, and adhere to established coding standards.
- **Accessibility:** The public-facing website generated by the CMS should aim for WCAG 2.1 AA compliance.

5. Out of Scope for Initial Release (Phase 1)

- E-commerce functionality.
- Advanced SEO features beyond basic meta tags and slugs.
- Complex analytics integration beyond basic tracking code.
- Public-facing user registration/profiles (only admin users initially).
- Advanced image manipulation within the Media Library (cropping, filters).
- Version control for Posts.
- Comment management for Posts.

6. Assumptions

- Dedicated Backend, Frontend, and QA resources are available.
- All necessary third-party licenses (if any) will be acquired.
- Hosting infrastructure will be provided and configured.
- Stakeholders will be available for regular feedback and sprint reviews.

7. Success Metrics

- **User Adoption:** Percentage of content creators actively using the CMS.
- **Content Creation Efficiency:** Time taken to create and publish a standard page/post.
- **Page Builder Usage:** Frequency and diversity of elements used in the Page Builder.
- **System Stability:** Number of critical bugs reported in production.
- **Performance Benchmarks:** Meeting load time and API response targets.

Thanks for reading...