

Layout & Design Templates

Complete guide to page templates, email templates, section configuration, thumbnail management, and design consistency

Version 1.1 | Updated 2026-04-11 | QA Teams, Platform Admins, Customer Operators, Implementation Teams

[1. Executive Overview](#)

[What This Module Does](#)

The **Layout & Design — Page & Email Template System** provides the complete visual management layer for the Hyphen Publishing Platform. It consists of two interconnected systems:

1. **Page Template System** — A visual drag-and-drop page builder that lets admins create, configure, and publish page layouts across 21 page categories (homepage, article, section, author, subscription, search, campaign, archive, shop, etc.). Published templates render on the Reader Portal as the public-facing magazine.
2. **Email Template System** — A template management system for all transactional, lifecycle, notification, and marketing emails. Admins can customize email content, branding, and behavior through a WYSIWYG editor with variable substitution, preview, and test-send capabilities.

[Who Uses It](#)

User	Role
Admins / Editors	Create, configure, and publish page templates; customize email templates
Content Editors	Use page templates to organize how content appears on the magazine
Marketing Teams	Create email campaigns using the template system
Readers	Experience the published page templates on the Reader Portal; receive styled emails
QA Teams	Validate template rendering, email delivery, and configuration workflows

[Business Problem Solved](#)

- Eliminates the need for code changes to modify page layouts or email styling
- Provides consistent branding across all reader-facing pages and emails
- Enables non-technical staff to control magazine appearance and communication
- Supports 20+ page types with 80+ template variants and 176+ section components
- Centralizes email branding, variable management, and delivery tracking

2. Feature Scope

In Scope

Page Template System

Feature	Description
Template Gallery	Browse and select from 80+ built-in templates across 21 categories
Visual Drag-Drop Editor	Canvas-based editor with section palette, properties panel, and live preview
Section System	176+ pre-built section components (hero, grid, list, detail, campaign, etc.)
Section Properties	Configurable properties per section (content source, filters, layout, visibility)
Template Versioning	Major/minor version tracking with rollback capability
Template Publishing	Publish templates to Strapi CMS for Reader Portal consumption
Page Categories	20 database-driven page categories with dynamic management
Static Pages	About, Privacy, Terms, and custom static page management
Homepage Editor	Dedicated homepage editing with featured carousel and section management
Banner Management	Promotional banners with scheduling (start/end dates)
Navigation Editor	Header and footer link management
Theme & Appearance	Global theme customization (colors, typography, spacing, header/footer styles)
Campaign Landing Pages	Template-driven campaign pages
Preview Mode	Preview templates before publishing with session-based preview API

Email Template System

Feature	Description
Email Template CRUD	Create, edit, delete, and manage email templates
Variable System	<code>{{variable}}</code> substitution in subject and body
Template Preview	In-editor preview with branding wrapper
Test Email	Send test email per template with <code>[TEST]</code> prefix
Email Branding	Global branding (logo, colors, footer, social links) via preferences
Email Campaigns	Campaign builder with audience targeting and scheduling
Email Logs	Full delivery log with status tracking (sent/failed/bounced)
Template Seeding	One-click seeding of 20 default templates
Multi-Provider Support	SMTP, SendGrid, Mailchimp with admin-selectable provider
Unsubscribe Compliance	RFC 8058, GDPR, CAN-SPAM compliant unsubscribe system
Email Classes	4 email classes with appropriate unsubscribe and compliance behavior

Out of Scope

- A/B testing of page templates (not implemented)

- Drag-and-drop email editor (HTML-based editing only)
- Real-time collaborative editing (single editor at a time)
- Template marketplace or third-party template import

[Related Modules / Dependencies](#)

Module	Relationship
Content Management	Templates display content from Strapi CMS
Reader Portal	Renders published page templates
Admin Settings & RBAC	Controls permissions for template management
Notifications & Email	Uses email templates for all outbound emails
Marketing & Outreach	Uses email campaign system
User & Subscription	Subscription lifecycle emails use templates

3. Roles Involved

Page Template System

Role	Capabilities
Super Admin	Full access to all template operations
Admin	Create, edit, publish, delete templates; manage categories, static pages, navigation, appearance
Editor	View templates; limited to editing page configurations assigned to their scope
Viewer	Read-only access to template gallery

Email Template System

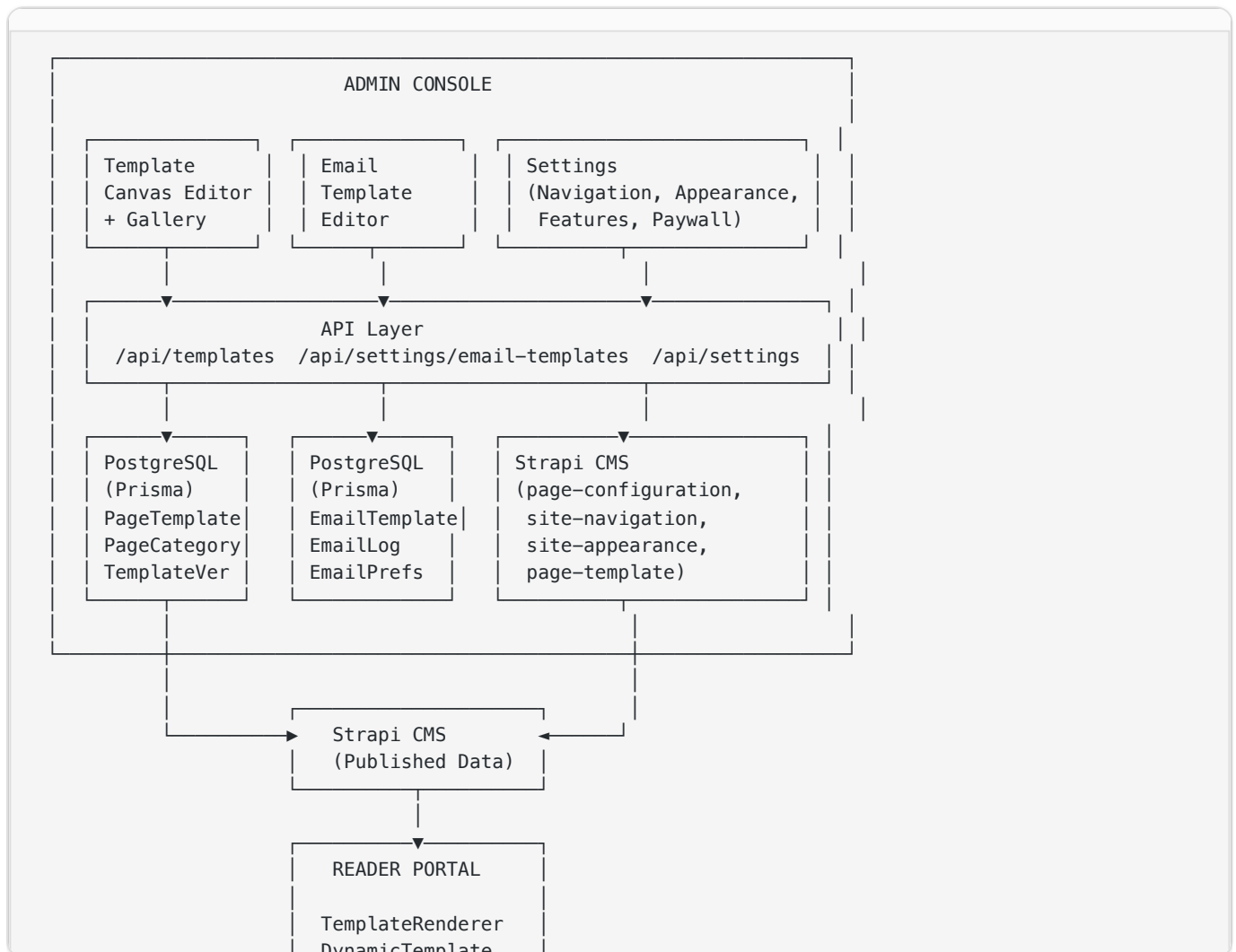
Role	Capabilities
Super Admin	Full access to all email settings
Admin (with <code>SETTINGS_UPDATE</code>)	Create, edit, delete email templates; manage preferences; send test emails; view logs
Admin (with <code>SETTINGS_READ</code>)	View email templates, preferences, SMTP status, logs
Marketing (with campaign permissions)	Create and send email campaigns

RBAC Permissions Required

Permission Key	Controls
<code>homepage: update</code>	Template CRUD, page category management
<code>homepage: read</code>	View templates and page configurations
<code>SETTINGS_READ</code>	View email templates, preferences, SMTP status
<code>SETTINGS_UPDATE</code>	Edit email templates, preferences; send test emails; seed templates
<code>NAVIGATION_READ</code>	View navigation and appearance settings (also used by Reader Portal internal token)

4. Architecture & Design Overview

System Architecture



Data Flow — Page Templates

1. Admin creates/edits template in Canvas Editor
2. API saves to PostgreSQL (PageTemplate + TemplateVersion) and writes versioned JSON file to `/data/templates/`
3. Admin publishes template → API syncs to Strapi CMS (`page-configuration` content type)
4. Reader Portal fetches page configuration from Strapi → resolves template → renders sections dynamically
5. Revalidation webhook triggers ISR cache refresh on Reader Portal

Data Flow — Email Templates

1. Admin creates/edits email template in Editor
2. API saves to PostgreSQL (EmailTemplate model)

3. **System event** triggers email (subscription welcome, OTP, campaign send, etc.)
4. **Email Pipeline** resolves template from DB → replaces variables → wraps with branding → sends via SMTP/provider → logs to EmailLog
5. **Admin views** delivery status in Email Logs page

[Key Entities & Models](#)

[Page Template Models \(PostgreSQL/Prisma\)](#)

Model	Purpose
PageCategory	21 page categories (homepage, article, section, author, etc.)
PageTemplate	Template definitions with metadata, sections, styles
TemplateVersion	Version history per template (major.minor)
Banner	Promotional banners with scheduling
StaticPage	Static content pages (About, Terms, Privacy, etc.)

[Strapi Content Types](#)

Content Type	Kind	Purpose
page-configuration	Collection	Per-page-type template configuration (supports draft/publish)
page-template	Collection	Template metadata synced from admin
page-version	Collection	Version snapshots for rollback
site-navigation	Single	Header links, footer columns, social links
site-appearance	Single	Global theme (colors, fonts, spacing)
homepage	Single	Homepage template selection and hero config
static-page	Collection	Static page content

[Email Models \(PostgreSQL/Prisma\)](#)

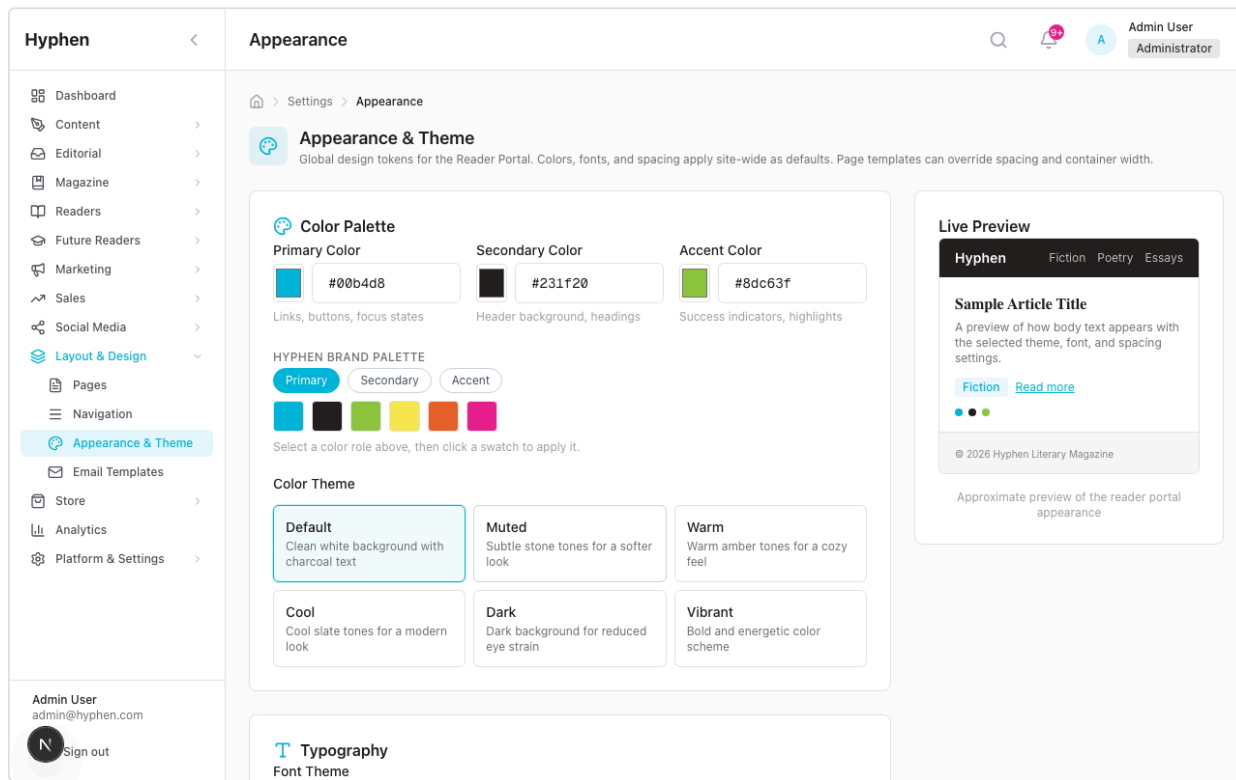
Model	Purpose
EmailTemplate	Template definitions with key, subject, HTML/text content, variables
EmailLog	Delivery log with status tracking, provider, error details
EmailPreferences	Singleton — branding, footer, unsubscribe, CRM settings
EmailCampaign	Campaign definitions with segment, status, metrics

[Settings Precedence Model](#)

The template system follows a 5-level precedence hierarchy:

Level 1: Global platform settings (DB)	– AUTHORITATIVE, NEVER overrideable
Level 2: Global site defaults (Strapi)	– AUTHORITATIVE with selective override
Level 3: Page template defaults	– OVERRIDE spacing/layout only
Level 4: Page-specific configuration	– Per-page overrides
Level 5: Content access rules	– OVERRIDE paywall per content

5. Prerequisites & Setup Requirements



Global appearance and theme settings control fonts, colors, and branding across all templates.

Infrastructure Prerequisites

Requirement	Details
Node.js	18.x minimum (20.x LTS recommended)
PostgreSQL	15.x+ with <code>uuid-oss</code> and <code>pgcrypto</code> extensions
Redis	7.x for caching
Strapi CMS	Running on port 1337 with admin access
Docker / Docker Compose	24.x / 2.20+

Environment Variables Required

Admin Console

Variable	Purpose	Example
<code>DATABASE_URL</code>	PostgreSQL connection	<code>postgresql://hyphen:...@localhost:5432/hyphen_admin</code>
<code>STRAPI_URL</code>	Strapi CMS URL	<code>http://localhost:1337</code>
<code>STRAPI_API_TOKEN</code>	Strapi API token	(from Strapi admin)
<code>NEXT_PUBLIC_READER_PORTAL_URL</code>	Reader portal URL (for revalidation)	<code>http://localhost:3001</code>
<code>READER_PORTAL_REVALIDATION_SECRET</code>	Shared secret for ISR revalidation	(random string)
<code>INTERNAL_API_TOKEN</code>	Shared secret for Reader Portal ↔ Admin API	(random string)
<code>EMAIL_ENABLED</code>	Enable email sending	<code>true</code>
<code>EMAIL_FROM</code>	Default sender address	<code>notifications@hyphen.co</code>
<code>SMTP_HOST</code> , <code>SMTP_PORT</code> , <code>SMTP_USER</code> , <code>SMTP_PASS</code>	SMTP configuration	(provider-specific)
<code>SENDGRID_API_KEY</code>	SendGrid API key (if using SendGrid)	<code>SG.xxxxx</code>
<code>VAPID_PUBLIC_KEY</code> , <code>VAPID_PRIVATE_KEY</code>	Push notification keys	(from <code>npx web-push generate-vapid-keys</code>)

Reader Portal

Variable	Purpose	Example
<code>NEXT_PUBLIC_STRAPI_URL</code>	Strapi CMS URL	<code>http://localhost:1337</code>
<code>STRAPI_API_TOKEN</code>	Strapi API token	(from Strapi admin)
<code>NEXT_PUBLIC_ADMIN_API_URL</code>	Admin console API URL	<code>http://localhost:3000</code>
<code>INTERNAL_API_TOKEN</code>	Same value as Admin Console	(matching secret)
<code>REVALIDATION_SECRET</code>	Same as <code>READER_PORTAL_REVALIDATION_SECRET</code>	(matching secret)
<code>NEXT_PUBLIC_SITE_NAME</code>	Site display name	Hyphen

Module Dependencies (Must Be Set Up First)

Dependency	Required For
Core Infrastructure (Docker, PostgreSQL, Redis)	Database and caching
Strapi CMS with content types created	Template sync, page configurations
Authentication & User Management	Admin login, RBAC
Prisma migrations run (<code>npx prisma migrate deploy</code>)	PageTemplate, EmailTemplate models

Seed Data Initialization

Before the template system is usable, seed data must be initialized:

1. **Page Categories** — Seed 21 built-in categories:

```
POST /api/page-categories/seed
```

2. **Page Templates** — Seed built-in templates:

```
POST /api/templates/seed
```

3. **Email Templates** — Seed default email templates (subscription, campaign, notification, OTP, and operational):

```
POST /api/settings/email-templates/seed
```

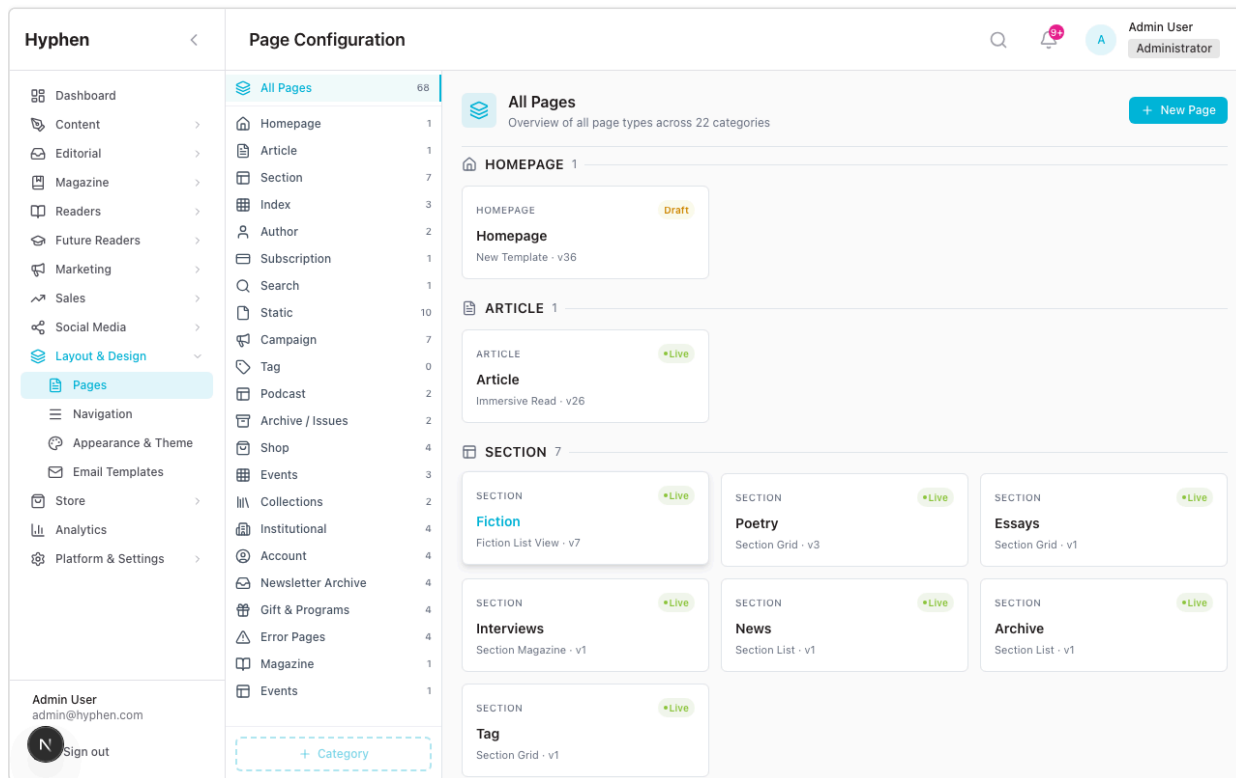
Or click the "Seed Missing" button in Settings > Email Templates.

Strapi Content Types Required

The following content types must exist in Strapi:

- `page-configuration` (collection, draft & publish enabled)
- `page-template` (collection)
- `page-version` (collection)
- `site-navigation` (single type)
- `site-appearance` (single type)
- `homepage` (single type)
- `static-page` (collection)

6. Configuration Guide — Page Template System



The page template editor with section layout, properties panel, and preview controls.

6.1 Accessing the Template System

Navigate to **Layout & Design** in the admin console sidebar:

Menu Item	Path	Purpose
Pages	<code>/layout/pages</code>	View and manage page types with their templates
Templates	<code>/layout/templates</code>	Browse the full template gallery
Categories	<code>/layout/categories</code>	Manage page categories
Static Pages	<code>/layout/static-pages</code>	Manage static content pages
Campaigns	<code>/layout/campaigns</code>	Campaign landing page layouts
Homepage	<code>/homepage</code>	Homepage-specific editor
Navigation	<code>/navigation</code> or <code>/settings/navigation</code>	Header/footer link management
Appearance	<code>/settings/appearance</code>	Theme customization

6.2 Understanding Page Categories

There are 21 built-in page categories:

Category	Slug	Description
Homepage	homepage	Main landing page
Article	article	Individual article pages
Section	section	Section/category listing pages (Fiction, Poetry, Essays, etc.)
Index	index	Index/directory pages
Author	author	Author profile pages
Subscription	subscription	Subscription/pricing pages
Search	search	Search results page
Static	static	Static content pages
Campaign	campaign	Campaign landing pages
Tag	tag	Tag listing pages
Archive	archive	Archive/back-issue pages
Magazine Issues	magazine-issues	Magazine issue viewer
Shop	shop	E-commerce pages
Event	event	Event pages
Podcast	podcast	Podcast pages
Collection	collection	Curated collection pages
Institutional	institutional	Institutional subscription pages
Account	account	Reader account pages
Newsletter Archive	newsletter-archive	Newsletter archive pages
Gift	gift	Gift subscription pages
Error	error	Error pages (404, 500)

[6.3 Working with the Template Canvas Editor](#)

[Opening the Editor](#)

1. Go to **Layout & Design > Pages**
2. Select a page type (e.g., Homepage)
3. Click on the assigned template or click **Edit Template**
4. The Canvas Editor opens with three panels:
 - **Left Panel:** Section Palette (drag sections from here)
 - **Center Panel:** Canvas Area (your page layout)
 - **Right Panel:** Properties Panel (configure selected section)

[Adding Sections](#)

1. In the **Section Palette** (left panel), browse section variants by category
2. **Drag** a section variant onto the Canvas Area

3. **Drop** it in the desired position (drop zones appear between existing sections)
4. The section appears on the canvas with default configuration

[Configuring Section Properties](#)

When you select a section on the canvas, the Properties Panel shows configuration options based on the section's behavior type:

Behavior Type	Properties Available
Listing	Content source (dynamic/manual/mixed), category filter, tag filter, max items, sort order, entity picker
Detail	Entity reference, display fields, related content config
Contextual	Relationship type, related entity config
Static	Title, subtitle, body text, CTA button, image

[Section Variant Categories](#)

Category	Examples	Count
Hero	HeroFullWidth, HeroImageLeft, HeroImageRight, HeroMinimal, HeroCarousel	5
Article Grids	Grid2Column, Grid3Column, Grid2WithSidebar, Grid4Carousel	4
Featured	FeaturedLeftGridRight, FeaturedAboveGrid, FeaturedMixed	3
Article Lists	ListHorizontalCards, ListVerticalCards, ListMinimal	3
Article Detail	ArticleHeader, ArticleContent, ArticlePaywallGate, etc.	10
Campaign	CampaignHero, CampaignForm, CampaignCountdown, etc.	11
Event	EventHero, EventListingCards, EventRsvpForm, etc.	7
Podcast	PodcastShowHeader, PodcastEpisodeList, PodcastMiniPlayer	6
Shop	ShopHero, ShopProductGrid, ShopProductHeader	9
Archive	ArchiveHero, ArchiveIssueGrid, ArchiveTimelineView	7
Newsletter	NewsletterArchiveHeader, NewsletterEditionGrid	4
And 10+ more categories	Sidebar, footer, account, institutional, error, etc.	100+

[Reordering Sections](#)

- **Drag** sections up or down on the canvas to reorder
- Section `order` property updates automatically

[Removing Sections](#)

- Select the section on the canvas
- Click the **Delete/Remove** button in the Canvas Toolbar or section context menu

[Live Preview](#)

- The **LiveTemplatePreview** component renders a real-time preview of the current template configuration
- Preview updates as you modify section properties

6.4 Template Versioning

Every template edit creates a version record:

Field	Description
Major version	Incremented for breaking changes (component removal, layout redesign)
Minor version	Incremented for non-breaking changes (section reorder, config update)
Change note	Optional description of what changed
Created by	Admin user who made the change

Viewing Version History:

1. Open a template in the editor
2. Click **Version History** to see all versions
3. Each version shows: version number, change note, author, timestamp

Rolling Back:

1. In the version history, find the desired version
2. Click **Rollback to this version**
3. A new version is created with the rolled-back content (non-destructive)

6.5 Publishing Templates

Publishing syncs the template configuration to Strapi CMS, making it available to the Reader Portal.

1. Open the template in the editor
2. Review the configuration
3. Click **Publish**
4. The API:
 - Sets template status to `active`
 - Syncs template data to Strapi `page-configuration` content type
 - Triggers Reader Portal ISR revalidation
5. Changes appear on the Reader Portal within seconds

6.6 Managing Static Pages

1. Go to **Layout & Design > Static Pages**
2. Click **Create** to add a new page (or select existing)
3. Fill in:

- **Title** — Page heading
- **Slug** — URL path (e.g., `about`, `privacy-policy`, `terms`)
- **Category** — Default: `static`
- **Template ID** — Optionally bind to a template
- **Content** — Rich text content
- **Active** — Toggle visibility

4. Click **Save**

6.7 Managing Navigation

Navigate to **Settings > Navigation** or **Navigation** page.

Header Links

- Add, edit, reorder, or remove main navigation links
- Each link has: label, URL, and optional icon
- Changes sync to Strapi `site-navigation` single type

Footer Configuration

- Configure footer columns (each with title and list of links)
- Add social media links (X/Twitter, Facebook, Instagram, LinkedIn, etc.)
- Set copyright text and company information

6.8 Theme & Appearance

Navigate to **Settings > Appearance**.

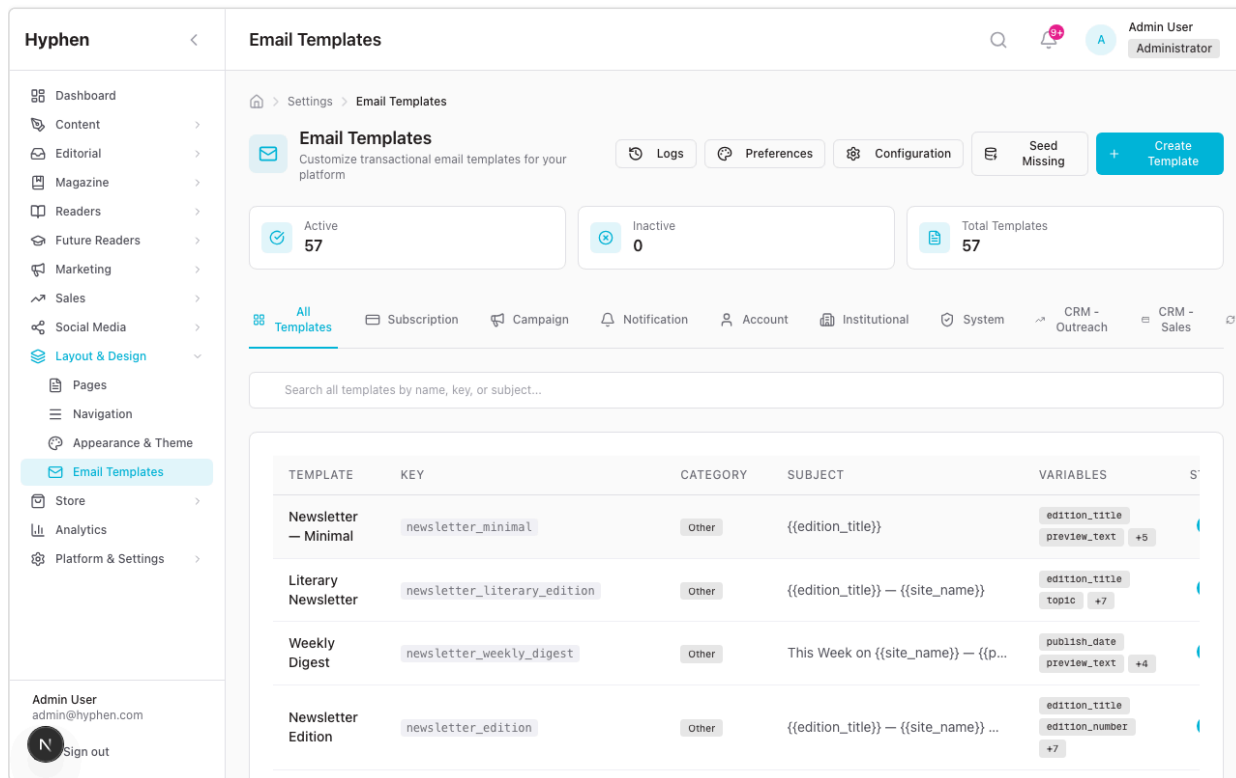
Available customizations:

Setting	Description
Color Theme	Primary, secondary, accent, background colors
Typography	Font family, heading sizes, body text sizes
Spacing	Container width, section spacing, content padding
Header Styles	Header background, text color, logo placement
Footer Styles	Footer background, text color, layout

Changes sync to Strapi `site-appearance` and are consumed by the Reader Portal's `ThemeProvider` which applies them as CSS variables and Tailwind classes.

Live Preview: The appearance editor includes a live preview panel showing how changes will look on the Reader Portal.

7. Configuration Guide — Email Template System



Manage email templates for notifications, campaigns, and transactional emails.

7.1 Accessing Email Templates

Navigate to the Layout & Design sidebar group (previously under Settings):

Menu Item	Path	Purpose
Email Templates	/settings/emails	Browse and manage all email templates
Email Configuration	/settings/emails/config	SMTP/provider settings
Email Preferences	/settings/emails/preferences	Branding, footer, unsubscribe settings
Email Logs	/settings/emails/logs	Delivery log viewer

7.2 Initial Setup — Seed Default Templates

Before email templates are usable, seed the defaults:

1. Go to Settings > Email Templates
2. Click the "Seed Missing" button in the header
3. This creates 20 default templates (only creates missing ones, never overwrites existing):

Category	Templates
Subscription (7)	Welcome, Renewal Reminder, Expiration, Payment Receipt, Future Readers Coupon, Institutional Invite, Gift Activation
Campaign (4)	Submission Approved, Submission Rejected, Invitation, New Campaign Notification
Notification (8)	New Submission, Reassigned, Action Needed, Author Response, Ready for Review, Publishing Approved, Change Requested, Article Live
System (1)	OTP Verification

7.3 Creating / Editing Email Templates

1. Go to **Settings > Email Templates**
2. Click **Create New** or select an existing template
3. Fill in:

Field	Description	Required
Key	Unique identifier (e.g., <code>subscription_welcome</code>)	Yes (immutable after creation)
Name	Display name (e.g., "Welcome Email")	Yes
Subject	Email subject line — supports <code>{{variables}}</code>	Yes
HTML Content	HTML email body — supports <code>{{variables}}</code>	Yes
Text Content	Plain-text fallback body	Recommended
Variables	List of variable names used in the template	Auto-detected
Active	Toggle template availability	Yes (default: true)

7.4 Using Template Variables

Variables are placeholders that get replaced with actual values at send time.

Syntax: `{{variableName}}` **Matching:** Case-insensitive (e.g., `{{userName}}` matches `username`)

System Variables (Auto-Injected)

These are always available in every template:

Variable	Source	Example Value
<code>{{site_name}}</code>	<code>NEXT_PUBLIC_SITE_NAME</code> env var	Hyphen
<code>{{site_url}}</code>	System settings	<code>https://hyphen.co</code>
<code>{{support_email}}</code>	CRM settings	<code>support@hyphen.co</code>
<code>{{current_year}}</code>	System	2026
<code>{{unsubscribe_url}}</code>	HMAC-generated per recipient	<code>https://hyphen.co/unsubscribe?token=...</code>

CRM Variables (Auto-Injected from CRM Settings)

Variable	Source
{{support_phone}}	CRM settings
{{admin_portal_url}}	CRM settings
{{reader_portal_url}}	CRM settings
{{user_guide_url}}	CRM settings
{{company_website}}	CRM settings
{{schedule_demo_url}}	CRM settings
{{pricing_page_url}}	CRM settings

Context-Specific Variables

Template Type	Variables
Subscription emails	{{userName}}, {{userEmail}}, {{subscriptionPlan}}, {{subscriptionEndDate}}, {{renewalLink}}
OTP	{{otpCode}}, {{expiryMinutes}}
Article notifications	{{articleTitle}}, {{articleAuthor}}, {{articleUrl}}
Campaign	{{campaignName}}, {{offerCode}}, {{redemption_instructions}}
Password reset	{{resetLink}}, {{userName}}

7.5 Previewing Email Templates

1. In the template editor, click **Preview**
2. The preview renders the template with:
 - Branding wrapper (logo, header, footer from Email Preferences)
 - Sample variable values
 - Actual styling as it would appear in email clients

7.6 Sending Test Emails

1. In the template editor, click **Send Test**
2. Enter a recipient email address
3. The system sends the email with:
 - [TEST] prefix in subject line
 - Sample variable values
 - Full branding wrapper
4. Check the Email Logs page to verify delivery

7.7 Configuring Email Branding (Preferences)

Navigate to **Settings > Email Templates > Preferences**:

Section	Settings
Header	Logo URL, background color
Footer	Company name, address, social media links, copyright text, custom HTML
Unsubscribe	Link text, preferences URL, grace period (0–30 days), confirmation required, reason options
Font & Colors	Font family, primary color
CRM	Support email, portal URLs, company website

“Important: Custom footer HTML is sanitized server-side to prevent XSS (strips scripts, iframes, event handlers, javascript: URLs).”

7.8 SMTP / Provider Configuration

Navigate to **Settings > Email Templates > Config**:

1. Select the email provider: **SMTP**, **SendGrid**, or **Mailchimp**
2. Enter provider credentials
3. Click **Test Connection** to verify
4. Save configuration

“Note: After changing provider configuration, the cached SMTP transporter may need a server restart to take effect (known anti-pattern, see [Limitations](#) section).”

7.9 Viewing Email Logs

Navigate to **Settings > Email Templates > Logs**:

Column	Description
Recipient	Email address + name
Template	Template key used
Subject	Rendered subject line
Status	sent , failed , bounced , pending
Provider	smtp , sendgrid , etc.
Sent At	Timestamp
Error	Error message if failed

Additional features:

- **Unsubscribe Analytics** — Collapsible section showing: total unsubscribes, by reason, by category, 30-day timeline
-

[8. How the Page Template System Works End to End](#)

[8.1 Template Lifecycle](#)

```

Create Template → Edit Sections → Configure Properties → Preview → Publish → Render on Reader Portal
                                                    |
                                                    Version Created
                                                    Sync to Strapi
                                                    Revalidate ISR
  
```

[8.2 What Happens When You Publish](#)

1. Template status set to `active` in PostgreSQL
2. Template data (sections, styles, banners) synced to Strapi `page-configuration`
3. Strapi stores as a published document (draft/publish workflow)
4. Reader Portal receives revalidation webhook
5. Next.js ISR regenerates affected pages
6. Readers see updated layout within seconds

[8.3 Reader Portal Rendering Pipeline](#)

1. Request arrives at Reader Portal (e.g., /fiction)
2. Next.js route resolves page type (e.g., "section")
3. Fetch page-configuration from Strapi for this page type
4. Extract `templateId`, `sections[]`, `banners[]`, `styles`
5. `TemplateRenderer` maps `templateId` → React component
6. `DynamicTemplate` iterates over `sections[]`:
 - a. Resolve section variant → `SectionRenderer` → actual component
 - b. Fetch section data based on `contentSource` (dynamic queries, manual IDs, mixed)
 - c. Apply personalization boost if `personalizationEnabled=true`
 - d. Render section component with data + properties
7. `PageStyleOverrides` applies page-level CSS overrides
8. `BannerRenderer` renders active banners (checks `startsAt/endsAt`)
9. `ThemeProvider` applies global appearance (colors, fonts, spacing)

[8.4 Section Data Resolution](#)

Each section can get content from different sources:

Content Source	How It Works
Dynamic	Queries Strapi with filters (category, tag, date range, sort order, max items)
Manual	Uses manually selected entity IDs (via EntityPicker)
Mixed	Combines manual pinned items + dynamic fill
Static	No data fetch — uses section configuration directly (text, images, CTAs)

9. How the Email Template System Works End to End

9.1 Email Pipeline Flow

1. System event triggers email (e.g., new subscription)
2. Determine email class (mandatory-transactional, transactional-lifecycle, operational, marketing)
3. Resolve template:
 - a. Look up EmailTemplate by key in DB
 - b. If not found, use hardcoded fallback
4. Replace variables:
 - a. Auto-inject system variables (site_name, support_email, etc.)
 - b. Auto-inject CRM variables
 - c. Replace caller-provided variables
 - d. Case-insensitive matching
5. Wrap with branding:
 - a. Load EmailPreferences from DB
 - b. Apply header (logo, colors)
 - c. Apply footer (company info, social links)
 - d. Add unsubscribe link (if email class requires it)
 - e. Add RFC 8058 List-Unsubscribe headers (if marketing/lifecycle)
6. Generate plain text fallback (strip HTML if textContent not provided)
7. Send via configured provider (SMTP/SendGrid/Mailchimp)
8. Log to EmailLog (template, recipient, status, provider, messageId)

9.2 Email Classes and Compliance

Class	When Used	Unsubscribe	Can Suppress?
mandatory-transactional	OTP, password reset, payment receipt	None	Never
transactional-lifecycle	Welcome, renewal, expiry, gift, institutional invite	Reader HMAC link	Respects reader prefs
operational	Institution inquiry, admin workflow notifications	None	N/A
marketing	Campaigns, newsletters, outreach	Reader HMAC link	Must respect opt-in

9.3 Unsubscribe Flow

1. Reader clicks unsubscribe link in email
2. Link contains HMAC token (generated from email + secret)
3. Reader Portal `/unsubscribe` page loads
4. If `requireConfirmation` enabled:
 - a. Shows confirmation message
 - b. Optional reason picker (admin-configured options)
 - c. Grace period notice if configured
5. Reader confirms → API processes unsubscribe:
 - a. Update `Reader.newsletterOptIn`
 - b. Set `AudienceContact.unsubscribedAt`
 - c. Create audit log entry
 - d. If grace period > 0: store as pending, process via daily cron
6. Reader sees confirmation with category-specific message

10. Business Logic & Rules

Page Template Rules

Rule	Description
Seed-on-write	Built-in templates are auto-copied to DB on first edit (preserves original)
DB precedence	DB templates override built-in templates by slug
Version always created	Every edit creates a TemplateVersion record
One active per page type	Only one published template can be active per page type at a time
Built-in protection	Built-in templates cannot be deleted (only archived)
Strapi sync on publish	Publishing always syncs to Strapi (not saved to Strapi on draft)
ISR revalidation	Publishing triggers Reader Portal cache invalidation
Section visibility	Sections with <code>isVisible: false</code> are skipped during rendering
Banner scheduling	Banners respect <code>startsAt</code> and <code>endsAt</code> — only active banners render
Personalization per-section	Sections with <code>personalizationEnabled: true</code> boost content matching reader preferences

Email Template Rules

Rule	Description
DB-first resolution	Templates resolved from DB by key; hardcoded fallback if not found
Case-insensitive variables	<code>{{userName}}</code> and <code>{{username}}</code> both match
System variables auto-injected	<code>site_name</code> , <code>support_email</code> , <code>current_year</code> always available
Caller overrides system	Caller-provided variables take precedence over system variables
Branding always applied	All emails wrapped with branding from EmailPreferences
Unsubscribe by class	Marketing and lifecycle emails include unsubscribe; transactional do not
AudienceContact suppression	Unsubscribed contacts excluded from all audience queries
Grace period enforcement	If configured, unsubscribes are deferred and processed by daily cron
EmailLog always created	Every email send (success or failure) creates an EmailLog entry
Test emails marked	Test sends have <code>[TEST]</code> subject prefix
Seed is additive	Template seeding only creates missing templates (by key), never overwrites

Style Precedence

Global Appearance (Strapi) → Template defaultStyles → Page-specific styles

Override scope:

- `spacing`, `containerWidth`, `fontFamily` can be overridden at template and page level

- Colors, global fonts, and navigation are authoritative from global settings
-

11. Use Cases & Operational Scenarios

Page Template Use Cases

#	Scenario	Steps
1	Change homepage layout	Open Homepage editor → Select different template → Rearrange sections → Publish
2	Add a featured section to category page	Go to Pages → Select Section page → Edit template → Drag "FeaturedAboveGrid" section → Configure content source → Publish
3	Create campaign landing page	Go to Layout > Campaigns → Create new → Select campaign template → Add hero, form, countdown sections → Configure → Publish
4	Customize article page	Go to Pages → Select Article → Edit template → Add/remove sections (header, content, comments, related, paywall gate) → Publish
5	Create static About page	Go to Static Pages → Create → Set title, slug, content → Activate
6	Revert a template change	Open template → Version History → Select previous version → Rollback
7	Add promotional banner	Homepage editor → Banner section → Create banner with image, link, schedule → Save
8	Change site theme	Settings > Appearance → Modify colors, fonts, spacing → Preview → Save
9	Update navigation	Settings > Navigation → Edit header links → Edit footer columns → Save
10	Preview before publish	Template editor → Preview mode → Verify layout → Close preview → Publish

Email Template Use Cases

#	Scenario	Steps
1	Customize welcome email	Email Templates → Find <code>subscription_welcome</code> → Edit subject/body → Preview → Test send → Save
2	Change email branding	Email Templates > Preferences → Update logo, colors, footer → Save
3	Add new notification template	Email Templates → Create New → Set key, name, subject, body, variables → Save
4	Verify email delivery	Email Logs → Filter by status/date → Inspect failed emails → Check error messages
5	Configure SMTP	Email Config → Select provider → Enter credentials → Test connection → Save
6	Send email campaign	Marketing > Email Campaigns → Create → Select template → Choose audience → Preview → Send
7	Review unsubscribe trends	Email Logs → Expand Unsubscribe Analytics → Review by reason, category, timeline
8	Seed missing templates	Email Templates → Click "Seed Missing" → Verify 20 templates created

12. End-to-End User Flows

Flow 1: Create and Publish a New Page Template

Step 1: Admin navigates to Layout & Design > Pages
Step 2: Selects a page type (e.g., "Section")
Step 3: Clicks "Edit Template" or creates a new template
Step 4: Canvas Editor opens with three panels
Step 5: Admin drags sections from the palette onto the canvas:
– HeroFullWidth (hero section)
– FeaturedAboveGrid (featured articles)
– Grid3Column (article grid)
– NewsletterInline (newsletter signup)
Step 6: For each section, admin configures properties:
– Content source: dynamic
– Category filter: current section's category
– Max items: 9
– Sort order: newest first
Step 7: Admin clicks "Preview" to verify layout
Step 8: Admin clicks "Publish"
Step 9: System syncs to Strapi + triggers revalidation
Step 10: Reader Portal shows updated section page layout

Flow 2: Customize and Test an Email Template

Step 1: Admin navigates to Settings > Email Templates
Step 2: Locates "Subscription Welcome" template
Step 3: Opens the editor
Step 4: Modifies subject line: "Welcome to `{{site_name}}`, `{{userName}}`!"
Step 5: Edits HTML body with updated welcome message
Step 6: Clicks "Preview" – sees rendered email with branding
Step 7: Clicks "Send Test" → enters own email address
Step 8: Receives test email with [TEST] prefix
Step 9: Verifies styling, variables, branding in email client
Step 10: Clicks "Save" to persist changes
Step 11: Future welcome emails use the updated template

[Flow 3: Configure Email Branding](#)

- Step 1: Admin navigates to Settings > Email Templates > Preferences
- Step 2: Updates Header section:
 - Upload logo image URL
 - Set header background color
- Step 3: Updates Footer section:
 - Company name, address
 - Social media links (Twitter, Facebook, etc.)
 - Copyright text
- Step 4: Updates Unsubscribe section:
 - Grace period: 7 days
 - Require confirmation: Yes
 - Reason options: "Too many emails", "Not relevant", "Other"
- Step 5: Clicks "Save"
- Step 6: All future emails use updated branding

[Flow 4: Manage Site Appearance and Navigation](#)

- Step 1: Admin navigates to Settings > Appearance
- Step 2: Adjusts color theme (primary color, accent color)
- Step 3: Selects font family for headings and body
- Step 4: Adjusts spacing (container width, section padding)
- Step 5: Previews changes in live preview panel
- Step 6: Clicks "Save" → Strapi updated → Reader Portal revalidated

- Step 7: Admin navigates to Settings > Navigation
- Step 8: Adds "Programs" link to header navigation
- Step 9: Reorders footer columns
- Step 10: Adds Instagram social link
- Step 11: Clicks "Save" → Changes appear on Reader Portal

13. QA Scenarios & Test Matrix

13.1 Page Template System — Test Scenarios

Setup Validation

#	Scenario	Precondition	Steps	Expected Result
T-PT-001	Seed page categories	Empty DB	POST /api/page-categories/seed	21 categories created
T-PT-002	Seed page templates	Categories seeded	POST /api/templates/seed	80+ templates created
T-PT-003	Verify Strapi content types	Strapi running	Check Strapi admin for required content types	All 7 content types present
T-PT-004	Verify env variables	Fresh deployment	Check STRAPI_URL, STRAPI_API_TOKEN, INTERNAL_API_TOKEN	All set and valid

Template CRUD (Happy Path)

#	Scenario	Steps	Expected Result
T-PT-010	List templates	GET /api/templates	Returns all templates (DB + built-in)
T-PT-011	Filter by category	GET /api/templates?category=homepage	Only homepage templates returned
T-PT-012	Create custom template	POST /api/templates with name, sections	Template created with <code>draft</code> status
T-PT-013	Edit template	PUT /api/templates with updated sections	Template updated, minor version incremented
T-PT-014	Delete custom template	DELETE /api/templates?id=xxx	Template deleted
T-PT-015	Cannot delete built-in	DELETE /api/templates?id=editorial-classic	Error: cannot delete built-in template

Template Versioning

#	Scenario	Steps	Expected Result
T-PT-020	Version created on edit	Edit template → Save	New TemplateVersion record created
T-PT-021	View version history	GET /api/templates/[id]/versions	List of all versions with change notes
T-PT-022	Rollback to previous	Select old version → Rollback	New version created with old content
T-PT-023	Seed-on-write for built-in	Edit a built-in template	Template copied to DB, then edited

Template Publishing

#	Scenario	Steps	Expected Result
T-PT-030	Publish template	POST /api/templates/[id]/publish	Status=active, synced to Strapi
T-PT-031	Verify Strapi sync	Publish → Check Strapi admin	page-configuration document updated
T-PT-032	Verify revalidation	Publish → Check Reader Portal	Updated layout visible
T-PT-033	Publish requires permission	User without homepage:update	403 Forbidden

Canvas Editor UI

#	Scenario	Steps	Expected Result
T-PT-040	Drag-drop section	Drag hero section to canvas	Section added at drop position
T-PT-041	Configure section properties	Select section → Change content source	Properties saved
T-PT-042	Reorder sections	Drag section to new position	Order updated
T-PT-043	Remove section	Select → Delete	Section removed from canvas
T-PT-044	Live preview	Edit sections → Check preview	Preview reflects changes in real-time
T-PT-045	Section palette filtering	Select page category	Only compatible sections shown

Reader Portal Rendering

#	Scenario	Steps	Expected Result
T-PT-050	Homepage renders with template	Publish homepage template → Visit reader portal /	Sections render as configured
T-PT-051	Section page renders	Publish section template → Visit /fiction	Section page shows template layout
T-PT-052	Article page renders	Publish article template → Visit /article/[slug]	Article page shows template layout
T-PT-053	Dynamic content loads	Section with dynamic source → Check reader	Articles fetched from Strapi and displayed
T-PT-054	Manual content loads	Section with manual IDs → Check reader	Specific articles displayed in order
T-PT-055	Banner scheduling	Create banner with future startsAt	Banner not visible before start time
T-PT-056	Banner scheduling (active)	Create banner with past startsAt, future endsAt	Banner visible
T-PT-057	Theme applied	Modify appearance settings → Check reader	CSS variables updated, styling changes
T-PT-058	Navigation dynamic	Update nav in admin → Check reader header/footer	Links updated
T-PT-059	Static page renders	Create static page → Visit /[slug] on reader	Page content displayed
T-PT-060	Personalized section	Section with personalizationEnabled → Logged-in reader with prefs	Content boosted by preferences

Page Categories

#	Scenario	Steps	Expected Result
T-PT-070	List categories	GET /api/page-categories	21 categories returned
T-PT-071	Get category templates	GET /api/page-categories/[slug]	Category with associated templates
T-PT-072	Category CRUD	Create/edit/delete custom category	Category managed

Static Pages

#	Scenario	Steps	Expected Result
T-PT-080	Create static page	POST with title, slug, content	Page created
T-PT-081	Edit static page	PUT with updated content	Page updated
T-PT-082	Delete static page	DELETE by slug	Page removed
T-PT-083	Unique slug validation	Create page with existing slug	Error: slug already exists
T-PT-084	Active toggle	Set isActive=false	Page not rendered on reader portal

13.2 Email Template System — Test Scenarios

Setup Validation

#	Scenario	Precondition	Steps	Expected Result
T-ET-001	Seed email templates	Empty DB	POST /api/settings/email-templates/seed	20 templates created
T-ET-002	Seed is idempotent	Templates exist	Run seed again	No duplicates; existing templates unchanged
T-ET-003	SMTP configured	SMTP credentials set	Check SMTP status endpoint	Connection successful
T-ET-004	Verify EMAIL_ENABLED	Env var set	Check system settings	Email sending enabled

Email Template CRUD

#	Scenario	Steps	Expected Result
T-ET-010	List templates	GET /api/settings/email-templates	All templates returned with categories
T-ET-011	Create template	POST with key, name, subject, htmlContent	Template created
T-ET-012	Edit template	PUT with updated subject/body	Template updated
T-ET-013	Delete template	DELETE by ID	Template removed
T-ET-014	Unique key validation	Create with existing key	Error: key already exists
T-ET-015	Requires SETTINGS_UPDATE	User without permission	403 Forbidden

Variable Replacement

#	Scenario	Steps	Expected Result
T-ET-020	Basic variable	Template with <code>{{userName}}</code> → send with <code>userName="John"</code>	"John" in rendered email
T-ET-021	Case-insensitive	Template with <code>{{UserName}}</code> → send with <code>username="Jane"</code>	"Jane" in rendered email
T-ET-022	System variables	Template with <code>{{site_name}}</code> → send	Site name appears
T-ET-023	CRM variables	Template with <code>{{support_email}}</code> → send	CRM support email appears
T-ET-024	Missing variable	Template with <code>{{unknown}}</code> → send	<code>{{unknown}}</code> left as-is or empty
T-ET-025	Caller overrides system	Send with <code>site_name="Custom"</code>	"Custom" used instead of system value

Email Preview & Test Send

#	Scenario	Steps	Expected Result
T-ET-030	Preview template	Click Preview in editor	Rendered HTML with branding wrapper shown
T-ET-031	Send test email	Click Send Test → enter email	Email received with [TEST] prefix
T-ET-032	Test email logged	Send test → check Email Logs	EmailLog entry created

Email Branding & Preferences

#	Scenario	Steps	Expected Result
T-ET-040	Update branding	Change logo, colors → Save	Future emails use new branding
T-ET-041	Update footer	Change company name, social links → Save	Footer updated in all emails
T-ET-042	Custom footer HTML sanitized	Enter <code><script>alert('xss')</script></code>	Script tag stripped
T-ET-043	Unsubscribe settings	Set grace period, require confirmation	Unsubscribe flow respects settings

Email Delivery & Logging

#	Scenario	Steps	Expected Result
T-ET-050	Successful delivery	Trigger subscription welcome email	EmailLog: status=sent, provider set
T-ET-051	Failed delivery	Send to invalid SMTP config	EmailLog: status=failed, errorMessage set
T-ET-052	OTP email logged	Trigger OTP email	EmailLog entry with <code>template_key=otp_verification</code>
T-ET-053	Subscription emails logged	Trigger renewal reminder	EmailLog entry with correct template key
T-ET-054	View email logs	Navigate to Email Logs page	List of all sent emails with filters

Email Compliance

#	Scenario	Steps	Expected Result
T-ET-060	Marketing email has unsubscribe	Send marketing email	Unsubscribe link in footer + List-Unsubscribe header
T-ET-061	Transactional no unsubscribe	Send OTP email	No unsubscribe link
T-ET-062	Lifecycle has unsubscribe	Send welcome email	Unsubscribe link present
T-ET-063	HMAC unsubscribe token valid	Click unsubscribe link	Token validates, unsubscribe processes
T-ET-064	Unsubscribed contact suppressed	Unsubscribe reader → Send campaign	Unsubscribed contact not in audience
T-ET-065	Grace period enforced	Set 7-day grace → Unsubscribe	Pending unsubscribe stored, processed after 7 days
T-ET-066	Confirmation required	Enable confirmation → Unsubscribe	Confirmation page shown before processing
T-ET-067	Reason captured	Enable reasons → Unsubscribe with reason	Reason stored in audit log
T-ET-068	CAN-SPAM address	Check marketing email	Physical address in footer (from EmailPreferences)

Email Campaigns

#	Scenario	Steps	Expected Result
T-ET-070	Create campaign	Marketing > Email Campaigns → Create	Campaign created
T-ET-071	Preview campaign	Select template → Preview	Rendered campaign email shown
T-ET-072	Send campaign	Configure audience → Send	Emails sent to audience segment
T-ET-073	Campaign logged	Send campaign	EmailLog entries for each recipient
T-ET-074	Unsubscribed excluded	Audience with unsubscribed contacts	Unsubscribed contacts not included
T-ET-075	Duplicate campaign	Click Duplicate	New campaign created with same settings

13.3 Cross-Module Integration Tests

#	Scenario	Modules Involved	Steps	Expected Result
T-XM-001	Template → Strapi → Reader	Templates, Strapi, Reader Portal	Publish template → Verify Strapi → Verify Reader	Full pipeline works
T-XM-002	Settings → Reader Portal	Settings, Reader Portal	Change appearance → Check reader	Theme updates
T-XM-003	Email template → Subscription	Email, Subscription	New subscriber → Welcome email	Email uses DB template with branding
T-XM-004	Feature flag → Section rendering	Settings, Reader Portal	Disable newsletter_popup flag → Check reader	Newsletter section hidden
T-XM-005	Navigation → Reader Header/Footer	Navigation, Reader Portal	Update nav links → Check reader	Links updated
T-XM-006	Campaign → Email → Unsubscribe	Marketing, Email, Reader Portal	Send campaign → Click unsubscribe → Verify	Full unsubscribe flow works
T-XM-007	Internal token auth	Admin Console, Reader Portal	Reader Portal fetches settings via internal token	Settings returned correctly
T-XM-008	Paywall + template	Templates, Paywall	Article template with paywall gate → Non-subscriber	Paywall overlay shown

13.4 Permission / RBAC Tests

#	Scenario	Steps	Expected Result
T-RBAC-001	Template edit without permission	User without <code>homepage:update</code> → Edit template	403 Forbidden
T-RBAC-002	Email template edit without permission	User without <code>SETTINGS_UPDATE</code> → Edit email template	403 Forbidden
T-RBAC-003	Template read without auth	Unauthenticated → GET <code>/api/templates</code>	Templates returned (public read)
T-RBAC-004	Email template read requires auth	Unauthenticated → GET <code>/api/settings/email-templates</code>	401 Unauthorized
T-RBAC-005	Internal token for reader portal	Reader Portal with valid <code>INTERNAL_API_TOKEN</code>	Settings returned
T-RBAC-006	Invalid internal token	Reader Portal with wrong token	401 Unauthorized

13.5 Edge Cases & Negative Tests

#	Scenario	Steps	Expected Result
T-EDGE-001	Publish with Strapi down	Strapi offline → Publish template	Error message; template not marked active
T-EDGE-002	Empty template sections	Create template with no sections → Publish	Valid but renders empty page
T-EDGE-003	Invalid section variant	Template with non-existent variant	Graceful fallback (section skipped)
T-EDGE-004	Circular content source	Section referencing itself	No infinite loop; empty or error
T-EDGE-005	Email send with no SMTP	EMAIL_ENABLED=false → Trigger email	Email not sent; logged appropriately
T-EDGE-006	Very long email body	Template with 100KB+ HTML	Email sends successfully
T-EDGE-007	Special characters in variables	Variable value with <code><script></code> tag	HTML escaped in output
T-EDGE-008	Concurrent template edits	Two admins edit same template	Last save wins; both create versions
T-EDGE-009	Missing Strapi content	Reader Portal with empty Strapi	Graceful fallback (default layout or error page)
T-EDGE-010	Template with hidden sections	All sections isVisible=false	Empty page body renders

14. Troubleshooting & Common Issues

Page Template Issues

Symptom	Possible Cause	Resolution
Template changes not visible on Reader Portal	Revalidation not triggered	Re-publish template; verify <code>READER_PORTAL_REVALIDATION_SECRET</code> matches
"Cannot sync to Strapi" error	Strapi down or token invalid	Verify Strapi is running; check <code>STRAPI_URL</code> and <code>STRAPI_API_TOKEN</code>
Empty page on Reader Portal	No page-configuration in Strapi for this page type	Publish the template for this page type
Section shows "No content"	Content source misconfigured or Strapi empty	Check section properties; verify content exists in Strapi
Theme changes not reflected	Browser cache	Hard refresh (Ctrl+Shift+R); verify Strapi <code>site-appearance</code> updated
Template gallery empty	Seed not run	Run <code>POST /api/templates/seed</code> and <code>POST /api/page-categories/seed</code>
Version history empty	Template never edited (built-in)	First edit creates initial DB record and version
Drag-drop not working	Browser compatibility	Use Chrome/Edge; clear cache

Email Template Issues

Symptom	Possible Cause	Resolution
Emails not sending	<code>EMAIL_ENABLED=false</code> or SMTP misconfigured	Check env var; test SMTP connection in Email Config page
Email styling broken	Custom footer HTML has invalid markup	Check Preferences > Footer; simplify HTML
Variables not replaced	Wrong variable name or missing from caller	Check template <code>variables[]</code> list; verify caller passes values
Test email not received	Spam filter or invalid email	Check spam folder; verify recipient address; check Email Logs
"No email template found" in logs	Template key mismatch	Verify template <code>key</code> matches the expected key; run seed
Unsubscribe link not working	<code>READER_JWT_SECRET</code> not set	Set env var for HMAC token generation
Branding not applied	EmailPreferences record missing	Navigate to Preferences page and save (creates default record)
Provider change not taking effect	Cached SMTP transporter	Restart admin console server
Campaign sends to wrong audience	Segment filter incorrect	Review segment query; verify <code>unsubscribedAt: null</code> filter
Email logs show "failed"	SMTP credentials expired or quota exceeded	Check error message; verify provider credentials

15. Known Limitations & Current Gaps

Page Template System

Item	Description	Severity	Workaround
No A/B testing	Cannot test multiple template variants against each other	Low	Manually switch templates and compare analytics
Single editor at a time	No collaborative editing or locking	Medium	Coordinate edits via team communication
ContentTypeBadge duplication	Shared component not yet extracted for all sections	Low	Non-blocking; cosmetic code cleanup
Inline SVG icon duplication	Same SVG icons duplicated across 50+ section files	Low	Non-blocking; candidate for shared icon utility
No template import/export	Cannot transfer templates between environments	Medium	Manual database migration

Email Template System

Item	Description	Severity	Workaround
Email config cache without invalidation	Provider config changes may require server restart	Medium	Restart admin console after changing SMTP config
No email rate limiting	Bulk sends process sequentially without throttling	Medium	For large subscriber bases, monitor delivery rates
No drag-drop email editor	Email editing is HTML-based, not visual drag-drop	Low	Use external HTML email builder, paste HTML
VAPID keys must be generated manually	Push notification keys not auto-generated	Low	Run <code>npx web-push generate-vapid-keys</code> during setup
Self-hosted social icons pending	Email footer social icons from external CDN (img.icons8.com)	Low	Replace with self-hosted assets in production
Institution inquiry templates hardcoded	CRM email drafts in route file, not in DB	Low	Intentional design — used as pre-written drafts for sales staff
Integration deep wiring partial	Some integration settings not fully connected to SDK initialization	Medium	Configure directly via env vars

Follow-Up Recommendations

- Email config hot-reload** — Call `resetTransporter()` from the email config update API to avoid restart requirement
- Email rate limiting** — Add queue system (e.g., Bull/BullMQ) for large-scale campaign sends
- Template import/export** — Add JSON export/import endpoints for template portability
- Collaborative editing** — Add optimistic locking or WebSocket-based real-time collaboration
- Self-host social icons** — Replace img.icons8.com CDN with `/public/icons/` assets

16. Appendix

A. Page/API Reference

Page Template APIs

Method	Endpoint	Permission	Purpose
GET	/api/templates	None (public)	List/filter templates
POST	/api/templates	homepage:update	Create template
PUT	/api/templates	homepage:update	Update template
DELETE	/api/templates	homepage:update	Delete template
GET	/api/templates/[id]/versions	homepage:read	Version history
POST	/api/templates/[id]/publish	homepage:update	Publish template
POST	/api/templates/seed	homepage:update	Seed built-in templates
GET	/api/page-categories	None	List categories
GET	/api/page-categories/[slug]	None	Category detail
POST	/api/page-categories/seed	homepage:update	Seed categories
GET/POST	/api/page-configurations	homepage:read/update	Page config CRUD
PUT	/api/page-configurations/[docId]	homepage:update	Update page config
POST	/api/page-configurations/[docId]/publish	homepage:update	Publish page config
GET/POST/PUT/DELETE	/api/static-pages/*	homepage:update	Static page CRUD
GET/PUT	/api/settings/navigation	NAVIGATION_READ/UPDATE	Navigation config
GET/PUT	/api/settings/appearance	NAVIGATION_READ/UPDATE	Appearance config
POST	/api/preview/session	Auth required	Create preview session

Email Template APIs

Method	Endpoint	Permission	Purpose
GET	/api/settings/email-templates	SETTINGS_READ	List templates
POST	/api/settings/email-templates	SETTINGS_UPDATE	Create template
PUT	/api/settings/email-templates/[id]	SETTINGS_UPDATE	Update template
DELETE	/api/settings/email-templates/[id]	SETTINGS_UPDATE	Delete template
POST	/api/settings/email-templates/[id]/test	SETTINGS_UPDATE	Send test email
POST	/api/settings/email-templates/seed	SETTINGS_UPDATE	Seed default templates
GET/PUT	/api/settings/email-preferences	SETTINGS_READ/UPDATE	Branding/preferences
GET	/api/settings/smtp/status	SETTINGS_READ	SMTP health check

[B. Template Status Enum](#)

Status	Description
draft	Template is being edited, not published
active	Template is published and in use
inactive	Template is deactivated but preserved
archived	Template is archived (built-in protection)

[C. Email Status Enum](#)

Status	Description
pending	Email queued for sending
sent	Email successfully delivered
failed	Email delivery failed
bounced	Email bounced (hard/soft)

D. 21 Built-In Page Categories

#	Slug	Label	Templates
1	homepage	Homepage	Editorial Classic, Magazine Grid, Featured Hero, Minimal Reader
2	article	Article	Standard, Immersive, Minimal
3	section	Section	Grid, List, Magazine
4	index	Index	Grid, List, Cards
5	author	Author	Profile, Minimal
6	subscription	Subscription	Pricing Cards, Pricing Comparison
7	search	Search	Interactive
8	static	Static	Standard, Centered
9	campaign	Campaign	Multiple variants
10	tag	Tag	Tag-specific layouts
11	archive	Archive	Timeline, grid variants
12	magazine-issues	Magazine Issues	Issue viewer
13	shop	Shop	E-commerce layouts
14	event	Event	Event page layouts
15	podcast	Podcast	Podcast page layouts
16	collection	Collection	Curated collection layouts
17	institutional	Institutional	Institutional subscription
18	account	Account	Reader account layouts
19	newsletter-archive	Newsletter Archive	Newsletter browse
20	gift	Gift	Gift subscription
21	error	Error	404, 500 pages

E. 20 Seeded Email Template Keys

#	Key	Category	Description
1	subscription_welcome	Subscription	New subscriber welcome
2	subscription_renewal_reminder	Subscription	Renewal reminder
3	subscription_expiration	Subscription	Expiration notice
4	subscription_payment_receipt	Subscription	Payment receipt
5	future_readers_coupon	Subscription	Coupon/promo code
6	institutional_invite	Subscription	Institutional invitation
7	gift_subscription_activation	Subscription	Gift activation
8	campaign_submission_approved	Campaign	Submission approved
9	campaign_submission_rejected	Campaign	Submission rejected
10	campaign_invitation	Campaign	Campaign invitation
11	campaign_notify_new_campaign	Campaign	New campaign notification
12	new_article_submission	Notification	New submission alert
13	article_reassigned	Notification	Article reassigned
14	action_needed	Notification	Action required
15	author_response	Notification	Author responded
16	ready_for_review	Notification	Ready for review
17	publishing_approved	Notification	Publishing approved
18	change_requested	Notification	Changes requested
19	article_live	Notification	Article published
20	otp_verification	System	OTP code

[F. Glossary](#)

Term	Definition
Section	A modular, reusable layout block within a page template (e.g., hero, grid, list)
Section Variant	A specific visual implementation of a section type (e.g., HeroFullWidth, Grid3Column)
Template	A complete page layout composed of ordered sections with default styles
Page Configuration	A Strapi document binding a page type to a template with specific section configurations
Content Source	How a section gets its content: dynamic (query-based), manual (hand-picked), mixed, or static
ISR	Incremental Static Regeneration — Next.js feature for on-demand page cache refresh
HMAC	Hash-based Message Authentication Code — used for secure unsubscribe tokens
Email Class	Classification of email type determining compliance behavior (mandatory-transactional, lifecycle, operational, marketing)
Email Pipeline	The canonical path for rendering and sending emails: resolve template → replace variables → wrap branding → send → log
Branding Wrapper	HTML shell applied to all emails with header, footer, logo, and styling from EmailPreferences
VAPID	Voluntary Application Server Identification — web push notification authentication

End of User Manual

[< Social Media Management](#)

[Complete guide to connecting social accounts, creating...](#)

[Magazine Issue Management >](#)

[Complete guide to creating magazine issues, managing...](#)