

News Portal — Admin Manual

This manual covers everything built so far, in the order you'd normally use it. It's written for a newsroom admin, not a developer — no code, no jargon.

How to read the "Known issues" markers: anything marked  ****Known issue**** is a real, confirmed gap or bug — not a hypothetical. If a feature isn't mentioned here at all, it hasn't been built yet (the project is being built in phases; see the bottom of this manual for what's still to come).

This manual is kept up to date every time a phase is signed off. If something here doesn't match what you see on screen, that's a bug worth reporting, not something to work around.

Quick reference

Every admin screen, its exact address, and what it controls. Replace `localhost:8080` with your site's real address once it's live.

Screen	Address	What it controls
Dashboard	<code>/admin/dashboard</code>	Overview widgets and charts
Activity Log	<code>/admin/activity-log</code>	Who did what, and when
Articles list	<code>/admin/articles</code>	All news articles
Write/edit an article	<code>/admin/articles/create</code> , <code>/admin/articles/{id}/edit</code>	One article's content, media, workflow, SEO
Article revisions	<code>/admin/articles/{id}/revisions</code>	Past versions of one article
Categories	<code>/admin/categories</code>	The subject sections (राजनीति, खेलकुद, etc.)
Tags	<code>/admin/tags</code>	Free-form topic labels on articles
Pages	<code>/admin/pages</code>	Static pages (About, Contact, Privacy, etc.)
Media Library	<code>/admin/media</code>	Every uploaded image, video, audio file, and PDF
Watermark settings	<code>/admin/media/settings/watermark</code>	Automatic watermark on uploaded images
Homepage Builder	<code>/admin/homepage-sections</code>	What appears on the public homepage, and in what order
Menus	<code>/admin/menus</code>	The navigation bar (and other menu locations)
Comments	<code>/admin/comments</code>	Moderation queue for reader comments
Blocked commenters	<code>/admin/comments/blocked</code>	Who is currently blocked from commenting
Comment settings	<code>/admin/comments/settings</code>	Guest commenting on/off, approval-required on/off
Polls	<code>/admin/polls</code>	Reader polls, their options, and live results
Newsletter subscribers	<code>/admin/newsletter/subscribers</code>	Everyone who has subscribed, and CSV export
Newsletter campaigns	<code>/admin/newsletter/campaigns</code>	Compose and send newsletter emails
Contact inbox	<code>/admin/contact-submissions</code>	Messages sent through the public contact form
Breaking News	<code>/admin/breaking-news</code>	The live homepage ticker's items — create/expire/order, pushed to readers in real time
Live Blogs	<code>/admin/live-blogs</code>	Live update streams attached to articles, pushed to readers in real time
Push Notifications	<code>/admin/push-notifications</code>	Compose and send browser push notifications to subscribed readers
Advertisements	<code>/admin/advertisements</code>	Ad slots across the site — image/HTML/AdSense, scheduling, device targeting, rotation, performance stats
Redirects	<code>/admin/seo/redirects</code>	301/302 old-path → new-path redirects
SEO Settings	<code>/admin/seo/settings</code>	Organization/site identity, robots.txt, social links, site verification codes
SEO Analyzer	<code>/admin/seo/analyzer</code>	A health-check report for published articles' title/description length and image alt-text coverage
Your profile	<code>/profile</code>	Your name, email, password, avatar, 2FA

Logging in

Go to `/login` and sign in with the email and password a developer/admin gave you. If your account has two-factor authentication (2FA) turned on, you'll be asked for a 6-digit code (or a recovery code) at `/login` → automatically redirected to `/two-factor-challenge`.

You manage your own account at `/profile`: name, email, phone, bio, language, timezone, avatar photo, password change, turning 2FA on/off, and "log out other devices" (ends every other browser session you're logged into, without changing your password).

⚠ **Known issue:** there is currently no admin screen to create or edit *other people's* user accounts (no "Users" page in the sidebar). New staff accounts (Reporters, Editors, etc.) must be created directly by a developer right now. If you need a new staff account, ask your developer rather than looking for a button — it isn't there yet.

Dashboard — `/admin/dashboard`

The first thing you see after logging in. Purely informational — nothing here is editable, it's a summary of the rest of the system.

Widgets (top of the page): total articles and a breakdown by status (draft / published / scheduled), total categories, total tags, pending comments awaiting moderation, total users, currently-trending articles (most-viewed lately), any active breaking-news items, and a feed of recent activity across the whole system (who published what, who edited a category, etc. — same data as the Activity Log, just a short recent slice).

Charts: daily visitor counts (last 14 days), articles published per month (last 6 months), how articles are distributed across categories, and user account growth over time.

Numbers refresh automatically about once a minute — you don't need to reload the page to get fresh totals, though a manual refresh always shows the latest.

Activity Log — `/admin/activity-log`

A running, paginated list of significant actions across the system — article status changes, category edits, media deletions, and so on — each with who did it and when. Read-only; there's nothing to configure here. Use it if you need to answer "who published this?" or "who deleted that category?".

Articles

This is the core of the CMS. An article moves through a review workflow before it's visible to readers (unless you have publish rights, in which case you can skip straight to publishing — see the workflow table below).

Writing an article

Go to `/admin/articles/create` (or **Articles** → "New article" from the articles list). The form is in sections:

- **Content** — Title (in Nepali), a Slug field (auto-generated from the title using a Nepali→Latin transliteration; edit it yourself if the auto-generated version reads awkwardly), a short Summary, and the main Content editor.
- **Media** (picked from the Media Library, not uploaded fresh into the article) — Featured image, a Gallery (multiple images), an Audio file, a PDF attachment, and a plain "Video / YouTube embed URL" text field (separate from anything you embed inline in the content editor). Click "**Choose from library**" to open a picker; inside that picker you can also **upload a brand-new file on the spot** (drag-and-drop or browse) — it lands in the Media Library automatically, exactly as if you'd uploaded it from `/admin/media` first.
- **Taxonomy** — one Category (required) and any number of Tags (check the boxes).
- **Attribution** — the Reporter/Author (only editable by roles with full edit rights — Reporters/Journalists always see their own name here), an optional Editor credit, and an optional Source name + Source URL (for wire-service or reprinted content).
- **Flags** — see the dedicated **Flags** section below.
- **Settings** — "Allow comments" (on by default; comment display itself isn't built yet — see Known issues at the end of this manual).
- **Related articles** — manually check other articles to feature as "related" on the public article page. If you don't check any, the public site automatically shows related articles based on shared tags and category instead — you only need this section if you want to override that automatic choice.
- **SEO** — SEO title, meta description, canonical URL, Open Graph title/description/image override, and Twitter card type. All optional — if left blank, the public page falls back to the article's own title/summary/featured image.

The content editor (CKEditor)

The toolbar supports: headings, bold/italic, bulleted and numbered lists, block quotes, links, tables, and inline images (uploaded the same way as the media picker — they land in the Media Library too). To add a caption under an inline image: click the image after inserting it, and a small toolbar appears above it with a caption-toggle button — click it and type the caption text.

To embed a YouTube or Vimeo video **inside the article body** (as opposed to the separate "Video / YouTube embed URL" field mentioned above): paste the video's URL using the toolbar's media-embed button. Only YouTube and Vimeo URLs are accepted for embeds — any other domain is silently removed when the article is saved, as a security measure.

Publishing workflow — who does what

An article moves through these states in order. Roles with full publish rights (Managing Editor, Admin, Super Admin) can jump straight to **Publish now** from any state before Published — the button always appears for them once the article exists, they don't have to click through every step.

Status	What it means	Who can move it forward
Draft	Being written	Reporter/Journalist: Submit for review . Anyone with publish rights: Publish now directly.
Submitted for Review	Writer has asked for review	Editor or above: Start review
In Review	An editor is looking at it	Editor or above: Approve , or Request changes (with a note explaining what to fix)
Changes Requested	Sent back to the writer	Reporter/Journalist can edit it again and Submit for review again
Approved	Editor signed off	Managing Editor/Admin: Publish now , or Schedule... for a future date/time
Scheduled	Will publish automatically	Publishes itself at the chosen time — a background check runs every minute, so it may appear up to ~60 seconds after the exact time you picked
Published	Live on the public site	Managing Editor/Admin: Archive
Archived	Pulled from the workflow	(End state — no further transitions)

All of this happens on the article's own page, `/admin/articles/{id}` (open an article from the list, or click **Preview** while editing it) — the buttons for whichever actions are currently legal appear at the top, under the status badge. A Reporter/Journalist can only edit **their own** articles, and only while those articles are in Draft or Changes Requested — once submitted, it's out of their hands until an editor sends it back.

Flags

Seven independent yes/no switches on every article, toggled from buttons on the article's own page (`/admin/articles/{id}`) or checkboxes while editing. They're independent of the workflow status above — an article can be Published and also Featured and also Trending, all at once.

Flag	Who can toggle it	What it does on the public site
Featured	Editor or above	Eligible for the homepage's Hero Slider / Featured section
Breaking	Managing Editor/Admin only	Shows a "Breaking" badge on the article's own page. Does not drive the homepage ticker — since Phase 11, the ticker is a separate, dedicated Breaking News manager (<code>/admin/breaking-news</code>); see that section below
Trending	Editor or above	Eligible for a Trending homepage section
Popular	Editor or above	(No dedicated homepage section reads this flag yet — see Known issues)
Editor's pick	Editor or above	Eligible for the homepage's Editor's Pick section
Exclusive	Editor or above	Stored on the article; nothing currently reads this flag on the public site
Live	Editor or above	Shows a "LIVE" badge on the article's own page. The badge also shows automatically (independent of this flag) whenever a real Live Blog attached to the article is currently Live — see Live Blogs below

⚠ **Known issue:** "Popular" and "Exclusive" flags can be set, but nothing on the public site currently uses them for anything — the homepage's "Most Viewed / Popular" section (see Homepage Builder below) actually ranks by real view counts, not this flag.

Scheduling

From an Approved article, click **Schedule...**, pick a date and time — a live preview shows the equivalent Bikram Sambat (BS) date next to the picker as you type, since the site displays dates to readers in BS, not AD. The article publishes itself automatically at that time; you don't need to be logged in or do anything else.

Revisions

Every save creates a snapshot. From an article's edit page, click **Revisions** (or go straight to `/admin/articles/{id}/revisions`) to see the full history with a line-by-line diff of what changed. Click **Restore** on any past version to bring it back — this doesn't delete history, it just adds a *new* revision on top with the old content, so restoring is itself undoable.

Where articles appear on the public site

A published article (status = Published, and its scheduled time, if any, has passed) appears at `/news/{its-slug}` , plus wherever the Homepage Builder or Menu Builder are configured to surface it (see those sections), plus its category archive page and any tag archive pages for tags it carries.

Categories — </admin/categories>

Categories are the site's subject sections (राजनीति, खेलकुद, प्रविधि, etc.) and can be nested to unlimited depth (a category can have sub-categories, which can have their own sub-categories, and so on).

To reorder or re-nest: drag a category by the `☰` handle on its left. Dropping it under another category makes it a child of that category; dropping it back to the top level makes it top-level again. This saves automatically the moment you drop it — no separate save button.

Fields when adding/editing one (</admin/categories/create> or </admin/categories/{id}/edit>): Name, Parent category (optional), a longer Description, an Icon (any emoji or short text), a Color (used as a small color dot next to the name in lists), a Sort order number, an uploaded Image, an Active toggle (inactive categories are hidden from the public site entirely), and SEO fields (title, description, Open Graph image).

Where categories appear on the public site

Each category has its own archive page at </category/{its-slug}> (paginated list of its published articles), and every top-level active category appears in the header navigation automatically — unless you've set up a Menu Builder menu for the Header location, in which case that menu takes over the nav bar instead (see Menu Builder below).

Tags — </admin/tags>

Free-form topic labels, not nested (flat list). Add/edit at </admin/tags/create> / </admin/tags/{id}/edit> — just a Name and a Language field.

Merging duplicates: if you end up with two tags that mean the same thing (e.g. a typo created a duplicate), check two or more tags in the list, pick which one should survive using the radio button next to each checked tag, and confirm. Every article on the tags that don't survive gets re-pointed to the one that does, then the others are deleted.

Where tags appear on the public site

Each tag has an archive page at </tag/{its-slug}> listing every published article carrying that tag. Tags also appear as clickable labels on individual article pages.

Pages — </admin/pages>

Static, one-off pages that aren't news articles — About Us, Contact, Privacy Policy, Terms, Team, Advertise, Careers, or a fully Custom page. Add/edit at </admin/pages/create> / </admin/pages/{id}/edit>: Title, a Template choice (the list above — this is just a label, it doesn't change the page's layout), a Language, Content (same rich-text editor as articles), an Active toggle, and SEO fields.

Where pages appear on the public site

Each page is at </page/{its-slug}>. If you create pages using the About/Contact/Privacy/Terms templates specifically, they also automatically appear as footer links — *unless* a Menu Builder menu has been set up for the Footer location, in which case that menu's links are used instead (see Menu Builder below).

Media Library — </admin/media>

The central place for every image, video, audio file, and PDF used anywhere on the site — articles pick their featured image/gallery/audio/ PDF from here rather than uploading fresh copies each time, and inline images typed into any rich-text editor land here too.

Uploading: drag files onto the dashed drop zone, or click to browse. You can select multiple files at once. After upload, images are automatically converted into three optimized sizes (thumbnail, medium, large, all as WebP) in the background — this can take a few seconds, so a just-uploaded image's thumbnail may briefly still look like it's loading.

Folders: the left sidebar shows a folder tree (nested, like Categories). Create a folder, then move files into it via each file's own edit panel.

Editing one file: click any item to open its panel. You can set Alt text (for accessibility/SEO — write it in Nepali, matching the rest of the site), a Caption (shown under the image wherever it's the featured image of a published article), a Credit line (e.g. "Photo: श्याम श्रेष्ठ" — shown alongside the caption), move it to a different folder, or delete it.

Crop / resize: from the same panel, use the built-in crop tool to crop an image to any rectangle, or the resize tool to fit it within a specific width/height box (never stretches or crops — just scales down to fit). Either action regenerates the three optimized sizes automatically.

"Used in" tracking: the same panel shows every place a file is currently used (which article's featured image, which page, etc.) so you can tell if it's safe to delete.

⚠ **Known issue:** there's no bulk-select — every action (delete, move to a folder) is one file at a time. If you need to clean up a lot of files, budget the time for it.

Watermark settings — </admin/media/settings/watermark>

Controls an automatic watermark stamped onto every image uploaded through the Media Library (not on images already stored directly on an article's older-style upload, if any exist — see Phase 5's design note in the project history if that distinction ever matters to you).

Restricted to Super Admin and Admin only — this is a real, deliberate restriction, not a bug: even Managing Editor, who otherwise has full control over Media, cannot see or open this screen (it's gated by a separate "settings" permission that only Super Admin/Admin hold). If a Managing Editor needs the watermark changed, a Super Admin or Admin has to do it.

Fields: Mode (None / Text / Logo image), the watermark Text (if Text mode) or a Logo image file to upload (if Image mode), Position (one of the four corners or center), and Opacity (0–100%).

Regenerate existing conversions: a button on the same page that reprocesses every already-uploaded file with your current watermark settings — use this after changing the watermark so older uploads get the new look too, not just new uploads.

⚠ **Known issues** (all real, all logged, none fixed yet):

- **Text-mode watermarks do not render Nepali/Devanagari text** — only Latin characters display correctly. If your watermark text needs to be in Nepali, it won't show up correctly. Use the Logo image mode instead until this is fixed.
- **No size/scale control** — the watermark's size isn't adjustable from this screen yet, and the current default renders quite small on large images.
- **No margin/offset control** — the watermark currently sits flush against the image's edge; there's no "distance from the border" setting yet.

These four watermark gaps are scheduled to be fixed together in one future pass — see PROGRESS.md in the project repository for the technical detail if you want it.

Homepage Builder — </admin/homepage-sections>

Controls what shows up on the public homepage (/) and in what order. Each row in the list is one **section**; the homepage renders them top-to-bottom exactly as ordered here.

To reorder: drag a row by its  handle. Saves automatically on drop.

To add a section: click "Add section", pick a Type (see the table below), an optional Title (shown as a heading above the section — leave blank to use the type's default name), a Source category (only required for the Category Block type), an Item count (how many articles it shows, 1–20), a Layout (Default/Compact/List — controls how big each article card is), and whether it's Active (inactive sections are kept configured but hidden from the public site — use this to temporarily pull a section without deleting its settings).

Each row shows a summary line (type · category · item count · layout) so you can see the whole homepage's shape at a glance without opening every section individually.

Section types and what they show

Type	What appears on the public homepage
Hero Slider	The most recent Featured articles, one large + up to 4 smaller alongside
Breaking News	A live-updating ticker of items from the dedicated Breaking News manager (/admin/breaking-news — not the article "Breaking" flag) — one headline at a time, holds ~5 seconds, fades to the next, pauses on hover. Updates instantly for readers who already have the page open when you add/remove an item, no refresh needed. A single breaking item shows statically with no animation.
Latest News	The most recent published articles, newest first
Trending	The most recent Trending-flagged articles
Featured	The most recent Featured-flagged articles (same flag as Hero Slider — most sites use one or the other, not both)
Editor's Pick	The most recent Editor's-pick-flagged articles
Category Block	The most recent articles from one specific category (you choose which)
Photo Gallery	A grid of the most recent articles that have at least one image in their Gallery (the multi-image field). Each tile shows that article's first gallery photo, falling back to its featured image only if it has no gallery photos
Video Gallery	A grid of the most recent articles that have a video URL set, with a play-button overlay. Each tile shows the article's featured image as the video thumbnail — make sure articles used here have one set, or the tile will be blank
Most Viewed / Popular	The most-viewed articles by real reader view count (not the "Popular" flag — see the Flags note above)
Poll	One specific poll's voting form (or live results, for a reader who's already voted) — see "Placing a poll on the homepage" below
Ad Block	One ad from the Homepage Block position, picked the same way as every other ad slot (weighted rotation, device targeting, schedule) — see Advertisement Manager below. No longer a placeholder as of Phase 12
Newsletter Signup	A real email-subscribe box (see the Newsletter section below) — no longer a placeholder as of Phase 10

If no ad is currently eligible for the Homepage Block position (none created yet, all inactive, or all outside their schedule/device match), an Ad Block section quietly renders nothing rather than an empty box — same "never show a stub" behavior as an empty Poll section.

Changes here appear on the public homepage immediately — there's no "publish" step and no waiting for a cache to expire; the moment you save a change, the next visitor to [/](#) sees it.

Placing a poll on the homepage (step by step)

A poll never appears on the homepage on its own — even a poll that isn't attached to any article — until you explicitly add a **Poll** section for it here. This is deliberate: it keeps the homepage's shape fully in your control, the same as every other section type.

1. Create the poll first, at </admin/polls> (see the Polls section below), if you haven't already.
2. Go to </admin/homepage-sections> and click "**Add section**".
3. Set Type to **Poll**. A new **Poll** dropdown appears — pick the poll you want to show.
4. Optionally set a Title (shown as a heading above the poll). Item count and Layout don't apply to a poll and can be left as-is.
5. Save. The poll now appears at this section's position in the homepage list — drag it by its  handle like any other section to move it.

If you delete the poll later (or the dropdown's selection is somehow left empty), this section quietly renders nothing rather than an empty box — the admin list's summary line for that row will say so ("no poll selected — renders nothing" or "poll deleted — section renders nothing") so it's obvious what to fix. An article-attached poll (set on the poll's own edit screen — see Polls below) is different: it renders automatically on that article's page, with no Homepage Builder section needed.

Placing the newsletter signup box on the homepage (step by step)

1. Go to </admin/homepage-sections> and click "**Add section**".
2. Set Type to **Newsletter Signup**.
3. Optionally set a Title (shown as a heading above the email field — leave blank for the default "न्यूज़लेटर सदस्यता" heading).
4. Save. The subscribe box now appears at this section's position — drag it by its  handle to move it, same as any other section.

There's nothing else to configure — the form itself always posts to the same subscription flow described under Newsletter below, regardless of where on the homepage you place it.

Controls the site's navigation — which links appear in the header, footer, and (once built) other locations, in what order, with optional nested sub-menus and "mega menu" preview panels.

Locations

Location	Where it actually renders on the public site
Header	The main navigation bar at the top of every page
Footer	The link row in the site footer
Mobile	The nav shown in the mobile hamburger menu — falls back to whatever the Header menu has if you haven't set up a Mobile-specific one
Sidebar	⚠️ No public page has a sidebar yet — see Known issues below

If you haven't created a menu for a location yet: Header falls back to a plain list of your top-level categories (so the nav bar is never empty); Footer falls back to About/Contact/Privacy/Terms page links (if those pages exist); Mobile falls back to whatever Header is showing. Nothing breaks by leaving a location unconfigured — you're just getting the original, simpler behavior until you set one up.

⚠️ **Known issue — Sidebar has no public render surface:** you can create a menu and set its location to Sidebar, and the admin will let you build it out completely — but there's currently no sidebar anywhere on the public site to show it in. The dropdown says this directly when you're picking a location, so you don't discover it by trial and error.

Setting up the Header menu (step by step)

1. Go to </admin/menus> and click **"Add menu"**.
2. Give it a Name (e.g. "Main Header Menu") and set Location to **Header**.
Save.
3. You land on the menu's own page (</admin/menus/{id}>). Click **"Add item"**.
4. Fill in: a Label (the text readers see), a Link type (see the table below), the specific target (a category/page/article picker appears depending on the link type you chose, or a plain URL box for Custom), and whether it's a **Mega menu** item (see below). Save.
5. Repeat for each top-level link you want.
6. To nest an item under another (a simple dropdown, or the mega-menu preview panel): drag it by its  handle and drop it under the parent item.

Link types

Link type	Points to
Category	One of your Categories — links to that category's archive page
Page	One of your CMS Pages
Article	One specific Article
Custom URL	Any URL you type in yourself (internal or external)

Nesting and dropdowns

Any item with children (dragged under it) shows a small dropdown on hover in the header nav — even if it isn't a "mega menu" item. This is the plain, simple case: a parent link plus a short list of child links, nothing more.

Mega menus

Check **"Mega menu"** on a **top-level** item that's linked to a **Category**, and hovering over it on the public site shows a wider panel with:

- any children you've nested under it (as plain links), then
- that category's 4 most recent articles as a live preview (title only, always up to date — it's not a snapshot, it re-checks the category's latest articles on every page load).

Mega menu only does anything for top-level, Category-linked items — the checkbox has no visible effect on child items or on Page/Article/Custom links.

Where menus appear on the public site

Header and Footer menus render in the site's header and footer on every page (see the Locations table above for the exact fallback behavior when one isn't configured). Changes save immediately and appear on the very next page load — no publish step, no cache delay.

Comments

Readers can leave comments on any article that has "Allow comments" switched on (see the article form's Settings section). Comments can be nested (a reader can reply directly to another comment), and guests (anyone not logged in) can comment too, unless you turn that off.

Moderation queue — </admin/comments>

Every comment appears here, filterable by tab (All / Pending / Approved / Rejected / Spam) — the count next to each tab shows how many are in that state. For each comment you can:

- **Approve** — makes it visible on the public article page.
- **Reject** — hides it, but keeps the record (visible again if you later Approve it).
- **Mark spam** — same as Reject, but categorizes it separately so spam doesn't clutter your Rejected tab.
- **Reply** — post a reply as staff, right from this screen. Staff replies are always shown immediately (never held for approval), and nest under the comment you replied to, same as a reader's own reply would.
- **Block commenter** — stops this same person (by account, or by IP address if they commented as a guest) from commenting again anywhere on the site. Optionally note a reason. Their past comments are **not** deleted automatically — do that separately with Delete if you want them gone too.
- **Delete** — permanently removes the comment (and, since replies are nested under their parent, effectively hides any replies under it too).

Blocked commenters — </admin/comments/blocked>

Lists everyone currently blocked, who blocked them, and why (if a reason was given). Click **Unblock** to let them comment again.

Comment settings — </admin/comments/settings>

Two switches, **Super Admin/Admin only**:

- **Allow guest comments** — on by default. Turn off to require a logged-in account (any role, including Subscriber) to comment.
- **Require moderator approval before a comment is public** — on by default, meaning every new comment starts in the Pending queue and stays invisible to readers until you approve it. Turn off to let comments appear immediately and moderate after the fact instead. Staff replies (posted from the moderation queue) are always immediate either way.

Where comments appear on the public site

At the bottom of an article page (</news/{slug}>), below a "Reply" link on every comment for nesting, and a honeypot-protected submission form. A reader typing into the hidden honeypot field (invisible to humans, only a bot's automated form-filler would touch it) has their submission silently rejected — this and a rate limit (6 submissions per minute per visitor) are the site's spam defenses on top of moderation.

Polls — </admin/polls>

Multiple-choice polls with live results. Each poll has a question and two or more options; readers vote once (enforced by a long-lived browser cookie plus the IP address recorded alongside each vote — a real, but intentionally simple, "heuristic" the master spec itself calls for, not airtight against someone clearing cookies).

Creating/editing a poll (</admin/polls/create>, </admin/polls/{id}/edit>): a Question, an optional "Attach to article" (leave as Standalone if you plan to place it on the homepage instead — see below), optional Starts at/ Ends at times, an Active toggle, and the Options list — add or remove options freely with the "+ Add option"/"Remove" buttons. Renaming an existing option keeps its votes; removing an option deletes its votes along with it. A poll needs at least two options at all times.

Live results: the edit screen shows a real-time-as-of-page-load bar chart of vote counts and percentages, right below the option editor.

Where polls appear on the public site

- **Attached to an article:** renders automatically on that article's page (</news/{slug}>), below the share buttons — nothing further to configure.
- **Standalone** (not attached to any article): renders **nowhere** on its own. You must explicitly place it via a **Poll** section in the **Homepage Builder** — see "Placing a poll on the homepage" above for the exact steps. A standalone poll you haven't placed anywhere is simply not visible to readers yet, by design — the same as any other homepage section you haven't added.

A reader who has already voted sees the live results immediately instead of the voting form on their next visit to that poll (remembered via the same voter cookie).

Newsletter

Subscribers — `/admin/newsletter/subscribers`

Every email address that has ever subscribed, filterable by status (Pending confirmation / Confirmed / Unsubscribed). Only **Confirmed** subscribers receive campaign emails — this is double opt-in: subscribing only creates a Pending row and sends a confirmation email; the address only becomes Confirmed once that link is clicked. Click **Export CSV** to download the full list (email, status, confirmed date, subscribed date). **Remove** deletes a subscriber's record outright.

****There is no "Approve" button on this screen, by design, not a missing feature.**** Confirmation is entirely self-service — the reader themselves clicks the link in the confirmation email, which is what flips Pending → Confirmed. No admin action is needed or possible in between; if a subscriber never clicks their link, they simply stay Pending indefinitely (and never receive campaigns) until they either click it or subscribe again. If a reader says they subscribed but see nothing — see "Email delivery" just below; in local development the confirmation email doesn't reach a real inbox at all unless you've configured a real mail driver/SMTP.

Campaigns — `/admin/newsletter/campaigns`

Compose an email to send to every Confirmed subscriber. Fields: Subject, Body (same rich-text editor as articles/pages), and an optional "Send at" time.

- Leave "Send at" blank to save as a **Draft** — nothing is sent until you open it again and click **Send now**.
- Set a future "Send at" time to mark it **Scheduled** — it sends itself automatically at that time (a background check runs every minute, same mechanism as scheduled article publishing, so it may go out up to ~60 seconds after the exact time you picked).
- Sending is **queued**, not instant: clicking Send now (or a scheduled time arriving) hands the job to the background queue, which emails subscribers in the background — the page confirms "queued", not "sent", and the campaign's status becomes Sending, then Sent once every email has been queued for delivery.
- A sent (or currently sending) campaign can no longer be edited.
- Every sent email includes an unsubscribe link automatically — you never need to add one yourself in the body.

Where the signup form appears on the public site

The Homepage Builder's **Newsletter Signup** section type — see "Placing the newsletter signup box on the homepage" above for the exact steps. Subscribing there starts the same double-opt-in flow described above.

Email delivery (development vs. production)

Every feature that sends email — newsletter confirmation links, newsletter campaigns, and Contact Inbox replies — goes through the same one mail configuration. If one of them isn't reaching a real inbox, all three aren't, and the fix is the same for all three.

In local development, this project's default is the `log` mail driver (`MAIL_MAILER=log` in `.env`), which is a standard Laravel local-dev default, not a bug: instead of actually sending anywhere, every email gets written to `storage/logs/laravel.log` as its full rendered HTML. This is why a real confirmation/reply email you'd expect in your own inbox during development never arrives — it was never sent out, by design, until you configure a real mail driver (see below).

To see a dev email right now: open `storage/logs/laravel.log` (or `docker compose exec app tail -f storage/logs/laravel.log` to watch it live while you subscribe/reply) and search for the subject or a snippet of the body — the confirmation link itself appears as a real, clickable `href` in the logged HTML, e.g. ``http://localhost:8080/newsletter/confirm/{token}``; copy that URL into a browser to complete the opt-in loop exactly as a reader would. The same applies to a contact-form reply or a newsletter campaign send — all three write to this same log file when running under the `log` driver.

****There is no mail-catcher UI (like Mailpit/Mailhog) set up in this project yet**** — `docker-compose.yml` doesn't currently include one. If you'd like a proper inbox-style UI for local testing instead of reading the log file, that's a small, standard addition (one more Docker service) worth asking your developer for.

For production, `MAIL_MAILER=log` must be replaced with a real transactional provider before launch — Postmark, Amazon SES, Resend, or standard SMTP are all supported out of the box (see `config/services.php` and `config/mail.php` for the exact credentials each one needs). Until that's configured, newsletter confirmation, campaign sending, and contact replies will all silently "succeed" from the admin's point of view while never actually reaching a reader — there is no error shown, since writing to the log is a completely valid delivery as far as the app is concerned.

Contact Inbox — `/admin/contact-submissions`

Every message sent through the public contact form lands here, newest first, filterable by status (New / Read / Replied / Spam). Opening a New message automatically marks it Read. Fields shown: name, email, phone (if given), subject (if given), and the message body.

Replying: type your reply in the box on the message's own page and click **Send reply** — this emails your reply directly to the sender's address (queued, same as newsletter sending) and marks the message Replied, recording who replied and when. The reply text itself is only kept in the Activity Log, not as a separate stored field on the message. In local development this reply email doesn't reach a real inbox unless a real mail driver is configured — see "Email delivery" under Newsletter above for where it actually goes and how to see it.

Mark as spam / **Delete** are available from the same page.

Setting up the public contact page (step by step)

The contact form only appears on a CMS Page that uses the **Contact** template — it has no URL of its own. If you don't already have one:

1. Go to `/admin/pages` and click **"Add page"**.
2. Give it a Title (e.g. "सम्पर्क"), set Template to **Contact Us**, and
write whatever introductory Content you'd like above the form (contact details, office hours, a map embed, etc. — this part is entirely optional; the form still renders even with empty Content).
3. Make sure **Active** is checked, and Save.
4. The page is now live at `/page/{its-slug}` (the slug auto-generates from the title, same as any other page — check the Pages list for the exact URL, or click **Preview**). The contact form renders automatically below the page's content — nothing else to configure.
5. If you haven't set up a Footer menu in the Menu Builder, this page also
starts appearing as a "Contact us" link in the site footer automatically (the same About/Contact/Privacy/Terms fallback the Menu Builder section describes).

You can have at most one page usefully serving as *the* contact page in practice — if you create more than one Contact-template page, each one independently shows the form, which is rarely what you want.

Where the contact form appears on the public site

On any CMS Page using the **Contact** template (see the setup steps just above). The form itself (name, email, phone, subject, message) is honeypot-protected and rate-limited (5 submissions per minute per visitor), same spam defenses as comments.

Article Reactions (reader-facing, not an admin screen)

Every published article page shows five reaction buttons (👍 ❤️ 😊 😞 😡) below the share buttons, with live counts. A reader (logged in or not) can pick one reaction per article; clicking the same one again removes it, clicking a different one swaps it. Logged-in readers are tracked by account; guests are tracked by IP address. There is no admin screen for reactions — they're purely a reader-facing engagement signal, with counts visible directly on the article page.

Search (reader-facing, not an admin screen)

There's no separate "Search" admin screen — search is powered automatically by whatever's already published. As soon as an article is published, it becomes findable within seconds; the moment it's unpublished or archived, it drops back out. Nothing to configure per-article.

What readers see:

- A search icon in the site header (every public page) opens a search overlay: type and results appear as you type, filterable by category, author, and a date range, with your search terms highlighted in the results. Arrow keys move between results; Escape closes it.
- `/search` is the same thing as a normal page — useful for sharing a search link, or for anyone browsing without JavaScript. It has the same filters as the overlay, as normal dropdowns/date fields on the page.
- Search only ever considers *published* articles, matching on title, summary, full content, category, tags, and author name — a reader searching an author's name or a tag finds the right articles even if those words aren't in the headline itself.
- Typos and partial words are tolerated (e.g. searching "क्रिकेट" or "क्रिक" still finds "क्रिकेट" articles) — this is Meilisearch's own typo-tolerance and prefix search, tuned for Devanagari.
- Results favor the best textual match first; among similarly-relevant results, more recent articles are shown ahead of older ones.

For developers: after changing `config/scout.php`'s index settings, run `php artisan scout:sync-index-settings` to push the change to Meilisearch. If the search index and the database ever look out of sync (e.g. after restoring a database backup), `php artisan scout:import "App\Domains\News\Models\Article"` rebuilds the index from scratch — normal day-to-day publishing needs neither command, since indexing happens automatically via a queued job on every save.

⚠ **Known issue:** the search overlay and `/search` page only expose category/author/date as filter controls — there's no tag dropdown in the UI, even though tag content is fully searchable (searching a tag's name finds articles carrying it). A `?tag=slug` link narrows results by tag under the hood, but nothing currently generates that link for a reader to click.

Breaking News — `/admin/breaking-news`

Controls the homepage's Breaking News ticker (the red banner that rotates through headlines). This is a **separate, dedicated manager** from the "Breaking" flag on an individual article (see the Flags table above) — the ticker is driven entirely by what you add here, not by which articles happen to have that flag checked.

Adding an item (`/admin/breaking-news/create`): a **Headline**, optionally linked to one of your Articles (clicking the item takes readers to that article), or a Custom URL instead if it's not tied to a full article (a plain headline with no link at all is also allowed). Optional Starts at/Expires at times, and an Active toggle.

Auto-start and auto-expiry: an item with a future Starts at time stays off the ticker until that moment, then appears on its own — and one with an Expires at time deactivates and drops off automatically once it passes. Both are driven by the same background check running every minute, so either may take up to ~60 seconds after the exact time you picked to actually appear/disappear for readers who already have the homepage open. No manual cleanup needed either way.

Reordering: drag a row by its `⋮` handle on the list page — saves automatically on drop, same as Categories/Homepage Builder.

Also send as a push notification: if you also have push-notification permissions, the "Add breaking item" form has a checkbox to immediately send this headline as a push notification to every subscribed reader when you create it — see Push Notifications below.

Where breaking news appears on the public site

Only on the homepage, and only if a **Breaking News** section has been added via the **Homepage Builder** — same "you must explicitly place it" rule the Poll section type follows. Once placed, the ticker ****updates live for readers who already have the homepage open**** — a new item you add, or an item that auto-expires, appears/disappears on their screen without them refreshing the page (pushed over a live connection). The article's own "Breaking" flag/badge (shown on the article page itself) is unrelated and unaffected by anything here.

Live Blogs — `/admin/live-blogs`

A live, running update stream attached to one article — for covering an election count, a sports match, a breaking event, anything that unfolds over hours rather than being written once.

Creating one (`/admin/live-blogs/create`): pick the Article it belongs to and give the live blog its own Title. It starts in **Scheduled** status — nothing shows on the public article page yet.

Managing one (`/admin/live-blogs/{id}`, or click **Manage** from the list): click **Go live** when you're ready to start — this is what makes the update stream and its **LIVE** badge appear on the public article page. While Live, a "Post an update" box lets you publish short updates one at a time; each appears instantly on the public page for anyone already reading the article, with no refresh needed on their end. Click **Pin** on any update to keep it at the top of the list regardless of when it was posted (useful for a key fact you want readers to see first); click **Unpin** to return it to normal newest-first order. **Delete** removes an update entirely. Click **End live blog** when it's over — the LIVE badge disappears and no further updates can be posted (Go live again if you need to resume covering the same event later).

⚠ **Known limitation:** updates can only be posted while the live blog's status is Live — there's no way to queue an update in advance for a Scheduled live blog.

Where a live blog appears on the public site

Automatically on its linked article's page (`/news/{slug}`) — no Homepage Builder or other placement step needed, unlike Breaking News or Polls. A reader with the article open sees new updates, pinning changes, and the Live→Ended transition itself, all without reloading the page.

Push Notifications — `/admin/push-notifications`

Sends a real browser push notification (via the Web Push standard/VAPID — works even when the reader doesn't have the site open in a tab) to every reader who has enabled notifications.

Composing one (`/admin/push-notifications/create`): Title, Body, and optionally link it to one of your Articles or a Custom URL (where tapping the notification takes the reader). Two buttons: **Save as draft** (does nothing until you come back and send it) or **Send now to every subscriber** (queues delivery immediately).

Sending a saved draft: from the list page, click **Send now** next to any Draft — same queued delivery, just done later than when you composed it. The button only ever appears next to a Draft — once a campaign is Queued or Sent, there's nothing left to click. If you somehow trigger a send on one that's already gone out (e.g. a stale page after a double click), you'll just see a friendly "already been sent" message, not an error page.

Sending is **queued, not instant** — clicking send hands the job to the background queue, which delivers to every current subscriber in the background; a subscription that's no longer valid (the reader uninstalled the browser, revoked permission, etc.) is automatically removed so future sends don't keep re-trying it.

⚠ **Known limitation, deliberate scope boundary:** unlike newsletter campaigns, a push notification cannot be edited or deleted once composed, and there is no scheduled-send-later option — compose-and-send (now or later, manually) is the whole flow. There's also no audience-segmentation option (e.g. "only readers who read category X") — every send reaches every current subscriber.

Subscription management (reader-facing, not an admin screen)

A bell icon in the site header lets a reader turn notifications on or off for their own browser — there's no admin roster of individual subscribers to manage by name, only a subscriber count shown on the Push Notifications list page for context. A reader can be subscribed whether or not they're logged in.

Advertisement Manager — `/admin/advertisements`

Manages every paid ad slot on the site: image banners, raw HTML/third-party ad snippets, and Google AdSense units, across nine positions (Header, Footer, Sidebar, Article Top, Article Middle, Article Bottom, Homepage Block, Popup, Sticky).

Creating an ad (`/admin/advertisements/create`): pick a **Position** and **Device targeting** (All/Desktop only/Mobile only), then an **Ad type**:

- **Image** — upload an image; optionally set a Click-through URL.
- **HTML** — paste raw HTML (script tags allowed — this is meant for third-party ad-network embed snippets); optionally set a Click-through URL to wrap it in a link.
- **AdSense Code** — paste the full ad-unit snippet Google gives you.
There's no separate click-through URL field for AdSense — Google's own script handles clicks and impressions for that unit.

Set a **Rotation weight** (default 1) and, optionally, **Starts at**/**Ends at** dates — an ad outside its date range simply stops being picked, no manual disabling needed. The **Active** checkbox is a manual on/off switch independent of the schedule.

When more than one ad shares a position, one is picked at random each time the slot renders, weighted by each ad's Rotation weight relative to the others sharing that position (weight 3 shows about 3x as often as weight 1). Only ads that are Active, within their date range, and matching the visitor's device are eligible — an ad that fails any one of those three checks is simply never picked, with no error shown anywhere.

Enable/disable and delete: from the list page, **Disable/Enable** toggles an ad without deleting it (handy for pausing a campaign temporarily); **Delete** removes it permanently.

Performance view (click an ad's title from the list): shows total Impressions, Clicks, and CTR, plus a day-by-day breakdown for the last 30 days. Counts are buffered and flushed into the database about once a minute — a just-rendered impression or click can take up to a minute to show up here. AdSense units always show 0 clicks/CTR here — Google measures those independently, not this site's own counter.

⚠ **Known limitation, deliberate design choice:** HTML and AdSense content is stored and rendered exactly as typed, with no sanitization — unlike CKEditor article content, which is always purified. Only staff holding the Advertisements permission (Admin/Managing Editor/Super Admin, by default) can enter this, the same trust boundary as any other admin-only raw-code field; there is no way for a reader or lower-permission staff to inject content through this screen.

Where ads appear on the public site

- **Header** — below the site header, on every page.
- **Footer** — above the site footer, on every page.
- **Sidebar** — a boxed unit on the category archive page and near the bottom of the article page. (The public frontend doesn't have a true two-column layout anywhere, so "Sidebar" is a named position rather than a literal sidebar column — see `docs/SITE-MAP.md` for the exact spots.)
- **Article Top / Article Middle / Article Bottom** — on every article page, near the top of the content, partway through, and after Related Articles respectively.
- **Homepage Block** — only where you've added an **Ad Block** section via Homepage Builder (see above); it doesn't appear on the homepage on its own.
- **Popup** — a dismissible modal shown once per browser session (per ad — dismissing one doesn't hide a *different* Popup ad shown later).
- **Sticky** — a dismissible bar fixed to the bottom of the screen, same

once-per-session dismissal.

A slot with no eligible ad renders nothing at all — no empty box, no "no ad" placeholder — exactly like every other "renders nothing rather than a stub" rule elsewhere in this admin (Homepage Builder's Poll section, etc.).

SEO Suite

Search-engine and social-sharing plumbing for the whole site — sitemaps, RSS feeds, structured data, and a few small admin screens to configure the parts that need a human decision (site identity, robots.txt rules, old-slug redirects).

What's automatic (no admin screen — just works)

- **XML sitemaps** at `/sitemap.xml` (the index — this is the one URL to submit to Google Search Console/Bing Webmaster Tools) linking out to separate sitemaps for articles (chunked 1,000 per file), categories, tags, authors, pages, and the homepage/search page. All reflect whatever is currently published/active, refreshed at least every 30 minutes.
- **Google News sitemap** at `/sitemap-news.xml` — published articles from the last 48 hours only, with the Nepali title and language tag Google News requires. Refreshes at least every 5 minutes (fresher than the regular sitemaps, since news crawlers expect that).
- **RSS feeds** — `/feed` (site-wide, 50 most recent published articles) and `/category/{slug}/feed` (per-category, same cap). Every category page links to its own feed automatically (look for the feed icon your browser or an RSS reader recognizes — there's no separate admin toggle).
- **Breadcrumbs** — every article/category/tag/author/page shows a Home → ... → current-page trail at the top of the content, and the same trail is embedded as structured data for search engines.
- **Structured data** — articles carry `NewsArticle` structured data (unchanged since Phase 6/7, now also including your logo as the publisher's logo if you've set one below); the homepage carries one sitewide `Organization` block (name, logo, social links) built from SEO Settings.
- **Canonical URLs and OG/Twitter Card meta** — every public page, including category/tag/author pages that never had this before Phase 13. An article's own SEO fields (from its edit screen) still take priority when set; everything else falls back to the SEO Settings defaults below.

SEO Settings — `/admin/seo/settings`

Super Admin/Admin only (same restriction as Watermark and Comment settings). One screen holding sitewide SEO defaults:

- **Organization/site name** and **Logo** — used in the homepage's Organization structured data, as the publisher logo on articles, and as the fallback share-image for any page without its own image.
- **Default meta description** — used on the homepage and any category/tag/author page that doesn't have a more specific description of its own.
- **Twitter/X handle** and **Social profile links** (one URL per line) — the latter become the Organization schema's social links.
- **Google News publication name** — used in the Google News sitemap; falls back to the organization name above if left blank.
- **Google/Bing site verification codes** — rendered as `<meta>` tags on every public page, for verifying site ownership in Search Console/Webmaster Tools.
- **robots.txt** — the exact text served at `/robots.txt`. Leave blank to use a sensible default (allow everything except `/admin`, with a pointer at the sitemap).

Redirects — `/admin/seo/redirects`

A simple old-path → new-path redirect table — use this whenever an article, page, or category's slug changes, so old bookmarks and search-engine-indexed links don't just start 404ing.

Adding one (`/admin/seo/redirects/create`): the **From path** (the old URL a reader might still request, e.g. `/old-article-slug`), the **To path** (where they should land — a path on this site or a full external URL), and the **Redirect type** — 301 Permanent (tells search engines to transfer the old page's ranking to the new one — use this for almost every real slug change) or 302 Temporary (only if this redirect itself might change again soon). Each redirect shows how many times it's been hit.

A redirect only ever fires once every real page/admin screen/API route has already been checked and none of them matched — it can never accidentally intercept a real page, even if a from-path happens to look similar to one.

Super Admin/Admin only. A read-only health-check table for every published article: title length and meta description length (both traffic-lighted against the recommended ranges — roughly 30–60 characters for a title, 70–160 for a description), and whether the featured image has alt text set. A summary card at the top also shows what percentage of your *entire* Media Library (not just articles) has alt text set. Nothing here is auto-fixed — it's a worklist; click **Edit** on any row to go fix it on the article itself.

Performance & caching (behind the scenes, no admin screen)

The homepage and every category/tag/author/page archive are cached for anonymous (not-logged-in) readers so they load faster under real traffic. There's nothing to configure — publishing, editing, or deleting an article (or editing a category, tag, or page) automatically refreshes exactly the pages that change, immediately, not after any delay. Logged-in staff always see a fresh, uncached page regardless.

⚠ **Known edge case:** this only reacts to changes made through the admin screens described in this manual. If content is ever changed by some other means that bypasses them entirely (direct database access, for instance), the affected page could keep showing the old version for up to an hour — in practice this should never come up in normal day-to-day use.

Theme and numeral toggles (reader-facing, not admin settings)

Two small controls appear in the header of every public page, for *readers* to adjust their own view — these aren't admin settings, and there's currently no admin screen to change the site-wide defaults:

- **Dark/light mode** (moon/sun icon, top right): toggles the whole public site's color scheme. Remembered per-visitor (their browser remembers their choice); doesn't affect what other visitors see.
- **Numeral style** (the "१ २ ३ / 123" toggle next to it): switches every number on the public site — dates, view counts, pagination — between Nepali (१ २ ३) and Arabic (123) digits. Also remembered per-visitor via a cookie.

Both default to Nepali digits / system-preference dark mode for a first-time visitor with no saved preference.

User roles and permissions

Eight roles exist. Super Admin and Admin can do everything; the rest are scoped to what a newsroom actually needs from that job title. "Everything in a module" below means create/view/edit/delete all apply.

Role	Articles	Categories/Tags	Pages	Media	Menus/Homepage	Everything else
Super Admin	Everything	Everything	Everything	Everything (incl. watermark settings*)	Everything	Everything
Admin	Everything	Everything	Everything	Everything (incl. watermark settings*)	Everything	Everything
Managing Editor	Everything, incl. publish/schedule/archive/mark-breaking	Everything	Everything	Everything except watermark settings*	Everything	Comments (full, incl. Settings [†]), polls, newsletter (incl. sending), contact inbox (incl. reply), live blogs, breaking news, push notifications, and Redirects all full access; SEO Settings/SEO Analyzer — Super Admin/Admin only, same as watermark settings*
Editor	View/edit any article, review, request changes, approve, feature-toggle — cannot publish, schedule, archive, or mark-breaking	Categories: view only. Tags: view + create (not edit/delete/merge)	—	View + upload	—	Comment moderation (view/approve/reject/spam/reply/block — not the Settings screen [†]), live blogs (view/create/edit, not delete), breaking news (view/create only, not edit/delete) — no push notification access at all
Reporter	Create, edit own drafts, submit for review	View only	—	View + upload	—	—
Journalist	Same as Reporter	View only	—	View + upload	—	—
Photographer	—	—	—	View, upload, edit (not delete)	—	—
Subscriber	View only (can't write)	—	—	—	—	View comments only

* Watermark settings (`/admin/media/settings/watermark`) are gated separately from the rest of Media — this is the one place Managing Editor doesn't have full control despite having it everywhere else in this table.

† Comment settings (`/admin/comments/settings`) are gated the same way as watermark settings — Super Admin/Admin only, via the same "settings" permission. Managing Editor and Editor can both moderate comments fully; neither can change whether guest comments or approval-required are on.

In plain terms: Reporters and Journalists write and submit; Editors review and approve (but can't push something live themselves); Managing Editors (and Admin/Super Admin) do everything including the final publish. Photographers only touch the Media Library. Subscriber is a read-only role with no admin-panel access at all beyond viewing (in practice, Subscribers can't even reach `/admin` — see the note below).

⚠ Note: **Subscriber is blocked from the entire `/admin` panel** — every other role listed above can at least log into `/admin/dashboard`, even if most screens then say "forbidden" for things outside their permissions.

What isn't built yet

If you don't see a feature mentioned anywhere above, it hasn't been built — this project is being built in numbered phases, strictly in order, and these are next in line (not necessarily in this exact order):

- **A REST API** — not built.
- **User account management** (creating/editing other staff accounts from the admin panel) — not built; see the note under "Logging in" above.