PRD Sample: Full-Product Artist Website

Jonas Monteiro: DJ, Producer, Musician & Singer-Songwriter

2026

Jonas Monteiro Artist Website

Document type: Product Requirement Document
Version: 1.0
Scope: Public-facing artist website + headless CMS + AI event assistant

---

1. Overview

This document covers the requirements for my personal artist website. I'm Jonas Monteiro: DJ, Producer, Musician, and Singer-Songwriter. The site needs to be a central hub for fans, event-goers, and booking agents: showcasing upcoming gigs with guestlist registration, music releases, videos, photos, news articles, an interactive venue map, and an AI-powered event assistant that answers questions in natural language.

The platform uses a headless CMS for all content management, with a Git-based content store. The public site is a statically generated web application served via CDN.

This format covers a complete web product built from scratch, with multiple public-facing sections, a CMS admin, and a third-party AI integration.

---

2. User Personas

Fan / Site Visitor A music lover who follows Jonas on social media or hears about an event. Visits the site to find out about upcoming gigs, listen to music, browse photos, and read news.

Event-Goer Attends Jonas's events. Uses the site specifically to find upcoming dates, join the guestlist for free/discounted entry, and subscribe to a calendar feed so events appear in their personal calendar.

Booking Agent / Promoter A venue representative or event organiser who wants to book Jonas. Uses the Booking page to submit an enquiry with event details, date, and budget.

Content Admin (Me / My Team) That's me (Jonas) or someone I delegate to. We need to be able to log into the admin panel to add or update events, upload news articles, manage music releases, and publish photos and videos (without having to write any code).

---

3. Data Models

3.1 Event

Field	Type	Required	Notes
`title`	string	Yes	Event name
`description`	text	Yes	Full event description with rich text
`event_image`	file	No	Custom image; falls back to venue image
`start_datetime`	datetime	Yes	Used for sorting and display
`end_datetime`	datetime	No	
`venue_id`	reference	Yes	Linked to Venue record
`is_pinned`	boolean	No	Pin to top of Upcoming list
`is_recurring`	boolean	No	Toggle for one-time vs recurring
`skip_next_recurrence`	boolean	No	Skip the next occurrence only
`event_type`	enum	Yes	`guestlist` or `booking`
`guestlist_details`	text	No	Shown on the guestlist form

3.2 Venue

Field	Type	Required	Notes
`name`	string	Yes	
`address`	string	Yes	
`city`	string	Yes	
`map_coordinates`	lat/lng	Yes	Used on the Map page
`venue_image`	file	No	Displayed on event cards
`website_url`	string	No	

3.3 News Article

Field	Type	Required	Notes
`title`	string	Yes	
`slug`	string	Yes	Auto-generated from title; unique
`body`	rich text	Yes	Full article content
`excerpt`	text	No	Used in listing cards; max 300 chars
`cover_image`	file	No	
`published_at`	date	Yes	Controls display order and public visibility
`tags`	string array	No	For filtering (future scope)

3.4 Music Release

Field	Type	Required	Notes
`title`	string	Yes	Track or album name
`type`	enum	Yes	`single`, `ep`, `album`, `mix`
`cover_image`	file	Yes	Album/track artwork
`release_date`	date	Yes	
`streaming_url`	string	No	Spotify, Apple Music, or SoundCloud link
`embed_code`	text	No	Raw embed HTML for SoundCloud / Spotify player

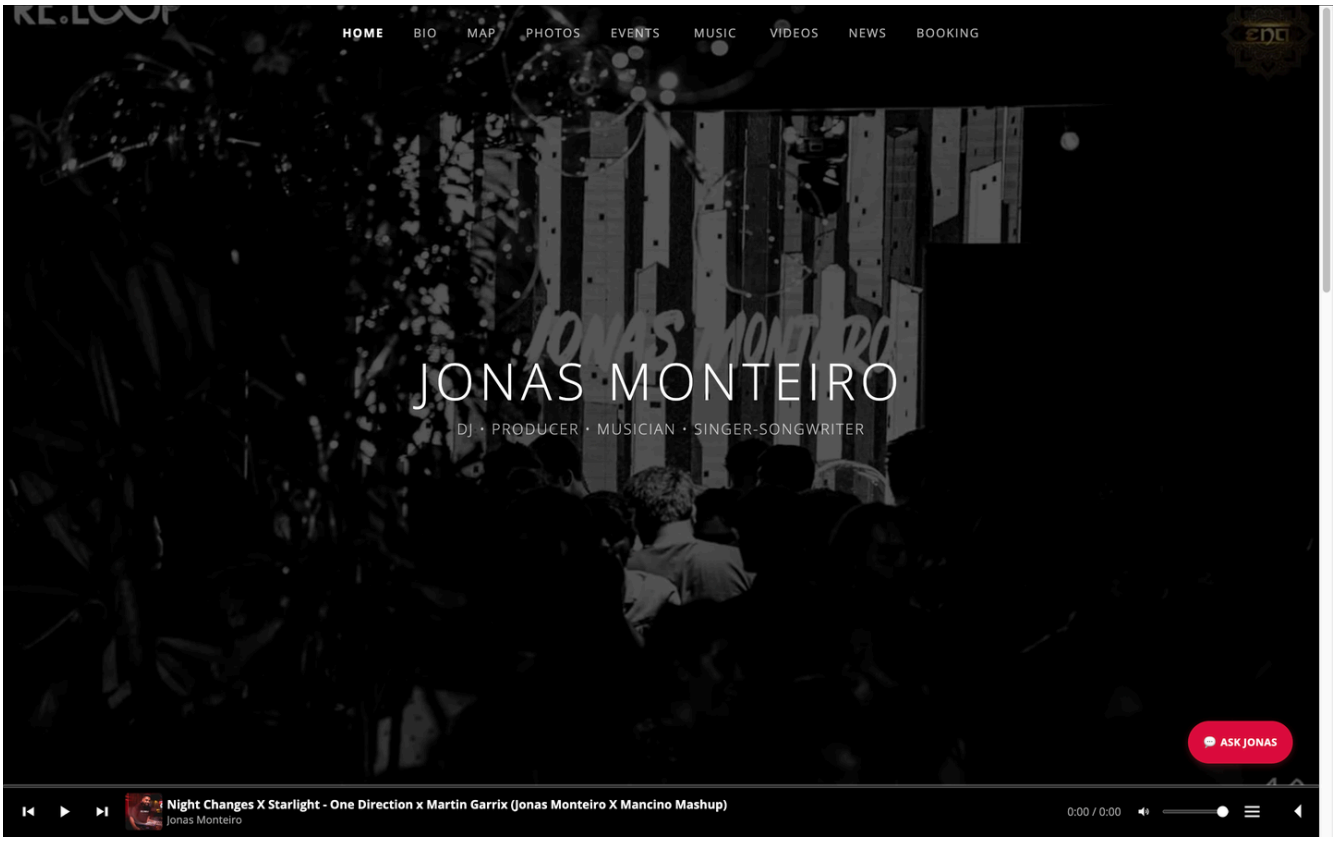
3.5 Video

Field	Type	Required	Notes
`title`	string	Yes	
`youtube_url`	string	Yes	Full YouTube URL; embed ID extracted automatically
`thumbnail`	file	No	Falls back to YouTube thumbnail
`published_at`	date	Yes	

---

4. Public Site Features

Section 1: Home / Hero

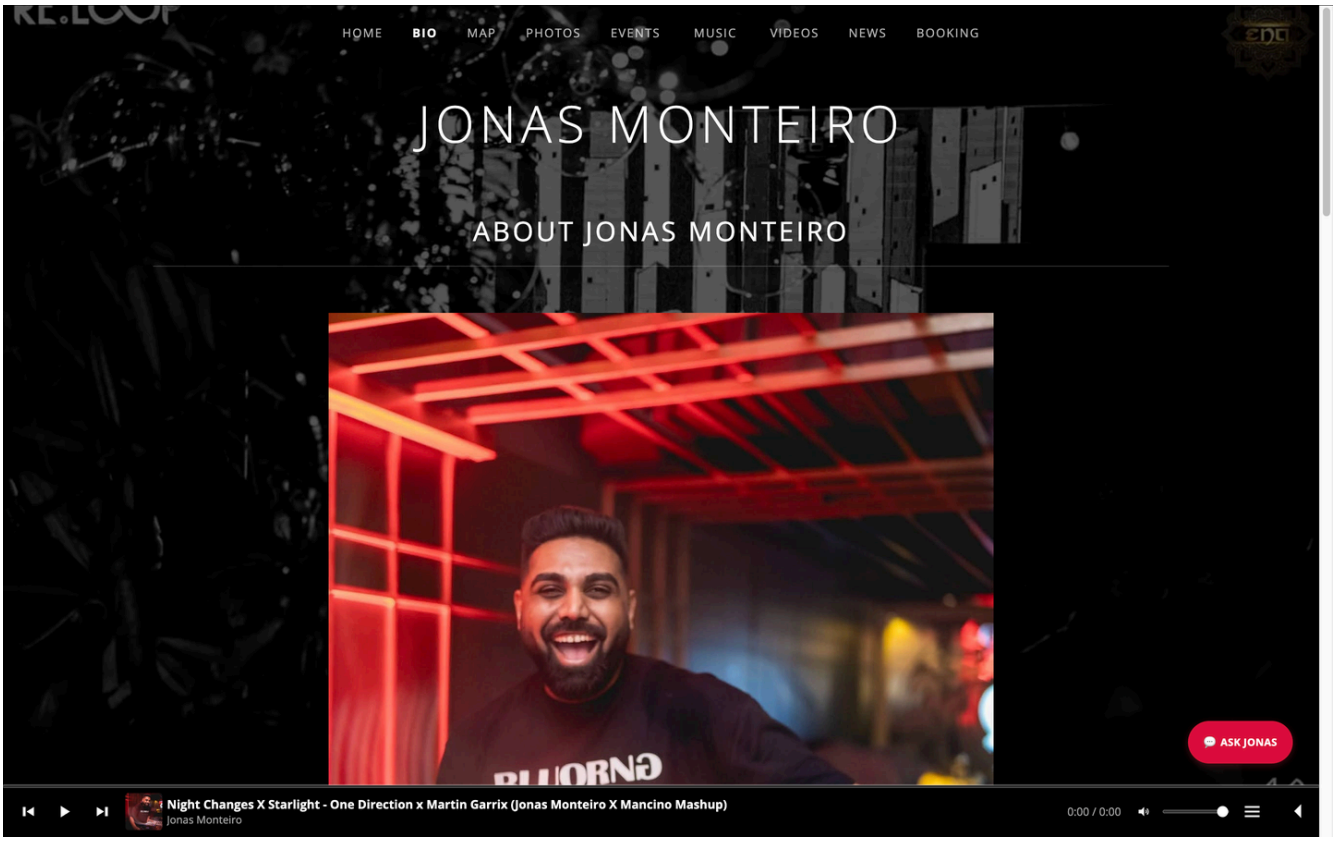


Route: /

Requirements:

1. Full-viewport hero section with artist photograph as background. Artist name (Jonas Monteiro) and subtitle (DJ, Producer, Musician, Singer-Songwriter) are centred in the hero.
2. Persistent top navigation bar: Home, Bio, Map, Photos, Events, Music, Videos, News, Booking. Active section is highlighted. On scroll, the nav background darkens for legibility.
3. A persistent music player sits fixed to the bottom of every page. It plays the latest release on page load (muted by default; user clicks play). Shows track title and artist name. Controls: play/pause, skip, mute/unmute, seek bar, volume.
4. The ASK JONAS AI chatbot button is fixed to the bottom-right corner on all pages.

Section 2: Bio

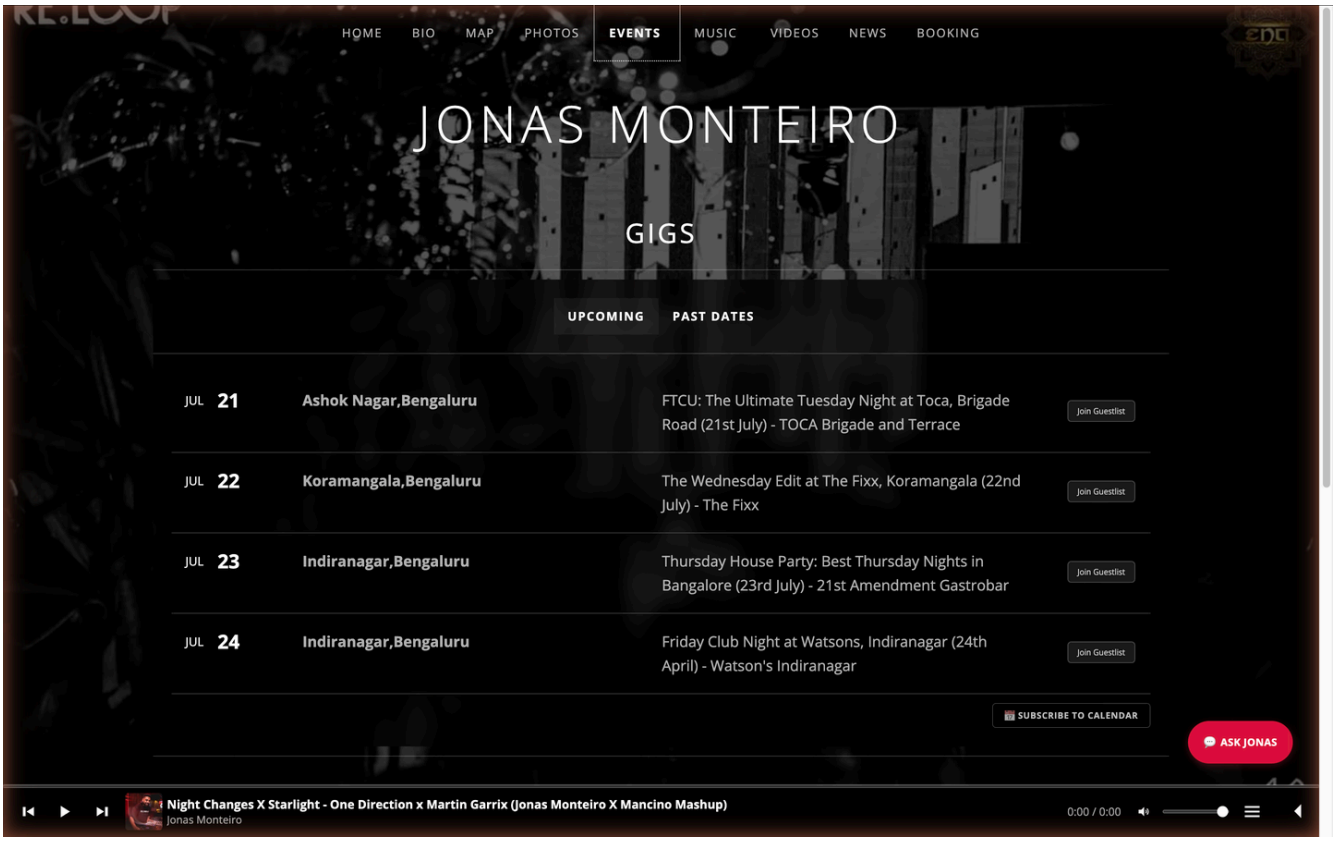


Route: `/bio`

Requirements:

1. Full-width hero image at the top of the page.
2. Bio text is managed in the CMS and rendered as rich text. Supports headings, bold, italic, and links.
3. Social media links are displayed alongside the bio (Instagram, Facebook, SoundCloud, Spotify).
4. Page title tag: “Bio | Jonas Monteiro”. Meta description is drawn from the first 160 characters of the bio body.

Section 3: Events / Gigs



Route: `/events`

Requirements:

1. Two tabs at the top of the page: **Upcoming** and **Past Dates**. Upcoming is the default.
2. Each event row displays: date (month + day number), venue city, event title, and a CTA button.
3. The CTA button is determined by `event_type`: if `guestlist`, show "Join Guestlist"; if `booking`, show "Book Tickets" (links to external URL).
4. Upcoming events are sorted by `start_datetime` ascending (soonest first). Past events are sorted descending (most recent first).
5. Pinned events (`is_pinned = true`) appear at the top of the Upcoming list regardless of date.
6. A "Subscribe to Calendar" link is shown below the Upcoming list. It links to a publicly accessible iCal feed (`/events.ics`) containing all upcoming events.
7. If no upcoming events exist, the Upcoming tab shows: "No upcoming events. Check back soon."

Guestlist Registration

When a visitor clicks "Join Guestlist", a modal or dedicated page opens with:

Acceptance criteria:

1. Form fields: First Name (required), Last Name (required), Email (required, email format), Phone (optional), Number of Guests (required, integer 1–4).

2. On submit, the entry is recorded and the user sees a confirmation message: "You're on the list! See you at [Event Name]."
3. If the email is already registered for this event: "You're already on the guestlist for this event."
4. The guestlist closes automatically at the event `start_datetime`. Submissions after that time show: "Guestlist is now closed for this event."
5. We need to be able to log into the admin panel and view or export the guestlist for each event.

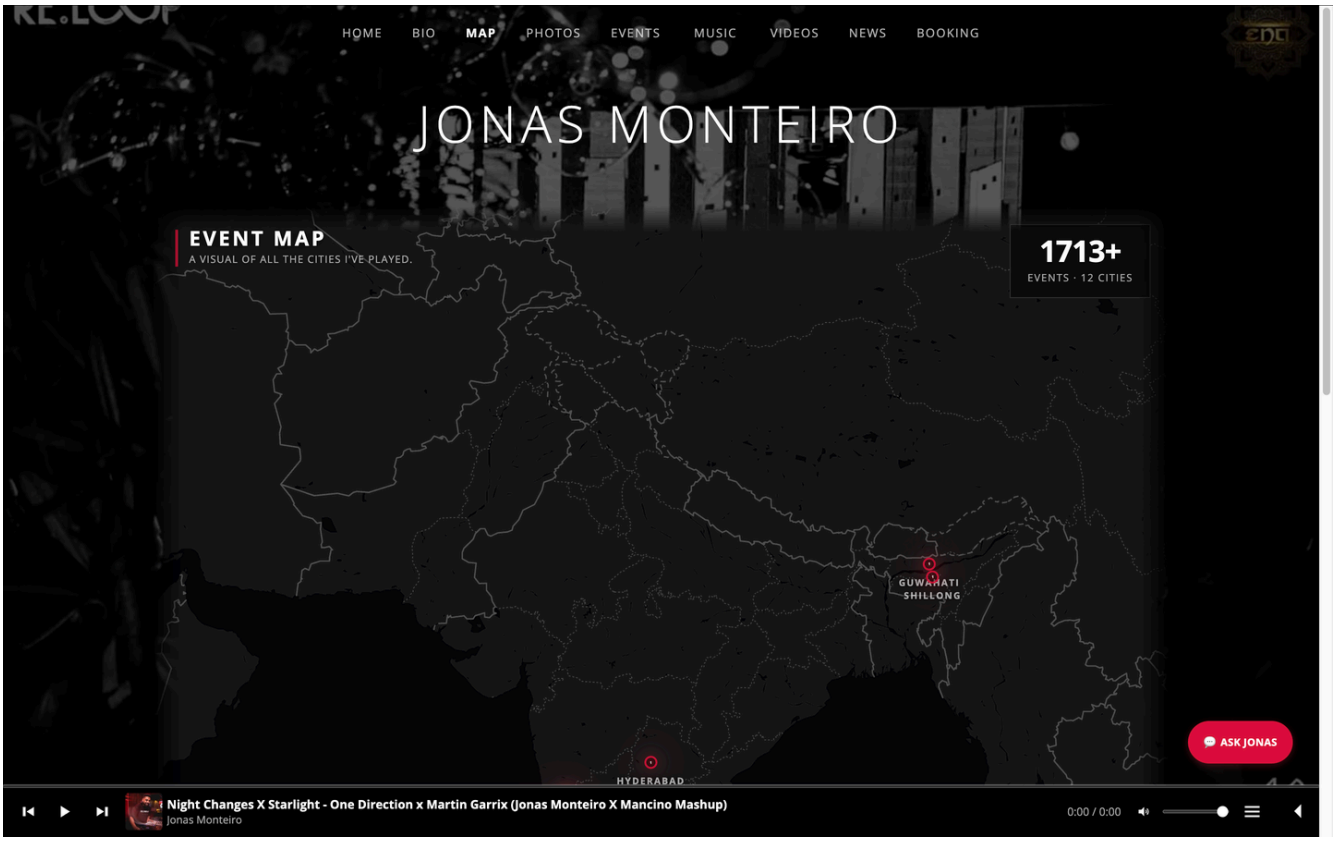
---

iCal Feed

Acceptance criteria:

1. `GET /events.ics` returns a valid iCalendar file (RFC 5545) containing all upcoming events.
2. Each event in the feed includes: `SUMMARY` (event title), `DTSTART`, `DTEND`, `LOCATION` (venue name and city), `DESCRIPTION` (first 300 characters of event description), `URL` (link to the event on the site).
3. The feed is publicly accessible with no authentication.
4. The feed is regenerated on each CMS publish that affects events.

Section 4: Map



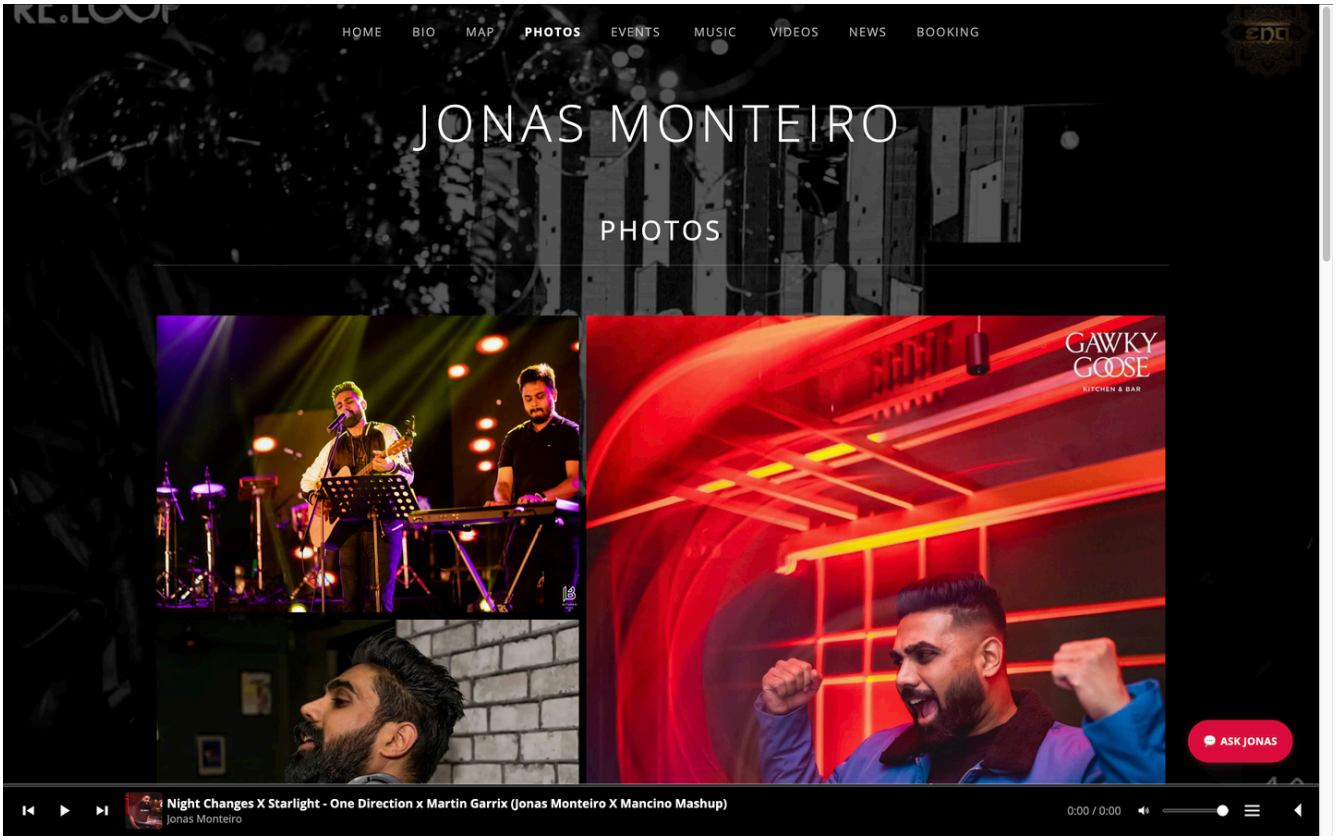
Route: `/map`

Requirements:

1. An interactive map (Google Maps or equivalent) fills the page. Each venue where Jonas performs is marked with a custom pin.

2. Clicking a pin opens an info window showing: venue name, address, and a link to view upcoming events at that venue (filters the Events page by venue).
3. Venue coordinates are managed in the Venues CMS collection.
4. The map loads with a default zoom level that shows all markers within the geographic cluster of venues.

Section 5: Photos

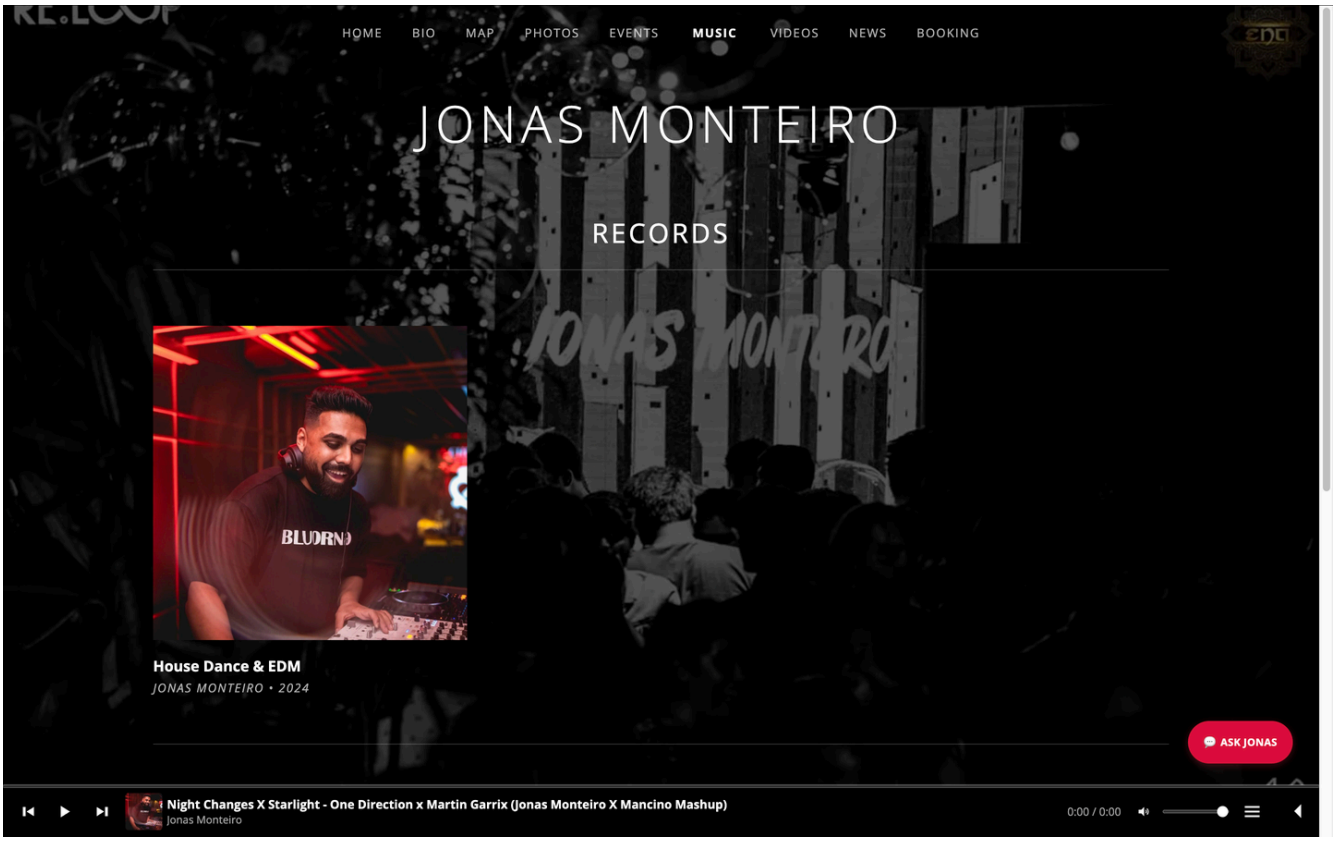


Route: `/photos`

Requirements:

1. Responsive masonry or grid gallery of photographs. The gallery configuration (set of images) is managed via the Photos Page CMS collection.
2. Clicking a photo opens a full-screen lightbox with navigation arrows to browse adjacent photos.
3. Images are served in modern formats (WebP with JPEG fallback) and lazy-loaded below the fold.
4. On mobile, the grid collapses to a single column.

Section 6: Music / Records

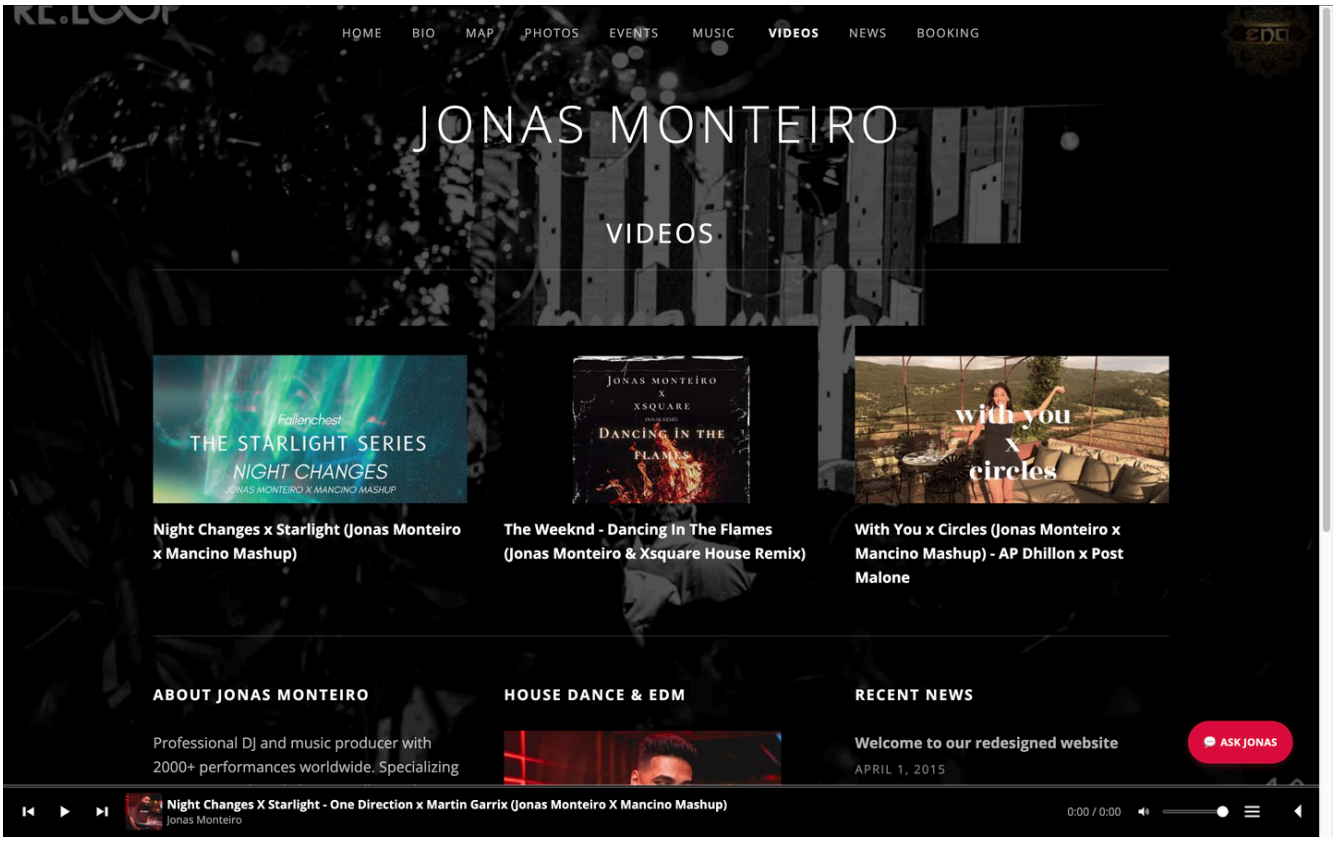


Route: `/music`

Requirements:

1. Music releases are displayed as cards: cover artwork, release title, type (Single, EP, Album, Mix), and release date.
2. Cards are sorted by `release_date` descending (newest first).
3. Each card links to the streaming platform (Spotify, SoundCloud, Apple Music) via `streaming_url`.
4. If an `embed_code` is provided, a playback widget is shown on hover or tap (e.g. a SoundCloud waveform player).

Section 7: Videos

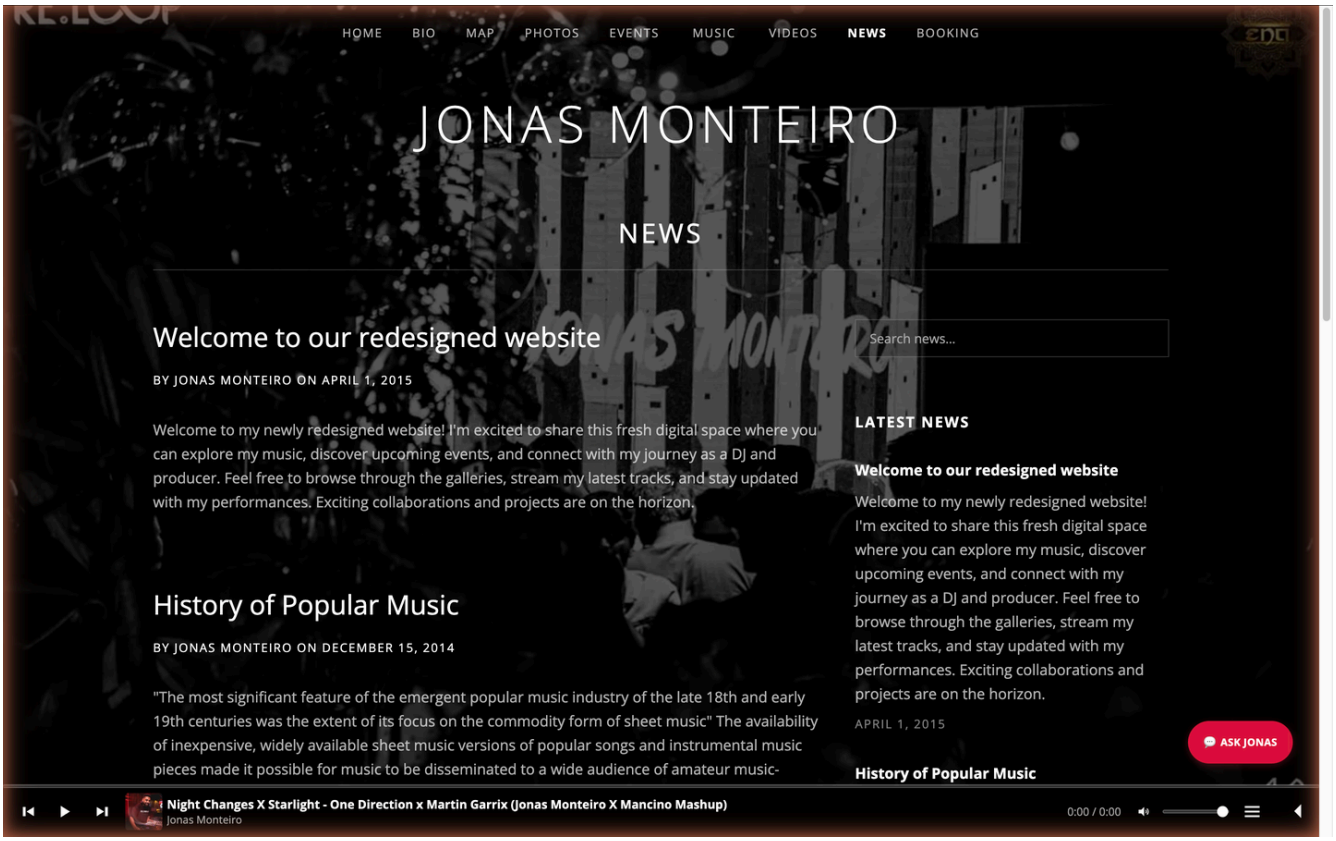


Route: `/videos`

Requirements:

1. Videos are displayed as a grid of thumbnail cards: thumbnail image, title, and published date.
2. Videos are sorted by `published_at` descending.
3. Clicking a thumbnail opens an inline YouTube embed (or modal) that plays the video without navigating away from the page.
4. Video thumbnails are fetched automatically from YouTube using the extracted video ID, unless a custom thumbnail is uploaded via the CMS.

Section 8: News



Route: `/news`

Requirements:

1. News articles are listed as cards sorted by `published_at` descending.
2. Each card shows: cover image (or placeholder), article title, publication date, and excerpt.
3. Clicking a card navigates to the article detail page at `/news/{slug}`.
4. The listing paginates at 12 articles per page with a “Load more” button or paginator at the bottom.

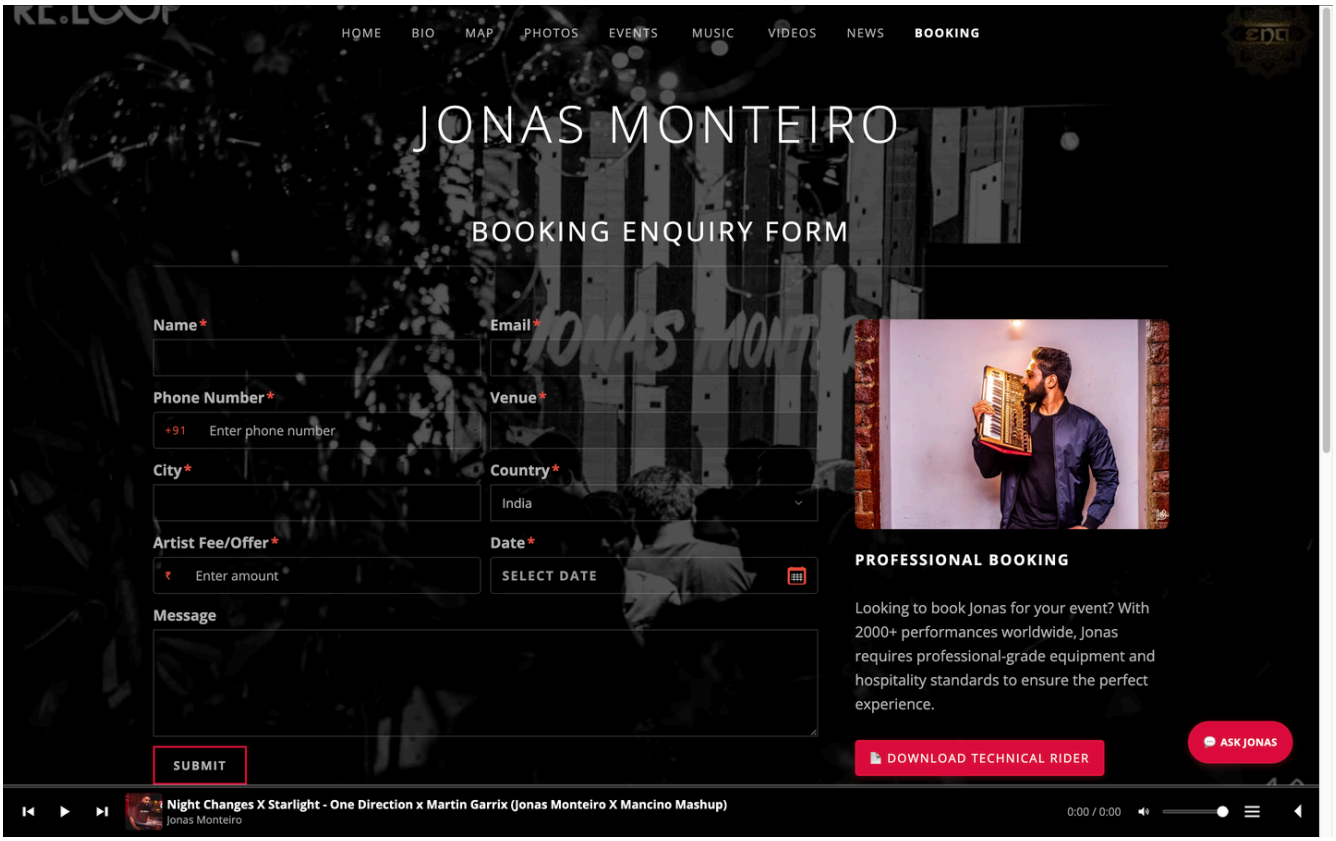


Route: `/news/{slug}`

Requirements:

1. Article detail page shows: cover image (full-width), title, publication date, author (optional), and full body text rendered as HTML.
2. A sidebar or related section links to recent articles.
3. If the slug does not exist or the `published_at` date is in the future, the page returns a 404.
4. "Back to News" link at the top navigates back to `/news`.
5. SEO: `<title>` is set to the article title; meta description is the article excerpt; Open Graph image is the cover image.

Section 9: Booking



Route: `/booking`

Requirements:

The booking page allows promoters and venue managers to submit an enquiry to book Jonas.

Form fields:

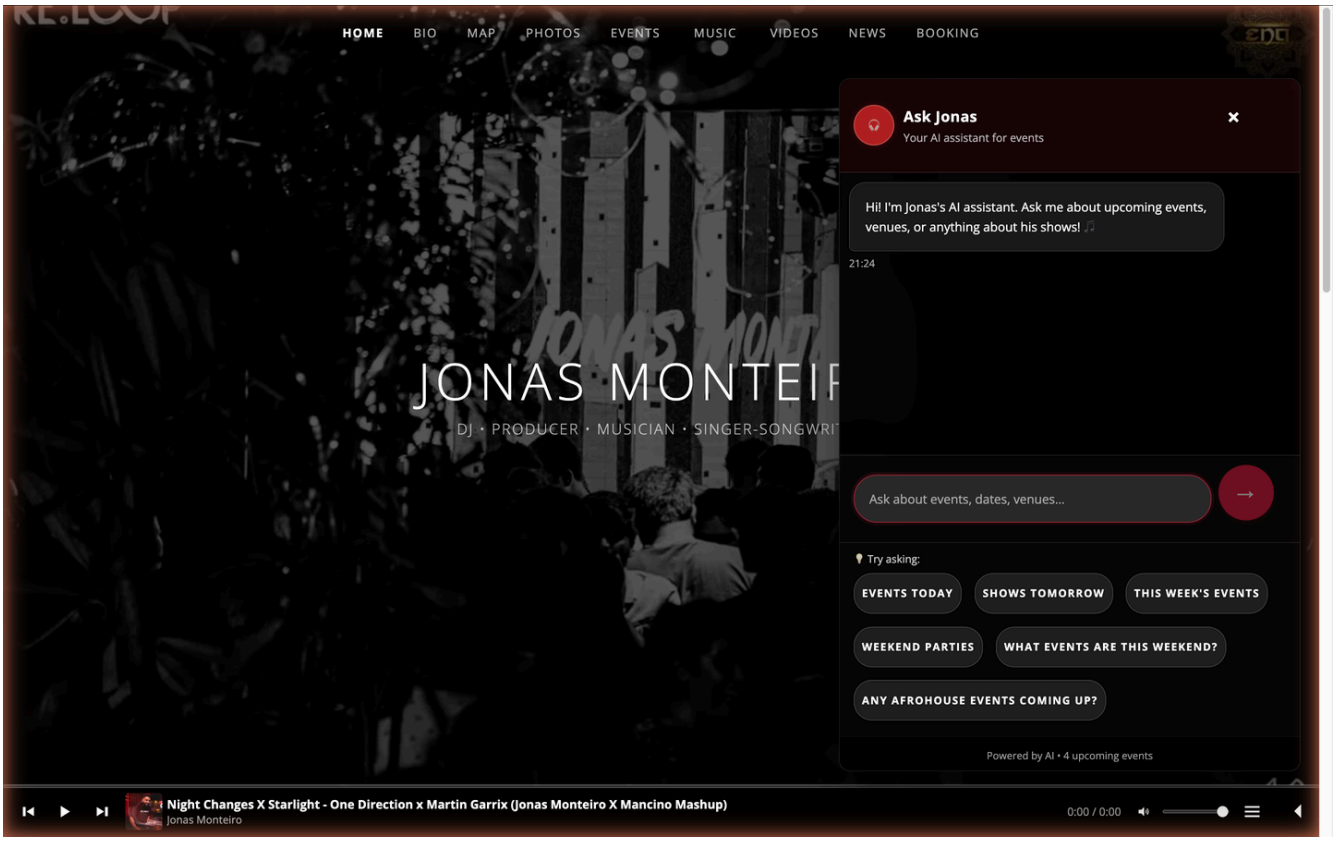
- Your Name (required)
- Email Address (required, email format)
- Phone Number (optional)
- Event Type (required) — dropdown: Club Night, Private Event, Festival, Corporate, Wedding, Other
- Event Date (required, date picker; must be a future date)
- Venue / Location (required, free text)
- Estimated Attendance (optional, number)
- Budget Range (optional) — dropdown: Under $500, $500–$1k, $1k–$3k, $3k–$5k, $5k+, Prefer not to say
- Message / Additional Details (optional, textarea)

Booking form acceptance criteria:

1. All required fields are validated on submit. Inline errors are shown beneath each field that fails.
2. Email format is validated: an invalid format shows “Please enter a valid email address.”
3. Event Date must be a future date. Selecting a past date shows “Event date must be in the future.”

4. On successful submission, a confirmation message is shown: "Thanks for reaching out! We'll be in touch within 48 hours." The form is cleared.
5. The submission is emailed to our configured address and stored as a record we can access from the admin panel.
6. If submission fails (network or server error), the form data is preserved and an error message is shown: "Something went wrong. Please try again or email us directly at [contact email]."
7. The booking form is protected against spam via honeypot field (not visible to human users).

Section 10: AI Event Assistant ("Ask Jonas")



Trigger: Fixed button labelled "ASK JONAS" in the bottom-right corner of every page.

Overview:

"Ask Jonas" is an AI-powered chat assistant that answers questions about upcoming events, venues, and dates in natural language. It is powered by a large language model with access to the current events dataset.

Requirements:

1. Clicking "ASK JONAS" opens a chat panel. The panel header shows "Ask Jonas" with the subtitle "Your AI assistant for events".
2. The assistant's opening message is: "Hi! I'm Jonas's AI assistant. Ask me about upcoming events, venues, or anything about his shows!"
3. Quick-action chips are shown beneath the opening message: *Events Today*, *Shows Tomorrow*, *This Week's Events*, *Weekend Parties*, *What events are this weekend?*, *Any Afrohouse events coming up?*
4. Tapping a chip sends it as a user message and the assistant responds with relevant event information.

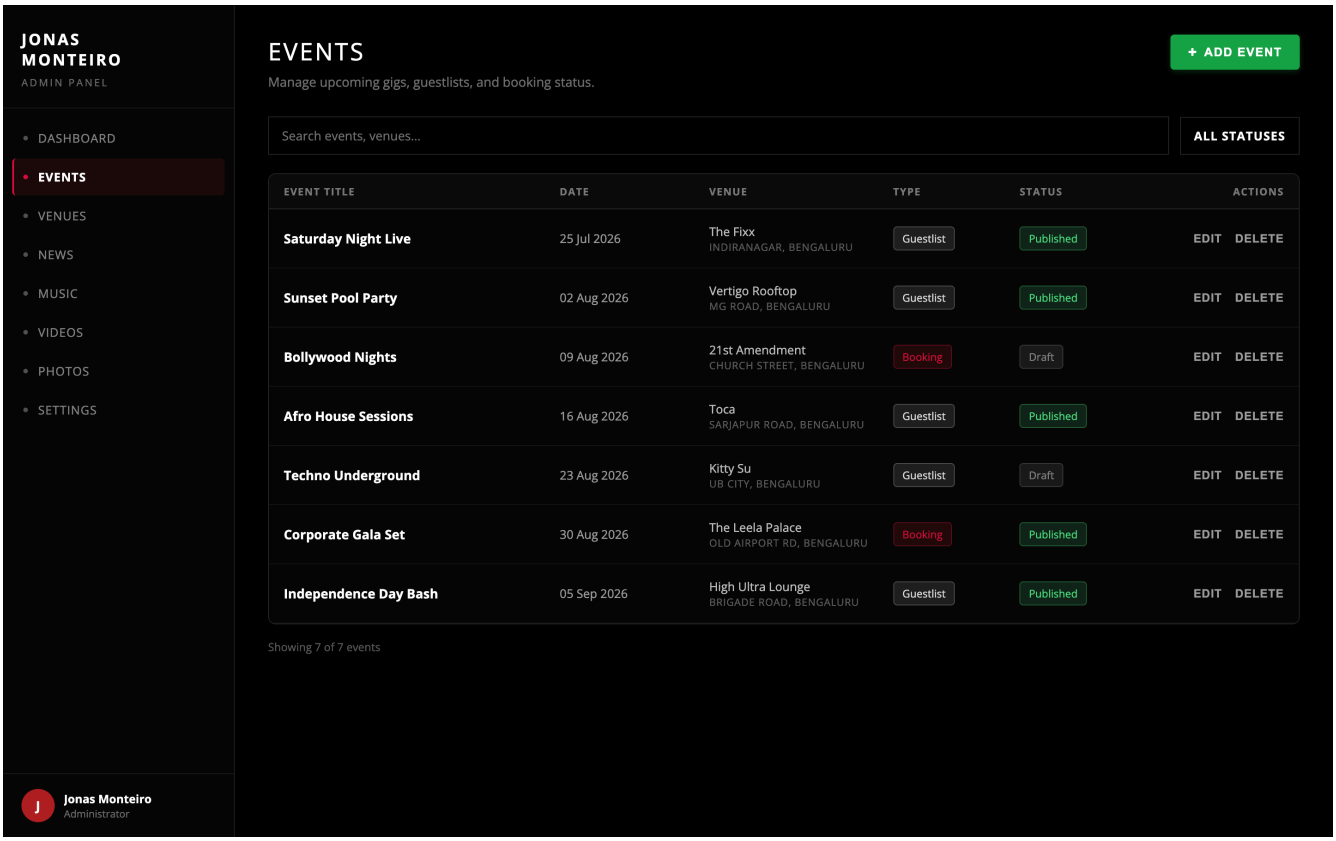
5. The assistant has access to the full events dataset (upcoming events only). It can answer questions about: dates, venues, event names, guestlist availability, and event type.
6. The assistant does not answer questions unrelated to Jonas's events. For off-topic queries, it responds: "I'm best at answering questions about Jonas's shows and events. Is there an upcoming gig you'd like to know about?"
7. The chat panel is closeable via an X button. The panel state is reset on close.
8. Footer note: "Powered by AI — [N] upcoming events" where N is the count of upcoming events in the dataset.

---

5. CMS Admin

The site uses a headless CMS with a Git-backed content store. Admin access is at `/admin`. All content changes are committed to the Git repository and trigger a site rebuild and deployment.

Admin: Events Collection

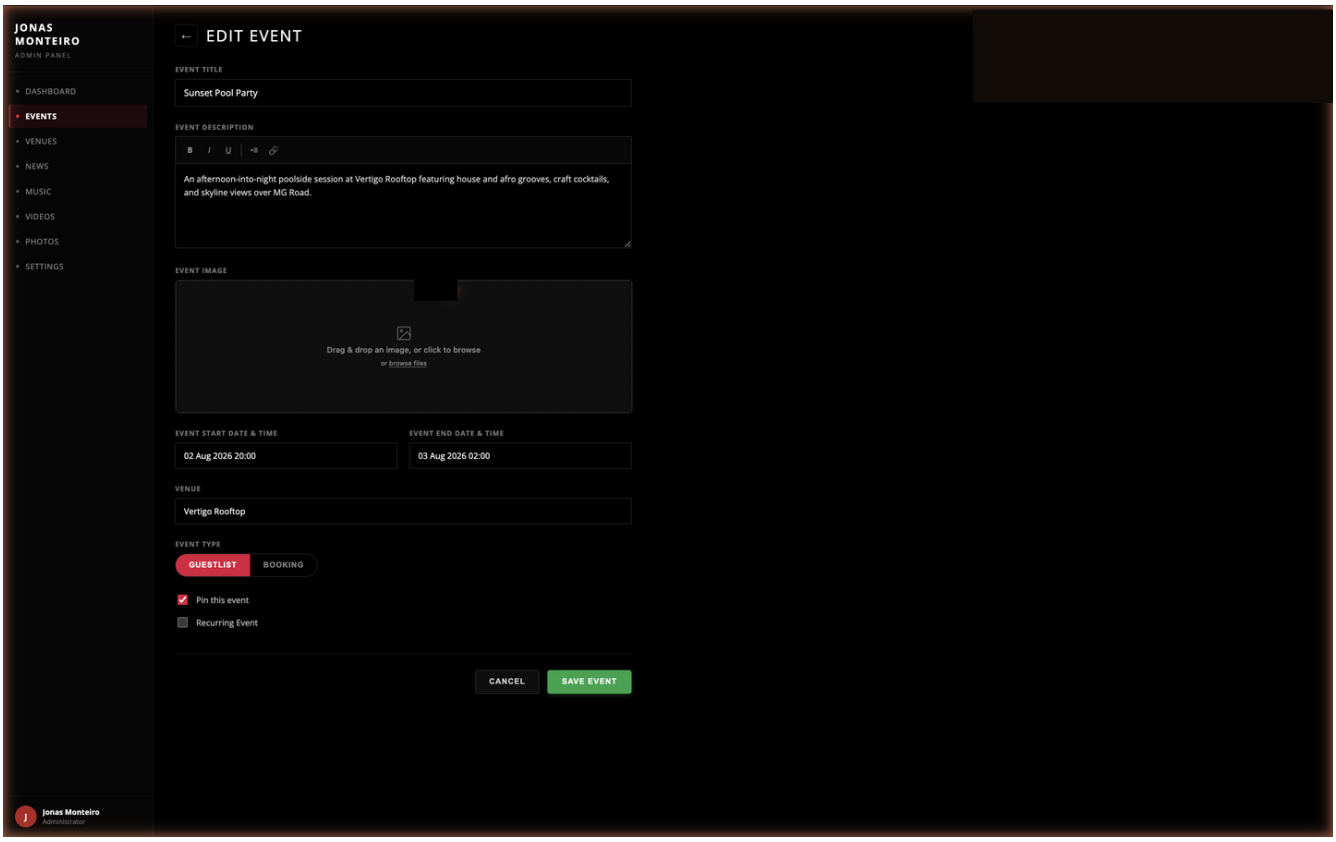


Sections managed: Events, Venues, Music, Videos, Photos, Bio, News, Settings

Events listing requirements:

1. Events are listed with columns: Event Title, Date, Venue, Type, Status.
2. Type badge shows Guestlist or Booking. Status badge shows Published or Draft.
3. A search field filters events by title. A status dropdown filters by Published / Draft.
4. An "Add Event" button in the top-right opens a blank event edit form.

5. Each row has Edit and Delete action buttons.

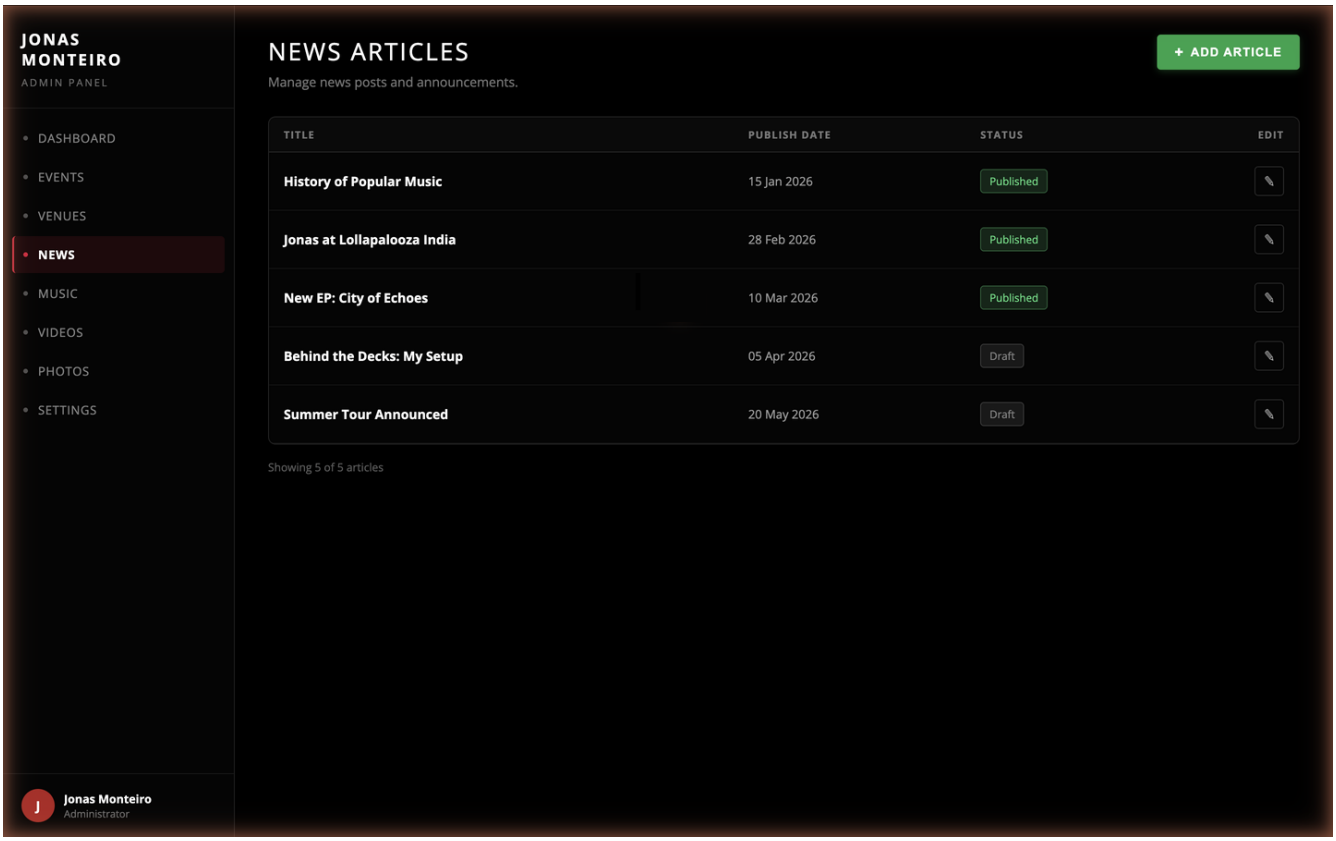


Event edit form fields:

- Event Title (text, required)
- Event Description (rich text editor)
- Event Image (drag-and-drop file upload, optional)
- Event Start Date & Time (datetime, required)
- Event End Date & Time (datetime, optional)
- Venue (text field)
- Event Type (pill toggle: Guestlist / Booking)
- Pin this event (checkbox)
- Recurring Event (checkbox)

Save Event button commits the record. **Cancel** discards changes and returns to the listing.

Admin: News Collection



News listing requirements:

1. All news articles are listed with columns: Title, Publish Date, Status.
2. Status is shown as a pill badge: green for Published, grey for Draft.
3. An "Add Article" button in the top-right creates a new article.
4. Each row has an edit icon linking to the article edit form.

News article edit form fields:

- Title (text, required)
- Publish Date (date)
- Cover Image (file upload)
- Excerpt (textarea, optional)
- Body (rich text editor with: headings, bold, italic, links, bullet lists, image upload)

Draft behaviour: Articles with a `published_at` date in the future are treated as drafts and not shown on the public site until that date is reached.

---

6. SEO & Technical Requirements

This section covers the search engine optimisation and discoverability requirements for the public site.

6.1 Page Titles and Meta Descriptions

Page	Title tag format	Meta description source
Home	`Jonas Monteiro: DJ, Producer, Musician`	Hardcoded site description
Bio	`Bio | Jonas Monteiro`	First 160 chars of bio body
Events	`Events | Jonas Monteiro`	Hardcoded
News listing	`News | Jonas Monteiro`	Hardcoded
News article	`[Article Title] | Jonas Monteiro`	Article excerpt
Music	`Music | Jonas Monteiro`	Hardcoded
Videos	`Videos | Jonas Monteiro`	Hardcoded
Booking	`Booking | Jonas Monteiro`	Hardcoded

All title tags follow the format `[Page Name] | Jonas Monteiro`. The site name suffix is applied globally.

6.2 Open Graph & Social Sharing

Acceptance criteria:

1. Every page includes `og:title`, `og:description`, `og:image`, `og:url`, and `og:type` meta tags.
2. The default `og:image` is a designated site-wide social sharing image (1200x630px). News article pages override it with the article cover image.
3. `twitter:card` is set to `summary_large_image` on all pages.
4. `twitter:site` is set to the artist's Twitter/X handle.
5. Event pages (individual event detail, if implemented) include an event-specific `og:image` using the event image or venue image.

6.3 Structured Data (JSON-LD)

Acceptance criteria:

1. The home page includes a `Person` schema for Jonas Monteiro with: `name`, `jobTitle` ("DJ, Producer, Musician"), `url`, `sameAs` (array of social profile URLs: Instagram, Facebook, SoundCloud, Spotify).
2. Each event in the Events section has an `Event` schema object with: `name`, `startDate`, `endDate` (if set), `location` (`Place` with venue name and address), `url`, `performer` (referencing the Jonas Monteiro `Person`).
3. News article pages include an `Article` schema with: `headline`, `author`, `datePublished`, `image`, `description`.
4. Structured data is rendered as a `<script type="application/ld+json">` block in the `<head>` of each relevant page.
5. Event structured data makes events eligible for Google's Event rich results (date, venue, and title shown directly in search results).

6.4 Sitemap & Crawlability

Acceptance criteria:

1. A `sitemap.xml` is generated at site build time containing all public URLs: static pages (home, bio, map, photos, music, videos, events, booking) plus all published news article URLs.
2. `robots.txt` permits crawling of all public pages. The `/admin` path is disallowed.
3. Canonical `<link rel="canonical">` tags are included on all pages to prevent duplicate content.
4. All pages use semantic HTML landmarks: `<header>`, `<main>`, `<nav>`, `<footer>`, and `<article>` on news detail pages.

6.5 Performance & Core Web Vitals

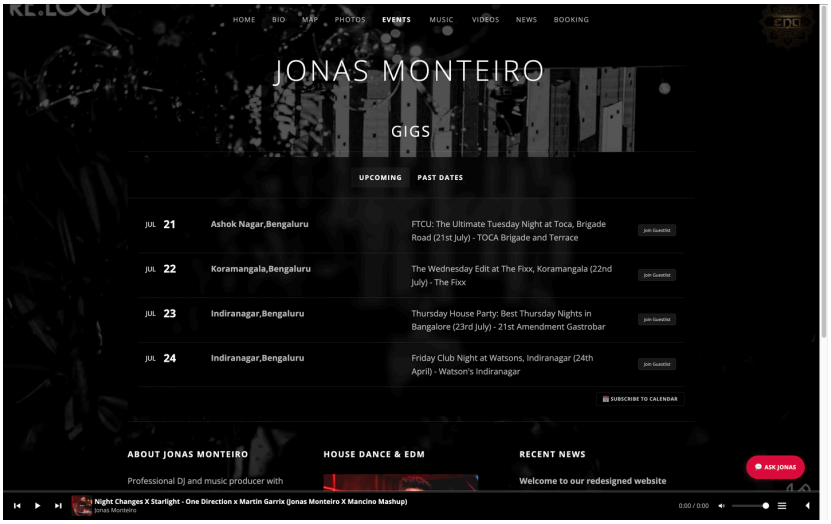
Acceptance criteria:

1. All images are served in WebP format with JPEG fallback. Images are sized appropriately per breakpoint (srcset).
2. The site achieves a Lighthouse Performance score of 90+ on desktop and 75+ on mobile.
3. The hero background image is preloaded (`<link rel="preload">`).
4. Third-party scripts (Google Maps, YouTube embeds, AI chatbot) are loaded lazily: only when the relevant section or component is in the viewport or activated.

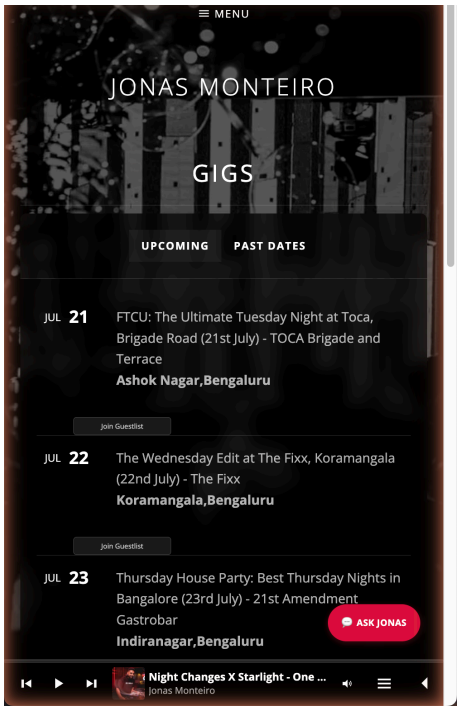
---

7. Responsive Design

The site is fully responsive across desktop, tablet, and mobile viewports.



Desktop (>=768px): Full horizontal nav; multi-column event rows; grid gallery.



Mobile (<768px): Hamburger menu; single-column event list; single-column photo grid.

Responsive breakpoints:

Breakpoint	Width	Layout changes
Desktop	$\geq$ 768px	Full horizontal nav; multi-column event rows; grid gallery
Mobile	$<$ 768px	Hamburger menu; single-column event list; single-column photo grid

---

8. Scope Boundaries

In scope:

- Public site: Home, Bio, Map, Photos, Events (Upcoming + Past), Music, Videos, News (listing + article detail), Booking
- Guestlist registration modal per event
- iCal calendar feed for upcoming events
- AI chatbot (Ask Jonas) powered by events dataset
- Headless CMS with Git-backed content store
- Collections: Events, Venues, Music, Videos, Photos, Bio, News, Settings
- SEO: page titles, meta descriptions, Open Graph, JSON-LD structured data, sitemap, canonical URLs
- Responsive design: desktop (>=768px) and mobile (<768px)
- Persistent bottom music player
- Static site generation (CDN-hosted)

Out of scope for v1:

- User accounts or fan login
- E-commerce (merch store, ticket purchases)
- Comments on news articles
- Recurring events with auto-generated future instances (managed manually)
- Analytics dashboard within the CMS
- Multi-language / internationalisation
- Push notifications for upcoming events
- Guestlist waitlist management (over-capacity handling)

---

9. Prerequisites

1. Domain name and DNS configuration.
2. Hosting and CDN provider (Netlify, Vercel, or equivalent).
3. CMS platform account for the admin content editor.
4. Google Maps API key (for the Map page).
5. AI platform API key for the Ask Jonas chatbot (e.g. OpenAI, Anthropic).
6. Transactional email provider for booking form submissions (SendGrid, Resend, or equivalent).
7. Social media profile URLs for JSON-LD `sameAs` and Twitter Card configuration.
8. Photography assets (hero image, gallery photos) and artist biography copy.

This is a sample PRD based on a real artist website. The format shown here covers a complete public-facing product with a headless CMS, AI integration, and SEO requirements. Adjust the sections, data models, and feature set to match your project.