🗂️INDICE MASTER — MagixPromotion Website (Wagtail + React)

> **Progetto:** Sito corporate per agenzia di booking/management artistico del Nord Italia
> **Stack:** Django 5.2 / Wagtail 7.x LTS (Backend) + React 19 / TypeScript / Tailwind (Frontend)
> **Template di partenza:** `template-strutturale/` (React SPA con Vite)
> **Data:** 16 febbraio 2026

Convenzioni per Agenti

Regola	Dettaglio
Context budget	Ogni task è progettato per stare sotto il 50% del context window. Leggere SOLO i file elencati in `FILES_IN_SCOPE`.
Nessun cross-read	Non leggere task di altre fasi salvo esplicito riferimento in `DIPENDENZE`.
Output atomico	Ogni task produce file specifici elencati in `OUTPUT_ATTESO`. Non creare file fuori scope.
Naming	App Django: `core`, `booking`, `events`, `artists`, `navigation`. Prefisso componenti React: nessun prefisso, PascalCase.
Lingua codice	Variabili/classi Python e TS in inglese. Commenti e docstring in italiano. Contenuti CMS bilingue IT/EN.
TDD obbligatorio	Ogni task DEVE iniziare dalla scrittura dei test (sezione `### 0. Test TDD`). Nessun codice di implementazione può essere scritto prima che i test esistano. Approccio Red → Green → Refactor.
Security audit	Ogni task DEVE includere una checklist di sicurezza: sanitizzazione input, protezione CSRF, permessi utente, escape HTML, validazione file upload, rate limiting dove applicabile.
Verifica integrazione	Prima di chiudere un task, verificare integrazione con: `urls.py`, `INSTALLED_APPS`, API router, navigazione frontend, `types.ts`, e coerenza con i task collegati.
Mappe e geocoding	Usare OpenStreetMap + Nominatim per mappe e geocoding. NON usare Google Maps API. Per la navigazione mobile, usare deep link `geo:` (Android) e `maps.apple.com` (iOS).
Dati azienda da CMS	I dati aziendali (telefono, email, indirizzo, P.IVA, social URL) provengono da `MagixSiteSettings` (wagtail.contrib.settings). Non hardcodare.
Traduzioni automatiche	Usare Gemini API come machine translator per wagtail-localize (>=1.13). Le stringhe UI vanno nei file `.po` (gettext). Le variabili non traducibili vanno in SiteSettings.
Wagtail 7.x notes	Deferred validation: i campi non-text required devono avere `null=True` per permettere save-draft (es. DateField). `USE_L10N` rimosso (Django 5.x). `wagtail-localize>=1.13` supporta fino a Wagtail 7.2; aggiornare quando esce supporto 7.4 LTS.
Sintesi attività in calce	A task completato, l'agente DEVE aggiungere in fondo al file una sezione `## SINTESI ATTIVITÀ` con: elenco puntato delle operazioni svolte, file creati/modificati, decisioni prese e motivazioni, eventuali deviazioni dal piano originale. Serve come log per la revisione futura.
Ottimizzazione post-task	Dopo il completamento e la documentazione della sintesi, un agente dedicato ottimizzerà il file `.md` del task: rimuove i blocchi di codice di esempio, le trasforma in istruzioni tecniche, i criteri di accettazione, la security checklist, la sezione TDD e la sintesi attività. Obiettivo: ridurre il peso del file preservando il valore documentale.
Scheda manuale post-task	Lo stesso agente dell'ottimizzazione genera un file `docs/NN-titolo-breve.md` con una scheda concisa del task appena sviluppato: scopo, componenti coinvolti, endpoint/URL esposti, configurazioni rilevanti, comandi utili. Formato uniforme, pronto per essere assemblato in un manuale operativo del progetto.

Struttura App Django

```
magixpromotion/
├── manage.py
├── config/                  # Progetto Django (settings, urls, wsgi)
│   ├── settings/
│   │   ├── base.py
│   │   ├── dev.py
│   │   └── production.py
│   ├── urls.py
│   └── wsgi.py
├── core/                    # HomePage, blocchi condivisi, templatetags
├── artists/                 # ArtistListingPage, ArtistPage, Snippet Genre
├── events/                  # EventListingPage, EventPage, Snippet Venue
├── booking/                 # BookingFormPage, logic, Celery tasks
├── navigation/              # Menu Snippet + Orderable
├── frontend/                # React app (da template-strutturale)
├── templates/               # Django/Wagtail templates (se SSR)
├── static/
└── media/
```

Mappa Fasi e Task

FASE 0 — Scaffolding

#	Task	Agente	Dipendenze
01	Setup progetto Django/Wagtail	Backend	Nessuna

FASE 1 — Data Models (Backend)

#	Task	Agente	Dipendenze
02	Snippet models (Genre, Venue, Promoter)	Backend	01
03	ArtistListingPage + ArtistPage	Backend	01, 02
04	EventListingPage + EventPage	Backend	01, 02
05	StreamField blocks custom	Backend	01
06	EPK/Media model + Collection permissions	Backend	01, 03

FASE 2 — Business Logic (Backend)

#	Task	Agente	Dipendenze
07	Booking form context-aware	Backend	03, 04
08	Celery task archiviazione eventi	Backend	04
09	Management command import CSV	Backend	02, 03
27	Permessi, Workflow e Isolamento Per-Band	Backend	01, 03, 04, 06

FASE 3 — API, i18n, Navigation

#	Task	Agente	Dipendenze
10	Wagtail API v2 configuration	Backend	03, 04
11	wagtail-localize setup	Backend	03, 04

#	Task	Agente	Dipendenze
12	Sistema menu dinamico (Snippet)	Backend	01

FASE 4 — Frontend Components

#	Task	Agente	Dipendenze
13	Setup frontend da template-strutturale	Frontend	01
14	Layout + Navigazione	Frontend	12, 13
15	Hero + HomePage	Frontend	10, 13
16	ArtistGrid + ArtistCard + ArtistDetail	Frontend	10, 13
17	EventListing + EventDetail	Frontend	10, 13
18	Booking Form component	Frontend	07, 10, 13
19	Trova la tua Band — BandFinder (Gemini)	Frontend	10, 13

FASE 5 — SEO, A11y, Performance

#	Task	Agente	Dipendenze
20	JSON-LD + Meta tags SEO	Full-stack	03, 04, 10
21	Accessibilità WCAG 2.1 AA	Frontend	14-19
22	Caching + ottimizzazione immagini	Backend	03, 04
23	Search backend (PostgreSQL/ES)	Backend	03, 04

FASE 6 — Test & Deploy

#	Task	Agente	Dipendenze
24	Test suite backend (Pytest)	Backend	02-09
25	Test suite frontend (Vitest)	Frontend	14-19
26	Configurazione deploy (Docker + CI)	DevOps	Tutti

Ambienti di Deploy

DEMO (locale / CI)

Parametro	Valore
Runtime	Docker Compose (dev)
DB	PostgreSQL (container)
Broker	Redis (container)
Web server	Django `runserver` / Gunicorn (opzionale)
Frontend	Vite dev server (porta 3000)
Dominio	`localhost`
Scopo	Sviluppo locale, CI pipeline, demo pre-rilascio
Task riferimento	Task 26 — sezione Docker Compose

LIVE (server remoto)

Parametro	Valore
Accesso	SSH/SCP — porta **100**
App server	Gunicorn (socket/systemd)
Reverse proxy	Nginx
DB	PostgreSQL (locale o managed)
Broker	Redis
Dominio definitivo	`magixpromotion.com` + `www.magixpromotion.com`
Alias staging	`new.magixpromotion.com` → punta allo stesso vhost
SSL	Let's Encrypt (Certbot)
Scopo	Produzione
Task riferimento	Task 26 — sezione Nginx + Gunicorn

Strategia domini:
Il vhost Nginx è configurato per il dominio definitivo `magixpromotion.com`.
`new.magixpromotion.com` è un `server_name` alias sullo stesso `server` block, usato nella fase di test/go-live.
Quando il sito è definitivamente LIVE, il record DNS di `magixpromotion.com` viene puntato al server; `new.` resta come alias e può essere rimosso in seguito.

```
# Esempio server_name (Nginx)
server_name magixpromotion.com www.magixpromotion.com new.magixpromotion.com;
```

Dati di Riferimento

- **CSV con 24 band:** `dati-band-Magixpromotion.csv`
- **Categorie:** Dance Show Band, Beat Show Party Band, Glam Rock Night, Rock Band, Folk Band, Tributo Italiano, Tributo Internazionale, Dee-Jay
- **Territorio:** Base operativa Nord Italia (Piemonte, Lombardia, Emilia), ma i venue e gli eventi possono essere internazionali
- **Sede:** Via dello Scabiolo, 15067 Novi Ligure (AL), Piemonte, Italia — Tel: +39 335 523 0855
- **Lingue:** IT (primaria), EN (secondaria)
- **Mappe:** OpenStreetMap + Nominatim (NO Google Maps API)