# **Content Africa — Page-by-page replication & developer guide**

**Purpose:** a single, developer-friendly document that explains how to replicate Content Africa (structured the same way as the UseAllies guide), reconstruct the sitemap, and implement the user flows, data model, AI features, and deployment checklist.

⚠️ This guide is implementation-focused: components, endpoints, DB schema, UI blocks, and testing suggestions. Use it as a living spec to hand straight to frontend/backend engineers.

**How to use this doc**

1. Read the **Sitemap & user flows** to understand the core product and personas (Brands, Creators, Admins, Community members).
2. Follow the **Page-by-page** section for UI composition and copy placement.
3. Use **Data model** + **API endpoints** for backend implementation.
4. Use the **Dev tasks & checklist** to create tickets in your tracker (Jira, Linear, GitHub Projects).

**Quick summary of the product being replicated**

* A lightweight SaaS marketplace that helps **brands find creators** and helps **creators monetize** via UGC, brand deals, courses and communities.
* Core marketing pages: Home (product pitch + AI prompt), Resources (UGC & Brand Deals), Communities (coming soon), Pricing (free for creators; transaction fee), Contact, Sign up / Login.
* Built around two main user personas: **Brand** (search & hire creators) and **Creator** (signup, profile, monetization options).

## 

## **Sitemap (recommended structure)**

/

/login

/signup

/pricing

/resources

/ugc-brand-deals (resources detail)

/communities (listing + community detail; current site shows "Coming Soon")

/join-community (lead capture form)

/contact

/privacy

/dashboard (creator + brand dashboards — protected)

/dashboard/creator/profile

/dashboard/brand/search

/dashboard/brand/briefs

/api/\* (backend endpoints)

/sitemap.xml

/robots.txt

***Note:*** *keep public content under top-level pages and put interactive features behind authenticated /dashboard routes.*

## **Page-by-page breakdown (what to build — block-level)**

### **1) Home (/)**

**Purpose:** hero to convert both brands and creators. Two primary CTAs: See All (browse creators / explore) and Sign up.

**Blocks (in render order):**

* Header: Logo (SVG), primary nav items (Communities, Pricing, Resources), Auth links (Login / Sign up).
* Hero: one-line value prop (AI-powered creator search + creators monetize). Support subtext. Primary CTA (Search creators / Get started). Secondary CTA (See All / Learn More).
* Metrics strip: quick trust numbers (e.g. creators onboarded, amount paid) — small counters.
* Feature cards: "UGC & Brand Deals", "Communities", "A.I. matching" — 3 columns.
* How it works: 3-step flow (Prompt → Match → Collaborate).
* Testimonials / social proof (if available or placeholder).
* Footer: Explore, Legal, Contact, social links.

**Components:**

* HeroPrompt — AI prompt input + sample prompts + microcopy.
* MetricCounter — animated numbers.
* FeatureCard — icon + headline + 1–2 lines.

**Data & API:**

* Public: marketing copy from CMS.
* Interactive: the hero prompt should POST /api/search-creators and return a paginated result set.

**2) Pricing (/pricing)**

**Purpose:** explain pricing model: free for creators; 4% transaction fee.

**Blocks:**

* Simple headline + short explanation.
* Fees breakdown: creators free; platform takes 4% on transactions (show example calculations).
* CTA to signup / FAQ link.

**Data & API:** static.

**3) Resources (/resources)**

**Purpose:** hub for guides and resource pages (UGC & Brand Deals is listed).

**Blocks:**

* Intro line: "Access amazing resources..."
* Resource tiles linking to /ugc-brand-deals and future assets.

**4) UGC & Brand Deals (/ugc-brand-deals or nested resource)**

**Purpose:** explain the main product offering: connect brands with UGC creators and brand deals.

**Blocks:**

* Hero with benefit bullets.
* How creators monetize.
* CTA: join / learn more.

**5) Communities (/communities) & Join Community (/join-community)**

**Purpose:** community feature (marketing page + lead-capture). The live site shows "Coming Soon" and provides a join form.

**Join form fields:** name, email, phone, country selector, CTA Submit.

**Backend:** capture leads to CRM (e.g., HubSpot, Airtable) or save to your DB and send confirmation email.

**6) Contact (/contact)**

**Purpose:** provide contact emails and social handles.

**Content:** hello@contentafrica.com and support@contentafrica.com (site contact emails).

**Backend:** make email addresses clickable mailto: links; optionally a contact form that sends to support inbox via transactional email provider (SendGrid, Postmark).

**7) Authentication: /login & /signup**

**Purpose:** support Google sign-in and email/password. Basic signup includes username, email, phone and country select.

**Flow:**

* Signup → onboarding: choose role (Brand / Creator / Coach) → profile creation (Creator: portfolio links & social handles; Brand: company info & payment setup).

**Security:** password hashing (bcrypt/argon2), rate-limit endpoints, email verification.

**Dashboard Expansion Overview**

The new User Dashboard extends the Content Africa platform to support deep analytics, campaign management, payments, and AI-driven discovery. It’s designed for three user roles:

* Creator — manage profile, monitor analytics, and track brand deals.
* Brand / Agency — discover creators, manage campaigns, track KPIs, and handle payments.
* Admin — oversee moderation, platform metrics, user verification, and transactions.

Dashboard Sitemap

/dashboard

├── overview

├── audience-insights

├── content-performance

├── campaigns

├── discovery

├── messages

├── payments

├── reports

├── settings

**Dashboard Modules — Detailed Developer Specification**

### 1. Overview / Snapshot

Purpose: Single-screen performance summary.

Widgets:

* Metric Tiles: Followers, Following, Engagement Rate, Earned Media Value, Avg Interactions, Notable Followers %.
* Follower Quality Gauge: “Nice vs Doubtful” followers donut chart.
* Activity Split Donut: Views / Likes / Comments.
* Recent Activity Timeline: 30-day performance chart.
* Alerts & Suggestions: System-generated insights or warnings.

Components:

* <MetricTile />, <DonutGauge />, <Sparkline />, <AlertList />

API: GET /api/dashboard/overview

Data: Aggregated analytics data from social APIs and internal metrics.

2. Audience Insights

Purpose: Deep-dive into followers’ demographics and authenticity.

Widgets:

* Geographic heatmap (Country/City)
* Age & Gender breakdown
* Language distribution
* Interest clusters and brand affinity
* Fake follower score and audit logs

API: GET /api/creators/:id/audience and POST /api/analytics/run-fraud-scan

Data Model:

AudienceProfile(id, creator\_id, country\_breakdown JSON, age\_breakdown JSON,

gender\_breakdown JSON, languages JSON, fake\_follower\_score FLOAT, last\_updated\_at TIMESTAMP)

3. Content Performance

Purpose: Measure content reach, engagement, and performance trends.

Widgets:

* Table: post thumbnails, metrics (likes, comments, reach, views)
* Top-performing posts carousel
* Reels/Stories performance line chart
* Paid post performance stats

API:

* GET /api/creators/:id/content
* GET /api/creators/:id/content/:post\_id/insights

Data Model:

ContentMetrics(id, post\_id, creator\_id, impressions INT, views INT,

likes INT, comments INT, engagement\_rate FLOAT, date\_collected TIMESTAMP)

4. Campaigns & Briefs (Brand Side)

Purpose: Manage influencer campaigns from brief creation to payments.

Features:

* Create and edit briefs
* AI-assisted creator matching
* Invite creators and manage offers
* Track campaign performance metrics
* Manage payments and deliverables

API:

* POST /api/briefs
* POST /api/briefs/:id/invite
* GET /api/campaigns/:id/metrics

Data Model:

Briefs(id, brand\_id, title, description, budget, deliverables, deadline, status)

Offers(id, brief\_id, creator\_id, amount, status)

CampaignMetrics(id, campaign\_id, impressions INT, clicks INT, conversions INT, spend FLOAT)

**5. Discovery / AI Search**

Purpose: Enable brands to find creators using AI prompts or filters.

UI: Prompt input, filters (followers, engagement, country, interests, fake follower max %).

API: POST /api/search-creators

Implementation: Vector embeddings with reranking and caching popular queries.

6. Messaging & Contracts

Purpose: Allow direct brand-creator communication.

Features:

* Real-time chat
* File attachments (briefs, NDAs)
* Contract templates and signing

API: POST /api/messages, GET /api/conversations/:id

### 7. Payments & Payouts

Purpose: Manage platform payments securely via Stripe Connect.

Features:

* Creator earnings & payout requests
* Brand invoice management
* Platform fee (4%) auto-calculation

API:

* POST /api/payments/checkout
* GET /api/payments/:id/status
* POST /api/payments/webhook

Data Model:

Transactions(id, offer\_id, amount DECIMAL, platform\_fee DECIMAL, payout\_status VARCHAR, created\_at TIMESTAMP)

8. Reports & Exports

Purpose: Export analytics and performance data.

Features:

* PDF/CSV export
* Weekly performance digest

API: GET /api/reports/:type

**UI & Frontend Guidelines**

Components:

* Reusable charts (<DonutChart />, <BarChart />, <LineChart />)
* Responsive layouts using Tailwind CSS grids
* Export and filter buttons on every analytics page

Design tokens:

* Font: Inter / Poppins
* Radius: 8px–16px
* Shadows: soft neutral tone
* Colors: use semantic palette (primary, success, warning)

**Backend Architecture**

Stack Recommendation:

* Node.js + Express / Django REST
* PostgreSQL for core data
* Redis for caching and job queues
* Vector DB (Pinecone / Weaviate) for AI search
* S3-compatible storage for media

**Testing & QA**

* Unit tests for analytics endpoints
* Integration tests for fraud scan and payments
* E2E tests: Login → Overview → Analytics → Export → Payment

**Deployment Plan for Dashboard**

Phase 1: Overview + Content Metrics + Campaign CRUD + Payments skeleton

Phase 2: Audience Insights + Fraud Detection + Messaging + Exports

Phase 3: Full AI discovery, automated insights, admin dashboards

API Summary

GET /api/dashboard/overview

GET /api/creators/:id/audience

POST /api/analytics/run-fraud-scan

GET /api/creators/:id/content

GET /api/creators/:id/content/:post\_id/insights

POST /api/search-creators

POST /api/briefs

POST /api/briefs/:id/invite

GET /api/campaigns/:id/metrics

POST /api/payments/checkout

POST /api/payments/webhook

GET /api/reports/:type

Developer Checklist for Dashboard

* Implement Overview dashboard widgets
* Connect real analytics data to charts
* Add AudienceInsights and FraudScan API
* Create AI Search with vector ranking
* Integrate payments & Stripe Connect
* Build Campaign & Messaging modules
* Add PDF/CSV export utilities

**Visual & UI details**

**Design tokens** (suggest):

* Font: Inter / Poppins for headings
* Spacing: 8px scale
* Rounded corners: 8px to 16px
* Buttons: primary CTA with moderate radius, drop shadow

**Tailwind utility mapping**: use classes like container mx-auto px-6, grid grid-cols-1 md:grid-cols-3 gap-8, text-xl font-semibold.

**Accessibility:** semantic headings, aria-label on form fields, focus-visible outline, color contrast ≥ 4.5:1 for body text.

**Data model (schema) — core tables/collections**

### **Users**

* id, email, username, password\_hash, role (creator|brand|coach|admin), phone, country, created\_at, last\_login

### **Creators (profile)**

* id, user\_id, display\_name, bio, skills (array), social\_links (IG, TikTok, YouTube), portfolio\_urls, sample\_media (array of media objects), rates, metrics

### **Brands**

* id, user\_id, company\_name, company\_website, billing\_info

### **Briefs / JobPosts**

* id, brand\_id, title, description, budget, deliverables, deadline, status

### **Offers / Contracts**

* id, brief\_id, creator\_id, brand\_id, amount, platform\_fee, status

### **Transactions**

* id, offer\_id, stripe\_payment\_id, amount, platform\_fee, payout\_status

### **Communities / Courses / Posts**

* id, title, slug, content, author\_id, members, posts

### **Messages & Notifications**

* message\_id, conversation\_id, from\_user, to\_user, body, read\_at, created\_at

## ***Backend endpoints (examples)***

*POST /api/auth/signup*

*POST /api/auth/login*

*POST /api/auth/oauth/google*

*GET /api/creators?query=&page=*

*POST /api/search-creators // accepts AI prompt + filters*

*GET /api/creator/:id*

*POST /api/briefs // brand creates brief*

*POST /api/offers // creator responds / brand makes offer*

*POST /api/payments/checkout // create Stripe PaymentIntent*

*GET /api/dashboard/stats*

*POST /api/join-community // lead capture*

## **AI "prompt → find creators" feature**

**High-level approach:**

1. Collect creator profiles and index them as vectors (using features like skills, text bio, categories, social metrics).
2. Accept a free-text prompt from the brand via /api/search-creators.
3. Convert the prompt to an embedding and run a nearest-neighbor search against the creator vector store.
4. Rerank results and return paginated results with match confidence.

**Payments & fees**

**Recommended:** Stripe Connect with platform fee. Platform takes 4% of transactions (visible on pricing page). Implement automatic payouts to creators; record platform\_fee on transactions.

**Flow:**

* Brand pays via Stripe Checkout.
* Platform takes 4% via application\_fee\_amount.
* Webhook listener for payment success, refunds.

**Admin & moderation**

**Admin tasks:**

* Approve/reject creators.
* Issue refunds, handle disputes.
* Manage featured creators & campaigns.

**Admin UI:** a secure dashboard with user management, transaction logs, content moderation queue.

**SEO, sitemap & analytics**

* Generate sitemap.xml dynamically.
* Add robots.txt.
* Use structured data.
* Integrate analytics (GA4).

**Testing strategy**

* Unit tests for API logic.
* E2E tests for flows (signup → onboarding → brief → search → offer → payment).
* Accessibility tests (axe-core).

**Deployment & infratructures**

**Frontend:** Next.js (React) deployed to Vercel.  
 **Backend:** Node/Express or Django.  
 **DB:** PostgreSQL + Redis.  
 **Vector store:** Pinecone / Weaviate.  
 **Storage:** S3-compatible.  
 **Auth:** NextAuth.js / Auth0.

**Dev tasks / Checklist**

**MVP tasks:**

* Landing page: hero, metrics, CTA
* Auth: email/password + Google OAuth
* Creator profile model & CRUD
* Brand brief model & CRUD
* POST /api/search-creators (AI prompt)
* Stripe Connect integration
* Join-community form
* Pricing & Resources pages
* Contact page + mailer setup
* Admin dashboard (basic)
* Sitemap.xml and robots.txt
* Analytics + error monitoring

**Polish / v2 tasks:**

* Full vector-search ranking for AI prompt
* Creator onboarding & KYC
* Advanced community features
* Messaging UI with real-time updates
* Automated email flows

*End of Content Africa Developer Guide.*