Muhammad Asad · AI Implementation & Growth

Case Study · N° 08

PRODUCT / FRONTEND + DB

A directory where money can't buy rank

Designed the 16-table database and built the directory that serves it: 575 products scored by a transparent model where sponsorship never touches ranking, on a ~2,000-page Next.js site.

Cognilium directory venture (manager: Ali Ahmed) · my work: the database + directory engineering

DISCIPLINE	Product / Frontend + DB
STACK	16-table schema (Postgres canonical / SQLite runtime) • better-sqlite3 • Next.js 14 • SSG/ISR • field-level provenance • build-time claim guardrails

MY ROLE

A Cognilium venture (manager and product/editorial lead: Ali Ahmed). My role is the engineering: the database design, the data pipeline, and the directory app, not the venture, the scoring brand, or the editorial.

575	16	0	~2,000
PRODUCTS STRUCTURED	TABLES, 62-COL HUB	SPONSORED IN RANKING	PAGES GENERATED

THE PROBLEM

The directory needed a structured, trustworthy data model for hundreds of legal-tech products and a fast site to serve it, with rankings that money could not distort.

WHAT I DID

- Designed a 16-table normalized schema around a 62-column products hub: vendors, categories, buyer-types, plans, integrations, plus the join tables. Postgres canonical, SQLite at runtime via better-sqlite3.
- Made the ranking unbuyable: the is_sponsored flag lives on a product but appears in zero ranking queries, so paid placement never touches sort order, and the published 0-100 score is the exact sum of its declared sub-scores (capability, transparency, validation, completeness, maturity).
- Put a provenance receipt on every field (source, method, verified_at, status) and shipped FTC-grade review rails (identity tier, disclosure, incentivization, salted fraud-IP hash) before the first review exists.
- Built the directory as a ~2,000-page Next.js app (43 page types, 22 components, SSG/ISR) reading the database directly, with build-time guardrails that fail the build if any published figure stops matching the data.
- Ingested 575 products and 508 vendors across 10 categories from the competitive scrape, normalized and deduped.

THE RESULT

A 575-product directory with a transparent, unbuyable scoring model and a provenance trail on every field. Pre-launch; the database structure is shown below.

THE JUDGMENT CALL · WHAT THE AI COULDN'T DO

The obvious schema lets money tilt the rankings: add a sponsored flag, let it weight the sort. I put is_sponsored on the product but kept it out of every ranking query, and made the published score the exact sum of its declared parts, so paid placement can't move a rank and the number can't be fudged. On an unlaunched directory, deciding money never touches sort order is a trust decision encoded in the schema, not a prompt.

PROOF

Status: Pre-launch (Cognilium venture), no public URL yet.

On request: The schema, the scoring model, and a walk-through of the app.

SEE IT

Legal-tech directory · database structure

16 tables · products = 62 columns · 575 products · 508 vendors

products 575

◆ id	tech_stack
slug	key_features
name	core_functionalities
→ vendor_id	lifecycle_stages
tagline	top_use_cases
overview	best_known_for
logo_url	critical_opinions
website	*+45 more*
target_users	

◆ primary key → foreign key

ENTITIES

vendors 508
- ◆ id
- slug
- name
- country
- founded_year
- founders
- website
- email
- phone
- social_media

categories 10
- ◆ id
- slug
- name
- description
- intro_content
- sali_iri
- meta_title
- meta_description

buyer_types 7
- ◆ id
- slug
- name
- description

plans 3
- ◆ id
- slug
- name
- price_year
- → stripe_price_id
- description

integrations 33
- ◆ id

users ACCOUNTS
- ◆ id
- name
- created_at

JOINS (M:N)

product_categories M:N
- → product_id
- → category_id

product_buyer_types M:N
- → product_id
- → buyer_type_id

product_integrations M:N
- → product_id
- → integration_id

TRUST & PROVENANCE

field_provenance PER-FIELD
- ◆ id
- → product_id
- field
- method
- source_label
- source_url
- extracted_at
- verified_at
- confidence
- value_checked

+1 more

sources PER-PRODUCT
- ◆ id
- → product_id
- type

product_intel CRAWL INTEL
- ◆ id
- → product_id
- kind
- note
- source_url
- captured_at

reviews FTC RAILS
- ◆ id
- → product_id
- → user_id
- rating
- title
- reviewer_identity_tier
- verified_use
- incentivized
- incentive_desc
- material_connection

+6 more

CAPTURE

leads CAPTURE
- ◆ id
- → product_id
- name
- status

events COUNTERS
- ◆ id
- → product_id
- type

Unbuyable ranking: is_sponsored lives on products but appears in zero ranking queries, money never touches sort order. Every field carries a provenance receipt (source, method, verified_at, status). Reviews ship with FTC-grade identity and disclosure rails before review #1 exists.

The database structure: a 16-table schema around a 62-column products hub, with a provenance receipt on every field and FTC-grade review rails. Sponsorship is stored but never ranked.