

The AI-Readiness Stack

Modernizing the CMS for an AI-First Future · Nardeep Singh, MSITM & Jovana Trudell · Renown Health



MACHINE-READABLE

SCORE YOUR OWN SITE 1-5 ON EACH LAYER

WHAT A 5 LOOKS LIKE

5

Retrievability

Can crawlers and agents reach the answer without executing JavaScript?

The answer is in the raw HTML, and another system can request it through an API.

1

2

3

4

5

4

Reuse

Does one fact live in one place and appear on many surfaces?

A phone number or provider name is edited once and updates everywhere it appears.

1

2

3

4

5

3

Governance

Who owns accuracy — and is the review trail recorded?

A named approver per content type, review date stored as a field, full audit trail.

1

2

3

4

5

2

Structured data

Is schema generated from the model, or hand-written per page?

JSON-LD is emitted from the content model automatically, not keyed page by page.

1

2

3

4

5

1

Content model

Is content typed and fielded, or freeform rich text?

Every content type has named fields. No page is one rich-text blob.

1

2

3

4

5

FOUNDATION

Total ____ / 25

Under 15, your next project is structure — not an AI tool. Buying a tool on top of a 12 produces drafts nobody can publish.

CHECK ONE · VIEW THE SOURCE

Open your top service line page and view the page source — Windows: Ctrl + U (Edge or Chrome) · Mac: Cmd + Option + U (Safari: enable Develop first). Search for the first sentence of your main answer. If it isn't there, it renders in JavaScript.

CHECK TWO · READ YOUR ROBOTS.TXT

Go to yoursite.com/robots.txt. Look for GPTBot, ClaudeBot, PerplexityBot, Google-Extended. The real question is whether anyone decided it on purpose.

CHECK THREE · ASK ONE QUESTION

Ask your web team: can another system get this content through an API, or would it have to scrape the page?

AI reads structure, and a page is a blob. You cannot ask a blob what a service line's conditions are — so the tool was never the problem.

THE ONE-SENTENCE
VERSION

Governance, prompts, and one next step



YOUR GOVERNANCE MODEL, IN ONE TABLE

The tier decides what AI may do and who signs off — not whether AI is allowed at all. Fill in your own row before you ask legal for anything.

TIER	CONTENT TYPE	WHAT AI MAY DO	WHO APPROVES
1. Clinical	Conditions, treatments, symptoms	Structure and readability only. No clinical claims.	Clinical reviewer + marketing
2. Marketing	Service lines, provider bios, campaigns	Draft and edit	Marketing + service line owner
3. Operational	Alt text, tags, metadata, internal links	Act, with sampled QA	Content ops spot-check

Yours:

The guardrails that get you to yes: no PHI in prompts · no AI-generated clinical claims · every change staged as a draft with a named approver · the agent inherits the requester’s permissions · full audit trail · legal reviews vendor terms before anything is built.

STARTER PROMPTS — OPERATIONS FIRST, NOT WRITING

These read your content and touch no clinical claims. Swap in your own document type names.

- **Missing fields:** find all documents of type provider where a field is empty, grouped by department.
- **Stale content:** every page whose last review date is more than 18 months old, oldest first.
- **Broken references:** documents pointing at a provider or location that no longer exists.
- **Near-duplicates:** pages covering the same condition, with title, URL, and word count.
- **Metadata:** pages missing a meta description or over 160 characters — propose one, staged as a draft.

IF YOU DO ONE THING

- **List your content types.**
If you can't name them, that's the first project — not the AI purchase.
- **Open your review log.**
Every edit a reviewer makes before approving is a recorded judgment about what good looks like.
- **Interrogate the next purchase.**
Ask what problem the tool solves, and whether the data it needs exists yet.

Questions after today? Reach out on LinkedIn.

Healthcare marketing is a small community, and none of us are competitors on this. If you get stuck, or you try something and it works, tell us — we'll answer.



Nardeep Singh, MSITM
SCAN TO CONNECT