Specimen Audit Report

Website Analysis & Machine Readiness - anonymised specimen

Tom Cranstoun

MACHINE EXPERIENCE Compatible

tagged · provenance · attested

Scan to learn more

This PDF is MX Compatible. It carries machine-readable structure, AI-governance provenance, and the metadata that lets a regulator, an AI agent, or a screen reader walk the document with the same confidence a sighted reader does. Scan the QR code or visit `https://mx.allabout.network/learn/mx-for-pdfs.html` to read what that means.

ISO 14289-1 Level 2 · pdfuaid:Part=1 · Provenance: EU AI Act · UK ICO · NIST AI RMF · EAA Directive 2019/882

Prepared by: Tom Cranstoun | CogNovaMX Ltd
Contact: info@cognovamx.com | https://allabout.network
Date: 12 July 2026
Report ID: specimen-media-WEB-AUDIT-20260712

> **Specimen report.** This is an anonymised specimen of a real audit delivered by the Web Audit Suite. The client's name, domain, and page URLs have been replaced; every finding, score, table, and recommendation is exactly as the pipeline produced it. It shows what you receive when we audit your site.

Executive Verdict

> **Bottom line.** Specimen Media Co runs on Unknown Platform and scores 58/100 on automated accessibility checks and 80/100 for SEO. It sits at MX Readiness Level 1 (Discoverable). The single most important next step is to resolve the 14 distinct WCAG issue types flagged below (a Priority 1 compliance obligation).

Top risks

1. **Commerce Visibility (10/100)** - Needs Improvement.
2. **Discovery Readiness (40/100)** - Could Be Better.
3. **MX Stack Completeness (61/100)** - Good.

Executive Summary

Table 1

Executive Summary

	Score	Verdict
Performance	**85**/100	##############--- Excellent
Accessibility	**58**/100	##########-------- Good
SEO	**80**/100	##############---- Excellent
Served-HTML Structure	**87**/100	################-- Excellent
MX Stack Completeness	**61**/100	##########------- Good
Agent Readability	**89**/100	################-- Excellent
Pipeline Survivability	**77**/100	##############---- Excellent
Machine Processing Speed	**103** ms/page	#################- Machine-Ready

The three machine metrics measure different things. ***Served-HTML Structure*** *is the semantic markup an agent reads before JavaScript runs;* ***Agent Readability*** *is how easily the content can be quoted once reached;* ***Pipeline Survivability*** *is whether a page survives an agent's fetch and ingest. A site can score low on one and high on another.*

Agent Readability was adjusted down by 4 points for site-wide gaps a machine cannot work around:

- **Generic containers without landmark roles on most pages** (-4): 8 of 12 pages rely on generic containers without landmark roles - expected from a component framework; adding landmark roles at the component level clears it

Across the audited set, the site appears to be an editorial or media publisher.

Across the audited set, Schema.org types indicate a content or editorial context.

Specimen Media Co runs on Unknown Platform. Across the audited set, Specimen Media Co scores 58/100 for accessibility and 80/100 for SEO.

The headline opportunity is addressing the 14 distinct WCAG AA issue types that generate 566 raw instances. One fix per category clears many instances, making remediation efficient. This compliance improvement also boosts machine comprehension by ensuring content is accessible to all users.

The audited set runs on an unknown platform, which constrains the specificity of rendering recommendations. Score thresholds for Governed are met (MSC 61, SDQ 66, DR 40), but no MX-namespaced governance metadata was detected; adding mx:status, mx:contentType, mx:audience, canonicalUri and provenance markers to a published page unlocks the next MX Readiness Level.

Change Since Our 1 July 2026 Audit

We last audited specimen-media.example on 1 July 2026. The table compares that audit with the current one across the headline measures. Some scores declined and the rest held steady; the table shows each change.

Table 2

Change Since Our 1 July 2026 Audit

Measure	1 July 2026	12 July 2026	Change
Performance	97	85	-12 (declined)
Accessibility	59	58	-1 (declined)
SEO	80	80	No change
WCAG AA issues	523	566	+43 (declined)

We include this comparison because it is what continuous monitoring delivers: each re-audit shows what moved and what held, so open items stay visible until they are closed.

Balanced Scorecard

Human Experience

Human visitors experience Excellent performance (avg 595 ms load time), Good accessibility across 11 pages, and Excellent search visibility.

Table 3

Human Experience

Dimension	Score	Band	vs Peers
Performance	85/100	Excellent	A (median)
Accessibility	58/100	Good	A (median)
SEO (content pages)	80/100	Excellent	A (median)

Machine Experience

Machines experience Excellent HTML structure at first fetch, Could Be Better discovery readiness (40/100), Good structured data quality (66/100), and Excellent agent readability.

Table 4

Machine Experience

Dimension	Score	Band	vs Peers
Served-HTML Structure	87/100	Excellent	A (median)
Discovery Readiness	40/100	Could Be Better	C (median)
Structured Data Quality	66/100	Good	B (median)
MX Stack Completeness	61/100	Good	B (median)
Pipeline Survivability	77/100	Excellent	A (median)
Security headers	-	-	-
Machine Processing Speed	103 ms/page	Machine-Ready	-

Benchmark median drawn from a curated audit dataset.

Reading the Report: Two Budget Lenses

The findings below are tagged with one of three buckets so different budget owners can navigate to the work that matters to them:

- **Compliance Risk**: accessibility (WCAG 2.1 AA), duplicate IDs, forms, and semantic structure. These are inclusion and legal-exposure issues; the budget owner is typically legal, HR, or an accessibility lead.
- **Cross-cutting Foundations**: performance and SEO. These affect both human visitors and AI agents; the budget owner is usually digital operations or a foundation engineering team.
- **Machine Readability Opportunity**: discovery readiness, metadata stack, llms.txt, structured data, agent cards, and pipeline survivability. These determine whether the site lands in agent answers; the budget owner is typically the CMO or head of digital.

Every priority block in the Findings section carries a **Bucket:** label matching one of the three above. The at-a-glance table sorts findings into bucket order so each budget owner can read straight to their own list.

MX Readiness Level

Table 5

MX Readiness Level

	Level	Name	Publisher Capability	Agent Outcome	
	0	Not Ready	Auto-generated boilerplate	Agents guess, hallucinate	
→	1	Discoverable	Deliberate metadata, publisher identified	Agents can discover	←
	2	Governed	Full MX fields, governance, provenance	Machines have structured governance context	
	3	Comparable / Attested	Cryptographic attestation, cross-source verifiable	Agents can search, compare, recommend	
	4	Transactable	Registered, priced, SLA-backed, alive	Agents can understand pricing and transact	
	5	Purchase-confident	Third-party audited, fiduciary-grade	Agents can guarantee accuracy at purchase	

Current Level: 1: Discoverable

Evidence: MX Stack Completeness 61/100 Core metadata layers are in place; governance fields and provenance declarations are missing. | Structured Data Quality 66/100 Schema.org data is present and valid; recommended properties and entity links are sparse. | Discovery Readiness 40/100 Machines can find the site but lack structured signals about its purpose and content policy. | Consistency 64% Most metadata patterns are consistent; a few gaps mean some pages deliver weaker signals than others.

To reach the next level: Score thresholds for Governed are met, but no MX-namespaced governance metadata was detected.ected on the audited pages. Add MX governance fields (mx:status, mx:contentType, mx:audience, canonicalUri, and provenance markers) to at least one published page to unlock Governed level. Without those fields a machine can discover the page but lacks the governance context for accurate comprehension.

What’s Working Well

We find SEO performance and structured-data quality across the audited set, giving a clear starting point for the improvements ahead.

Table 6

What’s Working Well

Dimension	Score	Highlights
Performance	Excellent	Excellent - 595ms average load time
SEO (content pages)	80	Excellent - titles, meta descriptions, canonical URLs in place
Security	1/5	1/5 headers present (HSTS, CSP, X-Frame-Options, X-Content-Type-Options absent); 0 of 12 URLs carry all five
Structured Data	66	Good - JSON-LD on every page with valid Schema.org vocabulary
Heading Quality	87	Excellent - single H1 on every page
Consistency	64%	64% - same metadata patterns across every page
Agent access	8/8	every tested agent receives HTTP 200

Positive patterns observed:

- All 8 tested AI agents can fetch the site: ClaudeBot (Anthropic), GPTBot (OpenAI), ChatGPT-User (OpenAI), PerplexityBot, GoogleOther (Google AI), Google-Extended, CCBot (Common Crawl), Plain request (no UA) all return HTTP 200 at inference time.
- JSON-LD is present in the served HTML of every page: every agent that fetches the raw HTML gets the structured data.

Findings

At a Glance

The table below is the prioritised action list for this audit. Each row names a finding, its compliance-risk bucket, and the effort to fix it. The numbered blocks below the table expand each finding with specific guidance.

We identified 12 finding(s) on the audited set, ordered by regulatory exposure first and then by priority within each category.

Table 7

At a Glance

#	Finding	Bucket	Priority	Effort	Impact
1	Insufficient Colour Contrast, WCAG 1.4.3	Compliance Risk	High	Medium	low-vision users may miss or misread affected content
2	Non-text Content Missing Text Alternatives, WCAG 1.1.1	Compliance Risk	High	Low	screen reader users may miss or misread affected content
3	Interactive Elements Missing Name, Role, or Value, WCAG 4.1.2	Compliance Risk	High	Medium	screen reader users may miss or misread affected content
4	Info and Relationships Not Programmatically Determined, WCAG 1.3.1	Compliance Risk	High	Medium	screen reader users may miss or misread affected content
5	Accessibility Issue, WCAG 3.2.2	Compliance Risk	High	Medium	all users may miss or misread affected content
6	No Bypass Mechanism for Repeated Blocks, WCAG 2.4.1	Compliance Risk	Medium	Low	sighted keyboard users may miss or misread affected content
7	Heading Hierarchy Skips Levels	Compliance Risk	Medium	Low	screen-reader and machine outline-builders may misread the page structure
8	Main Landmark Absent	Compliance Risk	Medium	Low	agents and assistive technology may not locate the primary content
9	Semantic Structure 40/100	Compliance Risk	Medium	Medium	machines lose structural context and infer page regions by position
10	Security headers absent: HSTS, CSP, X-Frame-Options, X-Content-Type-Options	Cross-cutting	Medium	Low	Missing security headers increase exposure to content injection and clickjacking
11	Structured Data Property Gaps	Machine Readability Opportunity	Medium	Medium	machines may extract these entities incompletely or skip them
12	Schema.org coverage is partial: Decoration (SDQ 66/100)	Machine Readability Opportunity	Medium	Medium	Agents can partially parse structured facts but key properties may be missing

Priority 1: Insufficient Colour Contrast, WCAG 1.4.3

Bucket: Compliance Risk

Finding: Text and its background do not meet the minimum contrast ratio, leaving the content hard to read for low-vision users and in bright-light conditions. This pattern appears 431 time(s) across the audited set, affecting low-vision users.

What to change and why:

- Raise the contrast ratio of the flagged text and its background to at least 4.5:1 for normal text (3:1 for large text). This satisfies WCAG 1.4.3 and keeps the text readable for low-vision users and in bright-light conditions.
- Fix the values in the design tokens or theme stylesheet once so the change propagates wherever the colour pair is reused, rather than patching individual pages.

Effort: Medium

Priority 2: Non-text Content Missing Text Alternatives, WCAG 1.1.1

Bucket: Compliance Risk

Finding: Images on the audited set carry no text alternative, so their content is unavailable to screen-reader users and to machines reading the page. This pattern appears 39 time(s) across the audited set, affecting screen reader users.

What to change and why:

- Add descriptive alt text to every informative image; mark purely decorative images with empty alt (alt="") so assistive technology skips them. This satisfies WCAG 1.1.1 and gives screen-reader users the same information sighted users get.
- Where an image is the only content of a link, the alt text must describe the link destination, not the picture, so keyboard and screen-reader users know where the link goes.

Effort: Low

Priority 3: Interactive Elements Missing Name, Role, or Value, WCAG 4.1.2

Bucket: Compliance Risk

Finding: Interactive elements lack an accessible name, role, or state that assistive technology and agents need to identify and operate them. This pattern appears 36 time(s) across the audited set, affecting screen reader users.

What to change and why:

- Give every custom control an accessible name and the correct role and state (prefer a native button/link/input; add ARIA only where no native element fits). This satisfies WCAG 4.1.2.
- A named, correctly-roled control is also what lets an agent understand what an interactive element does.

Effort: Medium

Priority 4: Info and Relationships Not Programmatically Determined, WCAG 1.3.1

Bucket: Compliance Risk

Finding: Visual structure (headings, lists, tables, form labels) is not exposed in the markup, so assistive technology and machines cannot reliably reconstruct it. This pattern appears 21 time(s) across the audited set, affecting screen reader users.

What to change and why:

- Expose the structure a sighted user sees (headings, lists, tables, form labels) in the markup so assistive technology and machines can reconstruct it. This satisfies WCAG 1.3.1.
- Use native semantic elements before ARIA; reach for ARIA only where no native element conveys the relationship.

Effort: Medium

Priority 5: Accessibility Issue, WCAG 3.2.2

Bucket: Compliance Risk

Finding: This form does not contain a submit button, which creates issues for those who cannot submit the form using the keyboard. Submit buttons are INPUT elements with type attribute “submit” or “image”, or BUTTON elements with type “submit” or omitted/invalid. This pattern appears 11 time(s) across the audited set, affecting all users.

What to change and why:

- Review the flagged instances against the relevant standard and remediate them at the template or configuration level so the fix applies wherever the pattern recurs.
- Add a check to the publish pipeline so the pattern is caught before it ships again.

Effort: Medium

Priority 6: No Bypass Mechanism for Repeated Blocks, WCAG 2.4.1

Bucket: Compliance Risk

Finding: Pages repeat navigation blocks with no mechanism to skip them, forcing keyboard users to tab through every link on each page before reaching the main content. This pattern appears 17 time(s) across the audited set, affecting sighted keyboard users.

What to change and why:

- Add a skip link as the first focusable element, or wrap the repeated navigation in a landmark, so keyboard users can jump straight to the main content. This satisfies WCAG 2.4.1.
- A served-HTML skip link also gives server-side agents an explicit main-content anchor they can follow.

Effort: Low

Priority 7: Heading Hierarchy Skips Levels

Bucket: Compliance Risk

Finding: Heading levels skip on 2 audited page(s) (for example an h2 followed by an h4), so the document outline a machine or screen reader builds does not match the visible structure.

What to change and why:

- Order headings without skipping levels (an h2 followed by an h4 forces assistive technology and machines to guess the structure). Use heading level for hierarchy and CSS for visual size.
- A clean heading outline is the spine an agent uses to summarise the page; fixing it improves both accessibility and machine comprehension.

Effort: Low

Priority 8: Main Landmark Absent

Bucket: Compliance Risk

Finding: 11 audited page(s) have no `<main>` landmark, so assistive technology and server-side agents cannot reliably locate the primary content among the navigation and chrome.

What to change and why:

- Wrap the primary content of each page in a single `<main>` landmark so assistive technology can jump to it and server-side agents can locate the content among the navigation and chrome.
- One `<main>` per page; everything that is not the page's unique content stays outside it.

Effort: Low

Priority 9: Semantic Structure 40/100

Bucket: Compliance Risk

Finding: Semantic-structure score 40/100: regions carry no role, ARIA landmark, or descriptive class, so machines fall back on positional inference to determine meaning. The worst rendered page (/signup) carries 32 generic containers of 49. A component framework renders these generic containers by design; adding landmark roles to the shared layout components clears the pattern site-wide.

What to change and why:

- A component framework emits generic containers by design, so this is a component-level configuration fix, not a rebuild: add landmark roles (or the semantic header/nav/main/footer/aside elements) to the layout components that wrap each region.
- Fixing the shared layout component clears the pattern across every page at once, since the same components render site-wide.

Effort: Medium

Priority 10: Security headers absent: HSTS, CSP, X-Frame-Options, X-Content-Type-Options

Bucket: Cross-cutting

Finding: Security headers absent: HSTS, CSP, X-Frame-Options, X-Content-Type-Options (All responses). Missing security headers increase exposure to content injection and clickjacking

What to change and why:

- Add the missing response headers at the server or CDN edge; each is a one-line directive that applies to all responses once configured.
- Set them once in the edge or server configuration rather than per page so coverage stays complete as new pages ship.

Effort: Low

Priority 11: Structured Data Property Gaps

Bucket: Machine Readability Opportunity

Finding: 49 Schema.org property gap(s) on the audited set across WebSite, WebPage: required or recommended properties are missing, so machines extract these entities less reliably.

What to change and why:

- Add the missing required and recommended Schema.org properties to the flagged entity types so machines can extract the entity reliably rather than guessing from surrounding text.
- Maintain the structured data in the template that renders each entity type so every instance carries the same complete markup.

Effort: Medium

Priority 12: Schema.org coverage is partial: Decoration (SDQ 66/100)

Bucket: Machine Readability Opportunity

Finding: Schema.org coverage is partial: Decoration (SDQ 66/100) (Homepage). Agents can partially parse structured facts but key properties may be missing

What to change and why:

- Add the missing required and recommended Schema.org properties to the flagged entity types so machines can extract the entity reliably rather than guessing from surrounding text.
- Maintain the structured data in the template that renders each entity type so every instance carries the same complete markup.

Effort: Medium

AI Agent Access Test

This test fetches the homepage using the User-Agent strings of known AI agents to verify whether this site is accessible at inference time.

Table 8

AI Agent Access Test

AI Agent	User-Agent	Status	Result
ClaudeBot (Anthropic)	`ClaudeBot/1.0`	200	Accessible
GPTBot (OpenAI)	`GPTBot/1.0`	200	Accessible
ChatGPT-User (OpenAI)	`ChatGPT-User/1.0`	200	Accessible
PerplexityBot	`PerplexityBot/1.0`	200	Accessible
GoogleOther (Google AI)	`GoogleOther`	200	Accessible
Google-Extended (Google AI-training opt-out)	`Google-Extended`	200	Accessible
CCBot (Common Crawl)	`CCBot/2.0`	200	Accessible
Plain request (no UA)	*(empty)*	200	Accessible

Summary: All 8 tested agents can access the site. No bot protection blocks AI agents at inference time.

Non-Standard Response Headers

No non-standard response headers detected. The site returns a clean, standard header set.

Error Page Test

This test fetches a deliberately non-existent page (`/zebedee.html`) to evaluate how this site handles errors for both human visitors and machines.

Table 9

Error Page Test

Check	Result
HTTP status code	200 (soft 404)
Custom error page	Yes, custom error page (not a bare server default)
Semantic HTML (`<main>`, `<nav>`, `<h1>`)	No
`<meta name="robots" content="noindex">`	No
Navigation back to valid content	Present - links back to same-site content found
Internal navigation links	48 links to same-site pages
MX governance tags	Absent
Schema.org JSON-LD	Present (the error page carries content schema, so machines can treat the 404 as a valid page; remove it)

The site returns HTTP 200 for a non-existent URL. Machines treat 200 as confirmation that a resource exists, so a soft 404 is invisible to them: a crawler building a dataset, an agent confirming a fact, or a validator checking a well-known path all record the dead end as valid content. The custom error page is a useful foundation. Two additions make it machine-aware: returning HTTP 404 (or 410) for missing URLs so machines can identify a dead end, and adding a `<meta name='robots' content='noindex'>` tag so crawlers do not index the catch-all page under every address they request. The absent navigation links mean an agent that lands here has no path to valid content; a home link and a short list of sections resolve that.

The Accessibility Tree

Machine visitors vary more than human ones. A small model on a phone works inside a tight context window. A foundation model arrives with browsing tools. A plain scraper never runs a model. A converter flattens each page to text before any model sees it; layout and scripts disappear in the process. A coding agent fetches a page once over HTTP and moves on. Raw markup, text projections, the accessibility tree, structured metadata - each kind of visitor reads a different layer. None of them read the visual layout, and you cannot know which one will arrive next.

Put the same meaning in every channel: page semantics, accessibility tree, metadata, and plain text. A page that stakes its meaning on one channel loses every visitor without it. The accessibility tree is the channel this section checks; it is shared by screen readers, so every fix here serves human and machine visitors alike.

Human visitors follow a path - step by step, page by page. Machines do not. They hit one page, once, and leave; they follow no path unless that page sends them somewhere. A site that hides its meaning across multiple pages is invisible to a machine landing in the middle. Every page has to stand alone for the machine while the journey still works for the human. These are complementary designs on the same pages.

This check reads each audited page as a tree consumer does and flags where behaviour, names, and structure fail to reach it. It covers what the WCAG scan does not: that section measures conformance per page; this one catches meaning that exists in only one channel.

Table 10

The Accessibility Tree

Measure	Result
Accessibility-tree score	66/100
Pages checked	10

Most pages pass tree-level checks; isolated structural gaps on a small number of pages.

Across the 10 pages we checked, 2 distinct issue patterns reduce what reaches the tree. 2 of these repeat across pages with the same structure, which marks them as template-level: one change in the shared component clears the finding everywhere it appears.

Form field labelled only by its placeholder (WCAG 2.1 3.3.2)

An input relies on placeholder text as its only label. The placeholder vanishes the moment the visitor types, is not reliably exposed as the accessible name, and disappears entirely in text projections of the page. Found across 2 components on 11 of 10 pages (a template-level pattern).

The fix: Add a real label (visible, or aria-label where the design demands it) in the form component; keep the placeholder as a hint, not the name.

Repeated landmarks with no distinguishing labels (WCAG 2.1 1.3.6)

The page carries more than one navigation (or complementary) landmark with no aria-label to tell them apart. In the tree they read as “navigation, navigation”: a consumer cannot tell the site menu from the footer links or the breadcrumb. Repeats on 8 of 10 pages with the same structure (a template-level pattern).

The fix: Give each repeated landmark a short aria-label in the template (“Primary”, “Footer”, “Breadcrumb”).

The following issue type(s) were also detected in the accessibility tree but are covered in the Accessibility section above and are not repeated here: Duplicate id used as an accessibility reference, No main landmark on the page, Data-bearing image with no text equivalent.

The full set, one row per pattern with every affected page counted, is recorded in the `specimen-media-accessibility-tree.csv` sidecar alongside this report.

Inspect the accessibility tree. Right-click any page, choose Inspect, open the Elements panel, click the >> icon, choose Accessibility, and toggle “Show Accessibility Tree”. The result is what tree consumers receive: if a control or a heading is missing from that view, it is missing for them. Chrome DevTools’ AI Assistance panel also accepts “Review accessibility” against any element this report flags.

Server Response Stability

Single load-time measurements can mislead. A page that returns in a few hundred milliseconds for a returning visitor may be served from a warm CDN edge. The same page on a genuine first visit could spend several seconds at the origin before the first byte arrives. To separate the two experiences, this section re-measures the slowest page from the crawl and a median-load control across several fresh visits, then compares those against the first-visit response. The result is two distinct verdicts per page: a first-visit cost (what a brand-new visitor actually pays) and a returning-visitor cost (what a repeat visitor experiences). The overall verdict for each page is the worse of the two, so a fast returning-visitor median cannot paper over a slow first-visit response.

Method: Each URL is re-measured across several fresh visits and scored on the median of those measurements. For each page we compare both the crawler's cold-cache baseline and the median of three fresh GETs: a response is treated as healthy at or below 1500ms, acceptable up to 3000ms, and slow above 3000ms. The overall verdict reflects the worse of the two views.

Slowest. The slowest page is `https://specimen-media.example/`. A first-time visitor sees the cold-cache cost: the crawler recorded 1744 ms on its initial fetch. **First-visit verdict: Acceptable but elevated**. Three fresh re-probes that followed returned 142ms, 130ms, 121ms, giving a returning-visitor median of **130 ms**. **Returning-visitor verdict: Healthy**.

Median-load control. The median-load control page is `https://specimen-media.example/llms.txt`. A first-time visitor sees the cold-cache cost: the crawler recorded 461 ms on its initial fetch. **First-visit verdict: Healthy**. Three fresh re-probes that followed returned 545ms, 468ms, 1460ms, giving a returning-visitor median of **545 ms**. **Returning-visitor verdict: Healthy**.

Verdict: Server response time is within healthy bounds for the slowest page across both first-visit and returning-visitor views.

Discovery Files

robots.txt

```
User-agent: *
Allow: /

Sitemap: https://specimen-media.example/sitemap.xml
```

The robots.txt declares no disallow paths, so every path is open to crawlers and machines. It references the sitemap, so a machine reading the file can locate the URL index directly.

sitemap.xml

Table 11

sitemap.xml

Attribute	Present	Assessment
`<loc>` URLs	5983 entries	Present
`<lastmod>`	Yes	Varied dates
`<changefreq>`	No	Missing (Google dropped this as a ranking signal in 2017; non-Google crawlers and AI agents still use it to gauge re-crawl cadence)
`<priority>`	Yes	Differentiated values

Sitemap grade: Partial

The sitemap declares 5983 URLs and grades Partial. Lastmod dates vary across entries, which tells machines which pages changed and when. The sitemap omits changefreq. Google dropped this as ranking signals in 2017, but non-Google crawlers and AI agents still read changefreq as a re-crawl cadence hint and priority as a relative-importance signal, so adding it is a low-effort way to broaden machine compatibility.

The sitemap lists 5983 URLs; 11 of the pages this audit reached are not among them. The full set is recorded in the `specimen-media-pages-not-in-sitemap.csv` sidecar alongside this report. Adding them to the sitemap lets search engines and machines discover all content.

We discovered that the sitemap lists multiple URL variants-trailing-slash and hash-fragment forms-for the same canonical resource, such as https://specimen-media.example/llms.txt appearing twice. Machines that skip URL normalisation will treat each variant as a distinct page, inflating token usage and risking contradictory or duplicated findings; we recommend publishing one canonical URL per resource in sitemap.xml and adding a tag to each affected URL.

llms.txt

The llms.txt carries a content-use policy, but lacks a site description and a page inventory; adding them would give machines a complete structured index. We also recommend serving llms.txt as an HTML page that wraps the plain-text content in a `<pre>` block, rather than the text/plain the llmstxt.org specification defines. Training crawlers such as

Common Crawl archive only a small fraction of plain-text files but crawl HTML pages from the sitemap reliably, so the HTML wrapper gets the file into the corpus while the <pre> keeps it rendering as readable plain text. The technique, with the reasoning and working code, is at https://mx.allabout.network/blog/your-site-is-already-training-ai.html.

llms-full.txt

Not found. A full content corpus at /llms-full.txt would let agents ingest the complete site in a single fetch. Without it, an agent that wants to index all content must crawl page by page.

agent-card.json (A2A)

No agent-card.json found at `/.well-known/agent-card.json` (HTTP 404). The A2A (Agent2Agent) protocol defines this location as the standard way to make services findable in agentic workflows. If this site offers transactional or service capabilities, publishing an agent card here is the most important gap to close for Stage 5 (Confidence).

Other discovery files detected

Table 12

Other discovery files detected

Path	Purpose	Quality
(5 paths - see sidecar)	Various	Soft 404 - server returns the home page for missing resources

Soft 404s detected (7 paths): The server returns HTTP 200 for these paths but does not serve the expected resource. AI agents and crawlers rely on HTTP status codes to determine whether a resource exists. The server should return HTTP 404 (or 301 to a canonical URL) for paths it does not implement. This is a web server configuration change, not a content change.

Reference: the IANA Well-Known URIs registry lists the full set of registered `/.well-known/` paths and their RFCs. If a path on that registry would be useful here, consider implementing it.

AI-Content Marking Readiness

This section reports whether Specimen Media Co's site marks AI-generated or AI-manipulated content in a machine-readable way. Two frameworks define what that marking looks like: EU AI Act Article 50, which expects it from 2 August 2026, and the European Commission's voluntary Code of Practice on marking and labelling of AI-generated content, published 13 June 2026. The probe inspects the homepage for four markers an agent could read without a human in the loop, and records which are present.

Note: This section describes regulatory frameworks in general terms only. Nothing here is legal advice. Requirements vary by jurisdiction, organisation type, and use case. Consult qualified legal specialists for guidance specific to your situation.

Table 13

AI-Content Marking Readiness

Attribute	Value
Origin	https://specimen-media.example
Reference	EU AI Act Article 50; European Commission Code of Practice on marking and labelling of AI-generated content (13 June 2026)
Readiness level	Level 0 (Unmarked)
Markers present	0/4
Verdict	unmarked

Markers

Table 14

Markers

Marker	Present	Detail	Note
MX provenanceOrigin	no	None	Machine-authorship declaration carried in the file (MX).
IPTC Digital Source Type	no	None	Standard machine-readable AI-content marker a detector reads.
C2PA Content Credentials	no	None	Content-authenticity manifest signal (presence recorded, not verified).
AI-disclosure meta	no	None	Generic, non-standard disclosure tag.

Probe findings

- No machine-readable machine-authorship marking was found on the homepage: no MX provenanceOrigin, no IPTC Digital Source Type, no C2PA signal, no AI-disclosure meta. A machine reading this page cannot tell whether its content was generated or altered by a machine. From 2 August 2026 the EU AI Act Article 50 expects AI-generated or AI-manipulated content to carry exactly such a marker.
- Marking readiness here is about whether content announces machine authorship in a machine-readable way. MX provenanceOrigin declares the authorship; a content-authenticity watermark (C2PA, SynthID) proves a file is synthetic. The two are complementary, and this probe reports readiness, not compliance with any regulation.
- **No machine-readable authorship markers were found.** A machine reading this page cannot tell whether its content was generated by an AI tool or written by a person - absence of markers means undeclared, not "human." If this site's content is entirely human-authored, a positive declaration would make that explicit and provide a trust

signal to regulators and AI citation engines. Options: add `MX provenanceOrigin: human` in page metadata, or apply IPTC Digital Source Type `humanCreated` to image assets. Either is better than silence.

A boundary this section keeps honest: a machine-authorship declaration (MX `provenanceOrigin`) states who or what authored the content; a content-authenticity watermark (C2PA, SynthID) proves a file is synthetic. They are complementary, and marking readiness here is a structural signal, not a certification that this site meets any regulation.

Soft 404s

This site answers HTTP 200 for addresses that do not exist. A control address that cannot be real still returned a 200 with a normal-looking page. This is a soft 404, and on this site it is the default for missing addresses, not an isolated case. For a person it is invisible. For a machine it is corrosive: a 200 is the signal that a resource is present, so every check of the form “does this exist?” now returns yes. An agent confirming a price, a product, a policy, or a declaration cannot tell a real answer from a placeholder. A crawler building a training set ingests the catch-all page under thousands of distinct addresses as if each were real content. A validator probing for a well-known file records it as published when nothing is there. The correct behaviour is to return 404 (or 410) for an address that does not resolve, and to reserve 200 for addresses that do. Until that holds, no presence claim derived from a fetch of this site can be trusted, including some made elsewhere in this report where the underlying probe was misled.

We probed 53 addresses that should answer 404 when they are absent; 11 returned a soft 404 instead. Among the well-known discovery paths, 7 of 45 were soft 404s. Severity: pervasive.

Structured Data Inventory

Table 15

Structured Data Inventory

Schema Type	Pages	Mandatory fields present	Optional fields added	Notes
Organisation	11	100%	100%	Complete
ListItem	9	100%	100%	Complete
WebSite	11	100%	0%	All required fields present; 100% of optional fields missing
SearchAction	11	100%	100%	Complete
WebPage	5	0%	100%	100% of required fields missing
Person	5	100%	100%	Complete
ImageObject	5	100%	100%	Complete
BreadcrumbList	9	100%	100%	Complete
Article	5	100%	100%	Complete
NewsArticle	5	100%	100%	Complete

Structured Data Quality: 66/100 - Schema.org data is present and valid; recommended properties and entity links are sparse.
Coverage: 11 pages with JSON-LD out of 11 total (100%)
Unique types: 10

Schema types are from Schema.org - a shared vocabulary machines use to read content without guessing. "Mandatory fields present" shows whether the required properties for that type are filled in. "Optional fields added" shows bonus properties that help machines understand the content more precisely. Higher is better on both.

Across the 11 pages we audited, structured data is solid. Adding recommended properties and increasing type diversity on the sampled pages gives machines more to work with.

SDQ Score Breakdown

The Structured Data Quality score is composed of seven measurable signals. This breakdown shows what Specimen Media Co earns in each.

Table 16

SDQ Score Breakdown

Component	Earned	Max	Meaning
Presence	10	10	schema.org JSON-LD is present on the page
Required property coverage	14	25	Every entity carries the properties its type requires
Recommended property coverage	13	15	Entities carry the properties their type recommends
Entity richness	7	15	Entities are described with enough properties to be useful
Cross-entity references	7	15	Entities reference each other (nested types and @id links)
Linked-data signals	6	10	Linked-data properties present (sameAs, mainEntityOfPage, isPartOf, about, mentions)
Vocabulary validity	10	10	Every @type is a valid Schema.org type

Component	Earned	Max	Meaning
Total	**66**	**100**	

We found Linked-data signals and Entity richness to be the lowest-scoring components, indicating sparse inter-entity connections and limited content detail in the schema markup. These gaps keep the site at a Decoration maturity level, where structured data is present but lacks depth and breadth for advanced machine comprehension.

Structured Data Findings

We identified 49 specific Schema.org property gaps. Each row names a single missing property on a single entity with a short note on why it matters to machines.

The full per-entity list is delivered alongside this report as a sidecar CSV: `specimen-media-structured-data-findings.csv`. The 49 rows describe individual Schema.org property gaps on specific entities; most of them share a small number of underlying patterns, shown below ranked by instance count.

Table 17

Structured Data Findings

Type	Severity	Property	Instances	Pages	Why it matters
WebSite	recommended	image	11	11	Site has no logo / hero image declared in structured data
WebSite	recommended	datePublished	11	11	No site-level publish date for crawler context
WebSite	recommended	author	11	11	Site has no top-level author/owner declared
WebSite	recommended	publisher	11	11	Site has no top-level publisher declared
WebPage	required	name	5	5	Page has no machine-readable title beyond

Each summary row covers multiple per-entity rows in the sidecar; the grouped view is for reading at a glance, the sidecar is for processing.

Severity legend (the values in the *Severity* column above):

Table 18

Structured Data Findings

Severity	Meaning
`required`	Schema.org spec requires this property for the type. Missing values break validation.
`recommended`	Schema.org strongly recommends this property. Missing values reduce richness.
`vocabulary`	The `@type` value (the JSON-LD class name an entity declares itself as) is not in the Schema.org vocabulary: typically a typo or an invented type.

Marker Reachability

Table 19

Marker Reachability

Marker	In served	In rendered	In head	Reachable <250KB	Injected by JS
JSON-LD structured data	Yes	Yes	Yes	Yes	No
Microdata (itemscope)	Not present	Not present	n/a	n/a	n/a
Open Graph meta tags	Yes	Yes	Yes	Yes	No
Twitter Card meta tags	Yes	Yes	Yes	Yes	No
MX governance meta tags	Not present	Not present	n/a	n/a	n/a
Canonical URL	Yes	Yes	Yes	Yes	No
Discovery links (llms-txt, sitemap)	Not present	Not present	n/a	n/a	n/a
Language declaration (html lang)	Yes	Yes	Yes	Yes	No
Skip link (accessibility)	Not present	Not present	n/a	n/a	n/a

All detected markers are present in the served HTML on the pages we audited. Server-side and browser-based agents see the same signals on the sampled pages.

Schema Maturity Level

Schema.org implementations fall into five maturity tiers. The transitions are not continuous. Each level requires structurally different work. Maturity is a structural classification: it depends on what the markup carries (typed blocks, required properties, cross-references, external identifiers), not on the SDQ score the markup happens to earn. A page can sit at Level 1 with a high SDQ score and at Level 3 with a moderate one. Score and level are reported separately.

Table 20

Schema Maturity Level

	Level	Name	What it looks like	
	0	Clean slate	No Schema.org markup present. The full maturity curve is open: every property added at this stage is net new capability.	
→	1	Decoration	Typed blocks present, with sparse properties and no nesting or cross-references. The structural primitives are in place; the next opportunity is to fill the required and recommended fields.	←
	2	Good schema	Full required and recommended properties, nested types where appropriate, valid vocabulary. The next opportunity is to wire entities together with @id cross-references.	
	3	Real graph	Level 2 plus @id cross-references between entities and linked-data signals (sameAs, mainEntityOfPage, isPartOf). The next opportunity is to anchor entities to external identifiers.	
	4	Verified linked data	Level 3 plus external identifiers (Wikidata QIDs, ISNIs, ORCIDs) and provenance metadata. Entities are anchored in the linked-data web.	

Current level: 1: Decoration
To reach the next level: Fill in the required and recommended Schema.org properties for each typed block (see structured_data_findings.csv for the specific gaps). Connect related entities inline or via @id references to canonical entities defined elsewhere on the site. Ensure every @type value is a valid Schema.org type.

The Structured Data Quality (SDQ) score and the Schema Maturity Level measure two different things. SDQ counts the properties present and validates them against Schema.org expectations; the level captures whether those properties are connected (cross-entity wiring, linked-data signals, external authority identifiers). Both numbers above are reported as-is from the audit data.

The classification is conservative: every Schema.org block on every audited page must clear a level's bar for this site to claim it, so a handful of thin blocks or pages without markup caps the level even when most pages individually sit higher. That is deliberate. An agent does not choose which page it lands on, so the level reflects what the weakest landing point guarantees.

5-Stage MX Journey

The MX Journey maps the five stages a machine follows when interacting with a website. Each stage builds on the previous one. A break at any stage propagates to all subsequent stages.

Table 21

5-Stage MX Journey

Stage	Name	Status	Score	Key Metric
1	Discovery	Partial	78	Issues: no
2	Citation	Partial	67	Schema.org: WebSite, SearchAction, Organisation (100% required properties)
3	Search & Compare	Site type does not require	–	No comparison content detected
4	Service Inquiry Readiness	Site type does not require	–	No pricing content detected
5	Transaction (N/A for services)	Site type does not require	–	No transaction forms detected

Each stage carries its own pass threshold, so Status and Score are not comparable across rows: a score that passes one stage can fall short on another with a stricter bar.

The pages we audited meet MX compatibility for this site type. Search-and-compare, price understanding, and purchase confidence do not apply here, because no transaction or checkout pages were detected.

Agent Reading Pipeline

Scoring a machine's metadata is not the same as scoring whether a machine can read each page at all. Pipeline Survivability runs eleven reading-resilience checks on every audited page. Each one asks whether a page survives a known agent-reading risk: truncation by the agent's fetch tool, condensing by the relevance layer, JavaScript-only content, tab disclosure, soft 404s, broken code fences, content negotiation drift, cross-host redirects, generic headings, content that begins too far into the document, or overhead-heavy pages where scripts, styles, and images outweigh actual content.

Every check runs on every audited page. The aggregate score weights truncation resilience, SPA resilience, and proper 404 signalling most heavily: these three determine whether each page is reachable to the agent at all. Boilerplate burial, tabbed disclosure, and delayed content start carry medium weight. The remaining checks contribute to the score but any single one slipping is less critical on its own.

- **Truncation Risk** - Fail · 4/11
  - *Means:* 4 page(s) flag for truncation risk; 2 of them exceed the 250 KB hard ceiling, the rest place main content too far into the document. Agents with limited fetch windows may stop reading before reaching the main content.
  - *Data:* Largest page: 837 KB (the home page). Thresholds: 250 KB hard ceiling; 50/75/100 KB content-offset windows. See specimen-media-pipeline-truncation-risk-pages.csv (4 pages).
- **SPA Shell** - Fail · 9/11
  - *Means:* Content requires JavaScript to appear. Server-side agents (ChatGPT, Claude, Perplexity) see an empty shell when they fetch these pages.
  - *Data:* Max gap score: 60. 0 means served and rendered match. See specimen-media-pipeline-spa-shell-pages.csv (9 pages).
- **Soft 404** - Fail · catch-all
  - *Means:* This site returns HTTP 200 for addresses that do not exist (a soft-404 catch-all), so agents and search engines cannot distinguish a missing resource from a real one. Missing addresses must return HTTP 404 or 410.
  - *Data:* A control address that does not exist returned HTTP 200; 7 of 45 well-known paths are soft-404s.
- **Boilerplate Burial** - Pass · 0/11
- **Tabbed Disclosure** - Pass · 0/11
- **Delayed Content Start** - Pass · N/A
- **Broken Code Fences** - Pass · 0/11
- **HTTP Content Negotiation (Vary)** - Pass · 0/11
- **Cross-Host Redirect** - Pass · 0/11
- **Generic Headings** - Pass · 0/11
- **Body Content Ratio** - Fail · N/A
  - *Means:* Prose content averages only 17% of served bytes. Scripts, styles, and images dominate; agents get little signal per byte.
  - *Data:* Average: 17%. Threshold: 30%.
- **Inline Tag Bloat** - Fail · 11/11
  - *Means:* 11 page(s) carry inline `<style>` or executable `<script>` blocks over 500 bytes. Externalising these to separate .css/.js files lets agents skip them during cheap fetches.
  - *Data:* 19 element(s) > 500 bytes. Largest single-page inline CSS block: 12364 B. Largest single-page inline JS block: 0 B. See specimen-media-pipeline-inline-tag-bloat-pages.csv (11 pages).
- **Head Weight** - Pass · N/A

Pipeline Survivability score: 77/100 Pages reach agents intact with no observed survivability issues.

Across the audited set, every page suffers from Inline Tag Bloat, which forces machines to download and parse excessive markup; this can cause them to miss or misinterpret key information. Truncation Risk and SPA Shell also appear on some pages, meaning agents may receive incomplete content or only a shell that requires JavaScript execution. Addressing

Inline Tag Bloat first offers the greatest improvement, while cleaning up truncation and converting shells into server-rendered markup will further strengthen machine comprehension.

For the methodology behind this section, the relevance layer concept, and the canary-token method that informs the check set, see **MX: The Protocols Appendix R: Testing Agent Comprehension** and **Appendix S: The Eleven Agent Reading Resilience Checks**.

Semantic Structure: landmark roles for machine navigation

When a page's containers are generic `<div>` elements with no role, no ARIA landmark, and no class name that describes what they are, machines lose structural context and fall back on positional inference ("the third region from the top is probably navigation") to determine meaning. The visual layout still works for sighted users; the structural information that machines need to index, cite, and represent each page is missing. Component frameworks emit generic containers by design, so this is usually a component-level configuration fix (add landmark roles), not a rebuild.

We measure semantic structure on both served and rendered HTML so we can tell whether the generic containers are in the source the publisher controls or something the JavaScript framework introduces at render time. Score 100 is a page whose regions carry landmark roles; score 0 is the worst case (every region is a generic nested container with no role).

Table 22

Semantic Structure: landmark roles for machine navigation

Source	Score (band)	Generic-container stats	Top container selectors
Served HTML	40/100 (many generic containers)	40 generic containers (59% of containers, depth 5)	`div` (222), `div.d-none.d-lg-block` (22), `div.d-lg-none` (19), `div.h-100` (14), `#__next` (9)
Rendered HTML	31/100 (many generic containers)	32 generic containers (65% of containers, depth 6)	`div` (227), `div.h-100` (67), `div.mb-3` (20), `div.row.g-0` (18), `div.text-center.d-flex` (13)

Worst page (served): /example-feature-article
Worst page (rendered): /signup

Rendered content on the worst page (https://specimen-media.example/signup) has the highest generic-container ratio, at 65 % of its elements being bare divs; this means machines lose structural context and must rely on positional inference to determine meaning.
The deep chain of six nested bare divs suggests a structural pattern resulting from a drag-and-drop builder or late-stage JavaScript injection.
The cheapest first move is to wrap the header, nav, main, footer, and aside landmarks in their semantic tags or assign appropriate ARIA landmark roles, and to replace generic divs with descriptive class names; this lowers the generic-container ratio without changing layout.

Security Headers

Table 23

Security Headers

Header	Status	Purpose
HTTPS	Yes	Encrypted transport
HSTS	No	Forces HTTPS, prevents downgrade attacks
Content-Security-Policy	No	Prevents XSS and injection attacks
X-Frame-Options	No	Prevents clickjacking
X-Content-Type-Options	No	Prevents MIME-type sniffing

4 of the five standard security headers are absent on every audited response: HSTS (Strict-Transport-Security), Content-Security-Policy (CSP), X-Frame-Options, X-Content-Type-Options. Adding them at the origin-server or CDN edge closes the corresponding attack surfaces without touching application code.

Coverage: 0 of 12 audited URLs carry all five headers; see the Security Headers appendix for the full exception list.

- `/`: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/signup`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/signin`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/forgot`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/programs`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/llms.txt`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/mx-to-the-max-tom-cranstouns-new-book-decodes-machine-experience-for-humans-and-agents`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/example-feature-article`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/matt-matt-show-episode-7-2026-halftime-report-vercel-ship-to-full-sail-the-aeo-goal-and-more`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/bland-brands-beware-optimizelys-new-identity-brings-human-creativity-to-its-agentic-ai-and-aeo-ambitions`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/ai-and-accessibility-incredible-potential-inconvenient-questions`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No
- **`/articles`**: HTTPS Yes · HSTS No · CSP No · X-Frame No · X-Content-Type No

HTTPS: 12/12 | HSTS: 0/12 | CSP: 0/12 | X-Frame-Options: 0/12 | X-Content-Type-Options: 0/12

Data Quality and Consistency

Cross-Page Consistency

Table 24

Cross-Page Consistency

Pattern	Coverage	Pages missing it
Schema.org JSON-LD	100%	None
MX governance tags	0%	11
Open Graph tags	100%	None
Twitter Card tags	100%	None
Skip link	0%	10
llms.txt link tag	0%	10
Canonical URL	100%	None
Exactly 1 H1	100%	None
Code examples present	0%	11
Self-contained sections	82%	2
Error/troubleshooting docs	0%	11
Lighthouse heading compliance	82%	2

Overall Consistency: 64% Most metadata patterns are consistent; a few gaps mean some pages deliver weaker signals than others.

Some pages in the 11-page sample are missing metadata patterns that others carry. Machines hitting different pages get different quality data. The Missing Pages column shows where to focus on the sampled pages.

Content Consistency

The audited set shows consistent metadata patterns across pages, with no brand-name, canonical-URL, meta-description, or entity divergence detected.

Table 25

Content Consistency

Check	Result	Notes
Brand-name parity	Pass	Brand name appears consistently across all 11 audited pages
Canonical URL duplicates	Pass	No duplicate canonical URLs detected across the 11-page audited set
Meta description length	Pass	Meta descriptions present on all pages; none flagged for length violations
Cross-page entity spread (same entity on multiple pages)	Pass	Schema.org entities reference consistent identifiers across the audited set

Every consistency check passes, so an agent reading any two audited pages receives the same brand, canonical, and entity signals.

Inline Code Duplicates

We found 6 identical inline fragment(s) repeated across multiple pages, totalling 187 KB redundant bytes. Extracting these to external CSS or JS files would reduce page weight, improve cacheability, and simplify maintenance.

Table 26

Inline Code Duplicates

Type	Bytes per fragment	Appears on N pages	Preview
css	10820	11	@font-face{font-family:'Inter';font-style:normal;font-weight
css	6644	11	.adroll_consent_container{position:relative}.adroll_consent_
js	590	11	adroll_adv_id="RHA3WXTZ3VEVPOSXEQHLJS";adroll_pix_id="OH7AVS
js	427	11	(function(w,d,s,l,i){w[l]=w[l]||[]; w[l].push({'gtm.
js	299	11	var _cio = _cio || []; (function() { var a,
css	581	8	.css-4xx2wk{display:-webkit-inline-box;display:-webkit-inlin

Recommendation: Move each duplicate fragment to a shared external file (`<link rel="stylesheet">` for CSS, `<script src=".">` for JS). The fragment hash in `consistency_analysis.json` identifies exactly which blocks are identical.

Infrastructure and Hosting

The site is served via **AWS Route53** (Amazon.com, Inc.). Platform is estimated to be **Unknown Platform**, though signals are ambiguous.

We linked no PDFs from the 11-page sample we crawled, and the sitemap declares no `.pdf` URLs either. This is a statement about what we sampled and what the sitemap reports, not a verdict about the wider document estate: PDFs do not appear in this count if they sit behind login forms, are linked only from uncrawled pages, are stored in unlinked directories, are kept out of the sitemap, or are hosted on third-party domains.

PDFs are part of the machine-readable estate but sit outside this HTML audit's scope. A dedicated PDF review checks each public document against the ISO 14289-1 (PDF/UA) baseline (Tagged, Declared, Verified) and returns a per-document verdict.

Text Patterns

Analysis of text patterns across audited pages found content reaching Probably AI on the AI-tells scale (1 of 11 pages scored). Machines do not consistently cite or label AI-generated content; this observation describes what the analysis found, not a conclusion about authorship.

/programs (Probably AI) - prose patterns, vocabulary.

/mx-to-the-max-tom-cranstouns-new-book-decodes-machine-experience-for-humans-and-agents (Occasional) - prose patterns, vocabulary, verbal tics.

/example-feature-article (Occasional) - prose patterns, vocabulary, verbal tics, copula density.

/matt-matt-show-episode-7-2026-halftime-report-vercel-ship-to-full-sail-the-aeo-goal-and-more (Occasional) - prose patterns, vocabulary, verbal tics.

/bland-brands-beware-optimizelys-new-identity-brings-human-creativity-to-its-agentic-ai-and-aeo-ambitions (Occasional) - prose patterns, vocabulary, copula density, verbal tics.

6 of 11 audited pages show AI-writing patterns.

The remaining 1 flagged page is in `specimen-media-ai-tells.json` alongside this report.

Most of the audited set (6 of 11 pages) carries the sentence rhythms, vocabulary, and structural tells associated with machine-assisted drafting. For a publisher, a pattern this widespread is worth an editorial look at the drafting workflow. This describes what the text analysis measured, not a conclusion about who or what wrote each page.

Next Steps

Recommended Actions

1. **Address Priority 1 findings**: address the 566 WCAG 2.1 AA accessibility issues identified (regulatory exposure)
2. **Review Priority 2-3 findings**: Semantic Structure improvements and metadata tuning that compound over time
3. **Consider optional enhancements**: optional patterns that give an early-mover opportunity in AI search

What's Next

Table 27

What's Next

Phase	Scope	Outcome
Critical Fixes	P1, P2, P3, P4, P5, P6, P7, P8, P9 (Compliance Risk)	Priority 1, 2, 3, 4, 5, 6, 7, 8, 9 resolved: WCAG 2.1 AA accessibility compliance restored
Full Implementation	P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12 (P1-P12)	Full machine readiness: every agent, search engine, and structured-data consumer can read, trust, and act on the site
Ongoing Monitoring	Continuous monitoring and quarterly audits	durable visibility in agent-mediated discovery
Machine-Ready Estate	Web estate + PDFs + data feeds + APIs + documents	The full machine-readable estate, beyond the web pages
Data-Sovereign Option	Regulated industries	Run the full audit pipeline on your own infrastructure - no client content leaves your network

This audit is a starting point. The outcome we work toward is a site any machine can read, trust, and act on, and a dated, attested record you can show to a regulator, a partner, or an acquirer on request. Reaching it (structured data, discovery files, accessibility, governance metadata, and re-audit on a schedule you set) is available as a managed service. We also run training sessions that give development teams the MX vocabulary and implementation patterns directly, so the gap between findings and fixes is weeks, not quarters. To take any of it further, contact CogNovaMX Ltd at info@cognovamx.com.

Audit Scores

Each dimension is measured independently. Served dimensions reflect each page before JavaScript runs; Rendered dimensions reflect what a browser produces after JavaScript executes. The Notes column explains the measurement method for each score.

Table 28

Audit Scores

Dimension	Score	Band
Served-HTML Structure	87/100	Excellent
Accessibility	58/100	Good
SEO (all pages)	80/100	Excellent
SEO (content pages)	80/100	Excellent

Dimension	Score	Band
MX Stack Completeness	61/100	Good
Structured Data Quality	66/100	Good
Discovery Readiness	40/100	Could Be Better
Heading Quality	87/100	Excellent
Agent Readability	89/100	Excellent
Pipeline Survivability	77/100	Excellent
Cross-Page Consistency	64%	Good

Working With Us

We run the automated pass and surface findings to priority. The next step is remediation, and we offer it in several forms:

- **Full-render, all-pages audience and age-awareness review.** We classify the entry page in this audit; the full-render version covers every page - age-assurance, consent, and date-of-birth data collection across the whole estate.
- **Full-site qualitative review.** We read every audited page for the content-quality patterns the automated pass samples only on the first few pages.
- **PDF estate accessibility remediation.** We tag the structure, declare conformance, and record an independent check across the document estate, aligned with Directive (EU) 2019/882.
- **On-premise, regulated-sector audit.** We run the whole pipeline against a local model on infrastructure you control, so no audited content leaves your network.
- **Implementation and remediation.** We bring the technical context from these findings into the work that resolves them.

To set up a remediation plan, contact info@cognovamx.com.

Appendix A: Pages Audited

- **/ (nav)** SEO 90 · A11y 70 · Backend 70 · Served 94 · Rendered 77
- **/signup** SEO 82 · A11y 70 · Backend 70 · Served 98 · Rendered 52
- **/signin** SEO 83 · A11y 60 · Backend 70 · Served 98 · Rendered 52
- **/forgot** SEO 82 · A11y 65 · Backend 70 · Served 73 · Rendered 52
- **/programs** SEO 84 · A11y 65 · Backend 70 · Served 95 · Rendered 77
- **./mx-to-the-max-tom-cranstouns-new-book-decodes-mach.** SEO 75 · A11y 45 · Backend 70 · Served 68 · Rendered 50
- **./diving-deep-with-palmata-contentful-uplevels-aeo-t.** SEO 74 · A11y 50 · Backend 80 · Served 82 · Rendered 65
- **./matt-matt-show-episode-7-2026-halftime-report-verc.** SEO 75 · A11y 45 · Backend 70 · Served 77 · Rendered 60
- **./bland-brands-beware-optimizelys-new-identity-bring.** SEO 75 · A11y 60 · Backend 70 · Served 82 · Rendered 65
- **./ai-and-accessibility-incredible-potential-inconven.** SEO 82 · A11y 45 · Backend 70 · Served 93 · Rendered 75
- **/articles** SEO 83 · A11y 60 · Backend 70 · Served 94 · Rendered 73

Backend: score for HTML served without JavaScript. Served: AI suitability from served HTML. Rendered: AI suitability after JavaScript.

The page marked (nav) is navigational: it routes visitors to content rather than containing it, and is excluded from the SEO content average. Content-pages SEO average: 80/100.

Appendix B: Link Inventory

We recorded every same-host internal link found on each audited page. External links are not tracked; this inventory covers same-host `<a href>` links only. Link status was not probed; for a dedicated broken-link audit, run a rate-limited crawler on the link set at a time that suits the site.

Per page, internal links range from 4 to 83, averaging 38 across 12 crawled pages. That is typical (benchmark median 20 per page).

Table 29

Appendix B: Link Inventory

Link class	Count
Same-host internal links (all pages)	460
External links (not tracked)	–
Anchor-only (`#fragment`) links	0
mailto / tel links	0

At 38 internal links per page on average, the internal navigation graph sits within the typical range for sites of this type (benchmark median 20). No hash-fragment links were found - the site navigates entirely by full-page URL, which is standard for content and service sites. No inline mailto or tel links appear in page content; direct contact routes through a form.

Appendix C: Image Efficiency

We reviewed 340 images across the audited set: 31 SVG, 90 PNG, 216 JPEG and 3 in other or unidentified formats. 279 of 340 (82.1%) carry alt text, leaving 61 without it. Each missing alt attribute is a place where a screen-reader user or a machine reading the page gets no description of what the image shows.

On loading strategy, 255 images are marked `loading="lazy"` and 0 `loading="eager"`, while 85 carry no loading attribute at all. No attribute is not the same as eager: the browser decides for itself when to fetch, which removes the explicit control that lazy and eager give you. Setting an explicit attribute on those images makes the fetch behaviour predictable for browsers and machines alike.

JPEG and PNG account for most of the images, but none use WebP, which typically reduces file size by 25-35 % over PNG or JPEG without visible quality loss.s.

Table 30

Appendix C: Image Efficiency

Format	Count	Share
SVG	31	9%
PNG	90	26%
JPEG	216	64%
Other	3	1%

Appendix D: Audit Methodology

Tools: Web Audit Suite v2.x (automated WCAG 2.1 AA accessibility checks, performance metrics, SEO scoring, LLM suitability, MX Stack Completeness, Structured Data Quality, Discovery Readiness, Heading Quality, Cross-Page Consistency)

Accessibility is assessed with an open-source automated testing tool that checks web pages against the Web Content Accessibility Guidelines (WCAG 2.1 AA).

Accessibility testing here is automated only. Automated tools reliably detect roughly a third to a half of WCAG 2.1 AA success criteria - typically contrast, missing alternative text, form-label association, and document-structure errors. They cannot evaluate the criteria that need human judgement: keyboard operability and focus order, the absence of keyboard traps, logical reading and tab sequence, meaningful link and heading text in context, error-recovery flows, and cognitive load. A clean automated pass is a necessary baseline, not a certification: an automated tool cannot certify WCAG 2.1 AA conformance, and a full conformance claim needs manual expert testing and assistive-technology walkthroughs. The accessibility score reflects the automated-checkable subset only.

MX-specific metrics: MX Stack Completeness measures all 7 metadata layers. Structured Data Quality (SDQ) scores JSON-LD entity richness. Discovery Readiness scores the robots.txt + sitemap + llms.txt + agent-card.json quartet. Cross-Page Consistency flags pages that deviate from site-wide patterns. Site Profile JSON enables cross-audit comparison. **Pipeline Survivability** runs eleven reading-resilience checks: truncation resilience, SPA shell resilience, soft-404 signalling, boilerplate balance, tabbed-disclosure avoidance, code-fence integrity, single-content-type negotiation, same-host redirects, heading specificity, early content start, and inline-tag bloat control. See **MX: The Protocols Appendix S** for the full taxonomy and **Appendix R** for the testing methodology.

Platform detection: We fingerprint the hosting platform from HTTP response headers, HTML signatures, asset paths, and class patterns. Platform identification is probabilistic – a site can obscure or mimic platform signals. We report the result as: No platform detected. No platform-specific fingerprint was detected, so the audit used conservative default rate limits, paced slowly enough to stay below typical shared-host thresholds, with exponential backoff and retry (up to 4 attempts) on rate-limit responses.

Frameworks detected: Next.js (medium confidence) - JS framework; **Bootstrap** - CSS framework. Framework detection scans JS component frameworks, CSS utility libraries, CMS plugins and page builders, and CDN/delivery layers from the audited pages. Confidence is high (3+ signals), medium (2 signals), or low (1 signal, treat as a hint). Low-confidence detections are noted but do not influence scoring.

Next.js renders content into generic containers - normal for a component framework, not a fault. Machines still need landmark roles (header, nav, main, footer) to map page regions; adding these at the component level is a config change, not a rewrite, and lifts the Semantic Structure score. Next.js provides server-side rendering, giving agents full HTML on first fetch with no JavaScript dependency.

Link inventory: We record every internal link found on every audited page with its URL, anchor text, and link type. We do not probe link status: a dedicated, rate-limited broken-link crawler handles that separately and avoids hammering the origin. Appendix B is a link inventory, not a broken-link list.

Scope: 11 pages examined | Platform: Unknown Platform | Analysis method: Automated checks with expert review | robots.txt: Present (3 directives)

Measurement completeness: Every probe completed during this audit, with no network errors or timeouts. The findings below rest on a full data collection.

What comes next. This report is the foundation, not the finish line. Implementing the recommendations requires the technical knowledge that produced them; we bring that forward. Our implementation engagements begin where this audit ends.

We work toward a site - and an estate of documents beyond it - that any machine can read, trust, and act on. It holds its own dated, attested record for anyone who needs to verify that claim. Reaching it - structured data, discovery files, accessibility, governance metadata, and re-audit on a regular schedule - is available as a managed service or as licensed tooling your team runs independently. We also run training sessions that give development teams the MX vocabulary and implementation patterns directly. To take any of it further, contact CogNovaMX Ltd at info@cognovamx.com.

About This Report

We audited 11 pages across specimen-media.example's site using the Web Audit Suite. We also reviewed the site's discovery files (sitemap.xml, llms.txt). We review each page across ten dimensions: performance (load time, Core Web Vitals), accessibility (WCAG 2.1 AA), SEO, semantic HTML structure, structured data quality, image efficiency, security headers, content consistency, discovery file coverage, and machine pipeline survivability.

We fetch every page twice: as a server-side agent sees it (raw served HTML, no JavaScript) and after full browser rendering. The gap between those two results is the served-versus-rendered gap: the share of content invisible to agents that do not execute JavaScript. Server-side agents, including those behind ChatGPT, Claude, and Perplexity, parse served HTML only.

We then review automated findings by hand before completing this report. The automated pass identifies what is present or absent; we read that against context, distinguishing platform constraints from implementation choices and findings worth acting on from those the platform makes unavoidable. Patterns we see repeatedly across sites on the same platform we note as characteristics of that platform rather than site-specific gaps. When new agent patterns emerge, we update what we look for.

How we build it. We use scripted SOPs running deterministic checks rather than inference. The crawl, the served-versus-rendered comparison, the structured-data extraction, the accessibility passes, the discovery-file probes, the platform fingerprinting and the per-section scoring all run as scripts producing byte-identical outputs on the same input. A small number of stages run a judgement pass over the resulting report; that is the only inference layer. Those judgement passes can run against a local model, so the whole audit runs inside the organisation's own network with nothing leaving it: relevant where content is regulated or privacy-sensitive. Every AI decision made during the audit is recorded in the provenance layer attached to this document - the AI and deterministic evidence sidecars embedded in the PDF. The only connection the audit makes to the internet is fetching the pages of the website being audited. Nothing is sent out.

Our scoring criteria follow published MX standards and proposed specifications maintained at The Gathering. Where established external standards apply: WCAG 2.1, Schema.org, RFC 9309, W3C: those take precedence. MX addresses governance and machine experience metadata in the areas those standards do not cover. The methodology behind every section of this report is documented in full in MX: The Protocols at MX: The Protocols.

What we cover here, and what MX covers. This report looks at the web estate: every page served over HTTP, examined for metadata, structured data, accessibility, and what machines can read. MX runs deeper, covering every document type a business publishes (PDFs, data feeds, API responses, structured documents) and the machines that

read them. The web estate is the foundation; the rest builds on it.

Audit scope. Findings throughout this report describe what we observed on the 11 HTML pages we examined in depth, drawn from a sitemap of 5983 URLs. We also reviewed the site's discovery files (sitemap.xml, llms.txt). Structural findings - a missing header, a soft 404 pattern, a discovery file gap - hold across the full estate and are noted as such. Verdicts scoped to the sampled pages should not be extrapolated to the full estate without a wider audit.

A note on llms.txt

The llms.txt convention places a structured description file at a site's root for AI systems to read, following the same pattern as robots.txt. The Discovery Files section below records its presence, transport type, and sitemap registration, and covers the two structural problems (content type and discovery) that limit most implementations.

Appendix E: Markdown Content Negotiation

Table 31

Appendix E: Markdown Content Negotiation

Check	Result
URL probed	https://specimen-media.example
HTTP status	200
Content-Type returned	text/html; charset=utf-8
Markdown served	No - server returned HTML regardless of Accept header

The site returns standard HTML to all requests, including those carrying `Accept: text/markdown`. Markdown content negotiation is a feature that lets a server deliver a lighter, markup-free page to agents that request it - reducing the parsing load on the agent side. It is an optional enhancement with no compliance obligations attached. One consideration before enabling it: Markdown conversion strips `<head>` metadata, governance fields, and discovery signals, so any page carrying MX fields, canonical URIs, or structured data in the document head would lose those signals for agents that receive the Markdown version. Whether the reduction in parsing cost outweighs that loss is a publisher decision; this probe records the current state.

Further Reading

The reference material cited in this report. Click the link on screen or scan the QR code on paper: both encode the same URL.

Table 32

Further Reading

Scan	Link and description
	Appendix R: Testing Agent Comprehension: the methodology behind the Pipeline Survivability measurements used in this report. https://mx.allabout.network/books/appendices/appendix-r.html
	Appendix S: The Eleven Agent Reading Resilience Checks: the full set of reading-resilience checks scored in the Agent Reading Pipeline section. https://mx.allabout.network/books/appendices/appendix-s.html
	Appendix M: Index of Metadata: the field dictionary governing the MX governance tags referenced in this report. https://mx.allabout.network/books/appendices/appendix-m.html
	Why llms.txt Probably Isn't Working: the two structural problems most llms.txt implementations have (MIME type and sitemap registration). https://mx.allabout.network/blog/llms-txt-guide.html
	The Gathering: the community-led open-standards body that governs the MX metadata standard. https://tg.community
	MX: The Protocols: the practitioner reference covering the scoring methodology and implementation patterns behind this report. https://mx.allabout.network/books/
	The Provenance Gap: the structural distinction between content that describes a claim and content that evidences it - and why machines treat unverified claims differently. https://mx.allabout.network/blog/the-provenance-gap.html
	MX Inspector: drop this PDF into the browser-based inspector to read the full provenance chain, attestation, and machine-readability score - no command-line tools required. https://mx.allabout.network/tools/pdf-inspector.html

This Report's Own Evidence Chain

MX is to machines what UX is to users: it asks not whether a human can read this report, but whether a machine can read it, verify it, and act on it. A standard is credible only when we run on it ourselves, so we built this report to the standard it measures. You can inspect this PDF and read every claim it makes directly in your browser at the MX Inspector - no command-line tools, no login, no installation required. Drop the file, read the chain.

We carry our own provenance. Every step that produced it is recorded in two adjacent JSON sidecars - one AI, one deterministic - and the full evidence chain travels inside the PDF's XMP metadata: extract it with `exiftool -b -XMP-mx:ProvenanceAiPayload specimen-media-report.pdf | jq .`. The PDF is a tagged ISO 14289-1 (PDF/UA-1) Level 2 document with a complete reading-order structure tree. What this audit measures on a client's behalf, this deliverable meets.

Machine-readable content is visible to agents and validators. Machine-trustworthy content adds a provenance layer - a dated, attested record that names who published it and under what rubric. Readable is what MX makes content; the provenance layer is what makes it trustworthy. The two do different jobs, and this report carries both. It is an example of what that looks like in practice.

Practice What We Preach: This Audit's Own Evidence Chain

A standard is credible only when we run on it ourselves. We hold this audit deliverable to the same MX standards we apply to the audited site; consider this working proof of the practice it recommends. Every consequential step that produced this report (LLM-driven prose passes, deterministic gate verdicts, multi-agent attribution probes, repair iterations) is recorded in two adjacent JSON sidecars next to this PDF.

The AI evidence chain records every non-deterministic step: the model identifier, the SHA-256 of the system prompt we ran (so an auditor can verify the rubric we used), the SHA-256 of the output it produced, a short excerpt of the model's reasoning, and the human-intervention state. This chain is designed as evidence for AI-governance regimes: EU AI Act, UK ICO AI guidance, US NIST AI RMF, and Colorado AI Act. The framework citations are claims of relevance, not compliance grants; conformance with each regulation remains a legal duty of the operator. This PDF holds the full AI evidence chain inside its XMP metadata under `xmp:ProvenanceAiPayload`. A regulator inspecting the PDF alone receives the entire chain; the adjacent `*.provenance.ai.json` is a copy of the same JSON for tooling that prefers file access.

The deterministic evidence chain is at `*.provenance.deterministic.json`. It records every rule-driven step: gate verdicts, CSV checks, regex matches, render steps, probe results, and the closing PDF conformance verdict. This chain is designed as evidence for EAA Directive 2019/882 accessibility-conformance. The deterministic file is named in the PDF's XMP metadata under `xmp:ProvenanceCompanion` so an inspector who has the PDF alone can walk to it on disk.

To extract the chain from the PDF, run `exiftool -b -XMP-mx:ProvenanceAiPayload specimen-media-report.pdf | jq .`. The `-b` flag is required so exiftool emits the raw payload; without it the output includes a label that breaks the JSON parse. The two chains share `auditId`, `startedAt`, `operator`, and a `provenance` header naming the exact git commit of the audit tooling that produced this run, so anyone can re-run it and verify byte-for-byte what we did. We prefer determinism to inference: explicit over inferred, recorded over remembered, a result you can reproduce over one we could only explain. Where a check can be made by a rule, a rule makes it, and the rule leaves a record rather than an opinion. That is why this chain shows what we did instead of asking you to trust a summary of it.

Verify this report yourself - no tools, no login, no installation. Drop this PDF into the MX Inspector to read the full provenance chain, operator identity, and attestation in your browser, or run three commands directly against the file:

1. Extract the full AI evidence chain from the PDF: `exiftool -b -XMP-mx:ProvenanceAiPayload specimen-media-report.pdf | jq .`
2. Confirm the operator identity: the JSON contains `operator.name`, `operator.email`, and `operator.organisation` naming the accountable individual.
3. Cross-reference the sidecar: `diff <(jq .auditId specimen-media-report.provenance.ai.json) <(exiftool -b -XMP-mx:ProvenanceAiPayload specimen-media-report.pdf | jq .auditId)`; both should return the same `auditId`.

The evidence does not require trust. It requires tools. That is the difference between a governance claim and a governance record.

The PDF itself is a structured, tagged document. It conforms to ISO 14289-1 (PDF/UA-1) at Level 2 with `pdfuaid:Part=1` declared in the XMP packet and a complete `/StructTreeRoot` with the document's logical reading order. This is the accessibility-conformance grade that the European Accessibility Act (EAA Directive 2019/882) expects of digital documents distributed to citizens of the EU and EEA. Producing the PDF at Level 2 is not a compliance grant; conformance with the EAA remains a legal duty of the operator distributing the document. What the tagged PDF provides is the structural prerequisite the EAA expects: a document a screen reader can traverse in semantic order and a regulator can verify with any conforming PDF/UA validator.

This practice is what MX expects of every artefact in the field. We apply it to ourselves.

This report serves two audiences: the humans who read it, and any machine that encounters it later - a validator checking the provenance chain, an assistant answering a query, or a regulator walking from a compliance clause to the evidence behind it. One deliverable, two audiences, no guessing.

Be Ready

A site audit measures what machines read from your web estate. MX readiness reaches further.

Three audits. One standard. Working together.

Site Audit. This report. What machines read from your web estate: structure, metadata, accessibility, provenance, pipeline survivability. A dated, attested record any third party can verify.

Repository Audit. What agents see inside your codebase. Most repositories were never written for the AI agents, copilots, and build tools now working inside them. The repository audit scores self-description, boundaries, checkability, and provenance - and maps the path to a repo that describes itself, checks its own rules, and repairs its own drift.

Developer Audit. Whether your team is ready. The MX vocabulary, the implementation patterns, and the habits that keep an estate machine-ready as the standard evolves. The goal is a team that holds the practice itself, not a dependency on outside help.

All three audits run locally. No content leaves your infrastructure unless you authorise it. You need all three in sync for MX readiness. Contact CogNovaMX to scope the full picture: info@cognovamx.com

Date: 12 July 2026

This is a sample run over a subset of the site. CogNovaMX Ltd can scope a full-estate audit.