

WHAT YOUR SITE IS ACTUALLY DOING.

Limited report. We could not run a full on-page scan of <https://motion.com>: it sits behind Cloudflare bot protection that blocks automated crawlers. What follows is built from Google's real-user field data (CrUX) for this site, gathered from actual Chrome visitors, which does not require loading the page. The speed and stability numbers below are real; the on-page UX, accessibility, and SEO checks could not run.

These are the Core Web Vitals real Chrome users experience on <https://motion.com>, straight from Google's field data. No login, no insider access, no Harvv pixel needed.

July 17, 2026 · External scan · 2 MOBILE + 2 DESKTOP LOADS · NO PIXEL DATA

We could not load <https://motion.com>. Cloudflare bot protection turned our scanner away, so nothing below should be read as a clean bill of health. It is what we could see from outside. The block itself is the headline finding: whatever stops us almost certainly stops AI crawlers from reading your pages too.

WORKING

What's already working. These held up across every load.

Too little held up to list. Start with the column on the right.

COSTING YOU

Ranked by what hurts conversion most. Full evidence below.

403 Your edge blocked our scanner (HTTP 403) — it is very likely blocking AI crawlers too

FIXING NEXT

AI CRAWLER BLOCKED.

2 HOURS

One focused change, on the pages people actually land on: Your edge blocked our scanner (HTTP 403) — it is very likely blocking AI crawlers too. We re-measure the same samples after.

01

Findings, ranked by what hurts conversion most

SEVERITY	FINDING	HOW WE KNOW
HIGH	Your edge blocked our scanner (HTTP 403) — it is very likely blocking AI crawlers too BOTH We tried to load https://motion.com and Cloudflare bot protection turned us away with HTTP 403, so no on-page scan could run. That matters beyond this report: the same rule that stopped us stops ChatGPT, Perplexity and Claude from reading your pages, which means AI assistants cannot cite you even when you are the best answer. Allow-list the AI crawler user-agents (GPTBot, PerplexityBot, ClaudeBot, Google-Extended) at your edge, then re-run this scan. How we know: median across loads	

FROM FINDING TO FIX

Want the fix, not just the finding?

Install Harvv and we turn each issue above into a ready-to-paste prompt for your AI coding assistant. Drop it into Cursor, Claude, or Copilot and the diagnosis becomes a concrete code change, written against this exact page.

Get the fix prompts

"How we know": **unlabeled** = a deterministic fact, identical on every load (e.g. element sizes). Most findings are this kind, so we only mark the exceptions: **median across loads** = a noisy lab metric, reported as a median. **real-user field data** = Google CrUX, actual Chrome visitors.

02

Performance: phone, desktop, and real visitors

METRIC	MOBILE	DESKTOP	READ
TTFB (lab median)	28 ms	20 ms	LAB
FCP (lab median)	227 ms	260 ms	LAB
LCP (lab median)	227 ms	260 ms	GOOD
Page weight (median)	0.0 MB	0.0 MB	OK

Google Lighthouse (lab): Performance **30** mobile / **55** desktop, SEO **92**, Accessibility **95**, Best Practices **92**.

Lab numbers are from a headless mobile browser on an unthrottled connection: treat them as a floor, not a typical experience. Add a Google API key to light up real-user field data (CrUX) and Lighthouse scores.

03

Tiny buttons are hard to tap on mobile

2 of 2 tappable items on this page come in below 44×44 pixels, the size Apple and Google both recommend for reliable tapping on a phone. The same ones came up small on every one of the 2 test loads, so this is the page itself, not a fluke.

When customers can't tap what they expect to, they get frustrated and many of them leave. They don't file a bug. They don't try again. They just leave. A desktop dashboard can't see this because it's the difference between a thumb and a cursor.

The buttons measuring below the minimum on this scan:

- a 61x18 "Cloudflare"
- a 43x18 "Privacy"

The fix is CSS-only on most sites: add padding around the icon (don't just change the icon size) so the actual tap area is at least 44×44 pixels. No redesign, no new assets.

04

Technical SEO & structured data

CHECK	RESULT
Title	Just a moment... (16 chars)
Meta description	MISSING
H1	1 on page
Canonical	MISSING
Structured data (JSON-LD)	NONE
Open Graph	INCOMPLETE

05

The fix checklist

Everything to fix, priority first, each tagged with the screen it affects and a rough effort. Work top to bottom.

☐ **Your edge blocked our scanner (HTTP 403) — it is very likely blocking AI crawlers too**

BOTH

VARIES

Effort is a rough read from the outside: "CSS only" means no new assets or backend work, "1 line" means a single tag, "Dev afternoon" means a developer needs to touch tracking or scripts.

06

What this report cannot tell you

Everything above is from the outside, looking at the page on a simulated phone and desktop. The questions that actually decide revenue need real visitors. Install the Harvv pixel (one script tag, 16 KB, zero personal data, no engineering project) and within about 72 hours you'd know which buttons real customers tapped and missed, how often Google Analytics is missing visits, and exactly where mobile shoppers stalled and left. This report shows you where to look. The pixel shows you how often it happens, and to whom.

07

How we did this, and what it can't prove

- **2 mobile + 2 desktop loads** of one URL from headless Chrome (iPhone viewport at 390px, desktop at 1366px), July 17, 2026. Enough loads to separate real defects from random noise, not a full-site crawl.
- **Lab numbers, not real-user numbers** (no field data was available for this run). Real devices on real networks run slower.
- **Friction is inferred, not counted.** We can prove a button is small. We can't, from the outside, count how often it causes a missed tap. That requires the pixel on a live page.

About Harvv, the source of this teardown

Harvv is a behavioral UX analytics platform (harvv.com). A lightweight JavaScript pixel captures how real visitors behave on a site (dead clicks, rage clicks, scroll depth, Core Web Vitals, JavaScript errors, and 50+ other signals) and the engine turns them into prioritized, plain-English findings. This teardown is the outside-in version of that: the same detectors run against a public page, with no pixel installed.

How to read it. Every finding here is a reproducible, automated measurement, not an opinion: element sizes, contrast ratios, load metrics, and structured-data checks that anyone can re-run against the same URL. The method is stated in full above. Automated testing catches a subset of issues, so this is a starting point, not a certification.

Full disclosure. Harvv makes the pixel that would measure the friction these findings imply, so we have a commercial interest. That is exactly why the findings are kept to things a reader can verify independently, and why nothing here is inflated: an unreproducible claim would undermine the tool it is meant to demonstrate.

Prepared by Harvv (harvv.com), a behavioral UX analytics platform. Last updated July 17, 2026.