HAD A FARM · RESEARCH REPORT · NOVEMBER 2026

A Clean Scan Isn't Compliance

Web accessibility on 50 Central Arkansas public websites, ahead of the ADA Title II deadline

76%
of homepages had automated WCAG 2.1 failures

4 of 12
"clean" homepages still failed a keyboard test

26%
load an accessibility overlay

Summary

On Nov. 4, 2026, we tested the homepages of 54 public-sector websites in Central Arkansas: cities, counties, school districts, state agencies and boards, public colleges, public authorities and tourism commissions. Fifty loaded for testing. We ran automated WCAG 2.1 checks on all 50, then walked through 16 of them with a keyboard.

- **38 of 50 homepages (76 percent) had at least one automated WCAG 2.1 A or AA failure.** Fourteen had 10 or more individual failures. The worst had 138.
- **12 homepages came back with zero automated failures.** We keyboard-tested every one of them. **At least four of the 12 still fail WCAG 2.1 AA** on problems we confirmed with screenshots: invisible skip links, keyboard focus that disappears, and menus that pull focus into links nobody can see.
- **The same few defects repeat everywhere.** Unnamed links and buttons (22 sites) and low-contrast text (21 sites) account for about half of every failure we recorded. Both usually live in a site's template, so one fix clears them on every page.
- **13 of 50 sites (26 percent) load a commercial accessibility overlay.** Four of the 12 "clean" homepages are among them. On one, the overlay itself added two invisible stops to the keyboard path.
- **Five sites play a looping background video with no way to pause it.**

The compliance date for state entities and larger local governments is **April 26, 2027**, less than six months away. Smaller local governments have until April 26, 2028.

The takeaway for anyone responsible for a public website: an automated scan is a useful first pass and a poor final answer. The failures most likely to lock someone out are the ones a scanner can't see.

The rule and the deadlines

In April 2024 the U.S. Department of Justice published a rule under Title II of the Americans with Disabilities Act that, for the first time, sets a technical standard for state and local government websites and mobile apps: **WCAG 2.1 Level AA**. In April 2026 DOJ issued an interim final rule that moved the compliance dates back a year. The standard did not change.

Who	Compliance date
State government entities, including agencies, boards, commissions and public universities that are part of state government	April 26, 2027
Cities, counties and other local governments with 50,000 or more residents	April 26, 2027
Local governments with fewer than 50,000 residents, and special district governments	April 26, 2028

Two related obligations matter in Arkansas:

- **HHS Section 504.** Health care organizations that receive HHS funding face a parallel web accessibility rule. HHS moved its dates to May 11, 2027 for recipients with 15 or more employees and May 11, 2028 for smaller ones.
- **Arkansas's own law.** Since 1999, Arkansas has required a technology access clause in covered entities' information technology contracts, built around nonvisual access standards (Ark. Code Ann. § 25-26-201 et seq.). It's an existing obligation that's easy to overlook.

The federal rule covers what a public entity provides or makes available online, including content a vendor builds or hosts under contract. It has limited exceptions for archived content, preexisting documents no longer in use, some third-party content and individualized password-protected documents. Each is narrower than it sounds. Read the rule, or get legal advice, before relying on one.

What we tested

We chose 54 sites to represent the public web a Central Arkansas resident actually uses: the region's largest cities and counties, the largest school districts, the state agencies people deal with most, public colleges, regional public authorities and services, and tourism and visitor bureaus.

Four sites could not be tested reliably from our test environment (two returned server errors and two blocked automated browsers), leaving 50.

Sector	Sites tested
State agencies, boards and offices	17
Cities	10
Public colleges and regional public authorities	8
School districts	7
Counties	4
Tourism and visitor bureaus	4
Total	**50**

We don't name individual sites in this report. The patterns matter more than any one site, and nearly every site we tested has something to fix. Full methodology is at the end.

What the automated scan found

Overall

Measure	Result
Homepages with at least one WCAG 2.1 A or AA failure	38 of 50 (76%)
Homepages with 10 or more failure instances	14 of 50
Homepages with at least one "critical" failure	18 of 50
Homepages with zero automated failures	12 of 50
Median failure instances per homepage	4
Highest on a single homepage	138
Total failure instances across all 50	554

A median of four sounds manageable, and for many sites it is. But a homepage is usually the most-polished page on a site, so these are floors, not ceilings.

By sector

Sector	Sites	With failures	Zero failures	Median instances
State agencies and boards	17	12	5	4
Cities	10	8	2	5
Public colleges and regional authorities	8	5	3	1.5
School districts	7	6	1	6
Counties	4	3	1	2
Tourism and visitor bureaus	4	4	0	15

Samples by sector are small, so read this table as a description of these 50 sites, not a ranking of sectors. Two patterns stand out anyway. Tourism sites, the most visual and marketing-driven in the set, all had failures and the highest median. State agencies had the widest range, from five clean homepages to the most failures in the study.

The most common failures

Failure	Sites	Instances	What it means for a user
Text contrast too low	21	140	Text is hard or impossible to read for people with low vision, or anyone on a phone outdoors
Links with no accessible name	21	140	A screen reader announces "link" with no clue where it goes
Images with no text alternative	8	49	The image's content, often the whole message of a graphic, is missing for blind users
Links that rely on color alone	3	33	Links inside paragraphs can't be told apart from plain text without color
Buttons with no accessible name	3	17	A screen reader announces "button" with no clue what it does
Embedded frames with no title	4	4	Maps, videos and forms in frames are unlabeled

Contrast and unnamed links alone make up about half of all 554 failures. Both are usually template problems:

- **Unnamed links are mostly icons.** Social media logos, search magnifiers, hamburger menus and arrow buttons with no text behind them. They sit in the header or footer, so they fail on every page.
- **Low contrast is mostly brand color.** A light accent color for links, labels or buttons that looks fine on a design comp and fails the 4.5:1 ratio in the browser. It repeats everywhere the color is used.

Fix the template and those failures disappear sitewide.

What the scanner missed

Twelve homepages returned zero automated failures. We tested all 12 by keyboard (method below) and confirmed every finding with a screenshot before counting it.

At least four of the 12 fail WCAG 2.1 AA on problems an automated scan doesn't report.

What we found	On how many of the 12	WCAG 2.1 criterion
"Skip to main content" link stays invisible when it receives focus	3	2.4.7 Focus Visible
Main navigation shows no visible focus at all	1	2.4.7 Focus Visible
Tabbing moves into closed drop-down menus, so focus lands on links you can't see	1	2.4.7 Focus Visible; 2.4.3 Focus Order
No way to skip repeated navigation	1	2.4.1 Bypass Blocks
Overlay adds invisible stops to the start of the keyboard path	1	2.4.7 Focus Visible

Three of the remaining eight passed every keyboard check we ran. The other five were inconclusive: a pop-up or chat widget took over keyboard focus during the test, the script couldn't record focus, or it flagged a possible problem we couldn't confirm by screenshot. We didn't count those either way.

Here's what those failures look like from the keyboard.

The invisible skip link. The skip link exists so keyboard users can jump past dozens of navigation links. On three of the clean homepages, pressing Tab once puts focus on the skip link, but it stays off screen or hidden. A sighted keyboard user sees nothing happen and has no idea the link is there. On one of them, the link is positioned 160,000 pixels above the top of the page, even while it has focus.

Focus that vanishes in the menu. On one site, a keyboard user tabbing through the main navigation sees no indicator at all: the page looks exactly the same whether focus is on a menu link or nowhere. Continue tabbing and focus drops into the drop-down menus, which stay closed, so the next several presses land on links that are invisible. To a keyboard user, the page appears frozen.

The overlay in the way. On another site, the first two presses of Tab land on parts of the overlay widget that have no size and no visible presence on the page. The site's own skip links come next, and they're positioned off screen. A keyboard user presses Tab four times before anything visible happens.

Hands-on testing of two other sites in this set, with 10 and 15 automated failures (far from the worst), found more serious problems: a quiz whose answer buttons can't be reached by keyboard at all, an interactive map that responds only to a mouse, and a skip link broken by a one-character typo. None of those showed up in automated results.

Other patterns

Accessibility overlays. 13 of the 50 sites (26 percent) load a commercial overlay script, the kind that adds a floating accessibility button. They span every kind of public body in the study except colleges and regional authorities. Several state agencies use the same product, which suggests a shared template.

An overlay changes how a page appears in the browser. It doesn't fix the underlying code that WCAG conformance is measured against, it can conflict with assistive technology people already use, and as we saw, it can add barriers of its own. Overlays can also rewrite a page as it loads, which may make automated results look better than the site's own code. Four of the 12 zero-failure homepages run one.

Motion with no pause. Five homepages play a background video that loops continuously with no pause button. Anything that moves for more than five seconds needs a way to stop it (WCAG 2.2.2). Other sites in the study got this right with a pause button.

Zoom blocked on phones. Two sites block pinch-to-zoom in their mobile settings, a one-line fix that fails WCAG 1.4.4.

Structure. 14 homepages have no "main" region marked in the code, 10 have no top-level heading, and 7 have no working skip link that we could find. These are best practices more than outright failures, but they're how screen reader users find their way around a page.

Why scanners miss the failures that matter

Automated testing engines read a page's code and check it against rules a machine can decide: does this image have a text alternative, does this text meet the contrast ratio, does this link have a name. They're fast, repeatable and genuinely useful. We used one for this report.

They can't answer the questions that decide whether a person can actually use the page:

- Can a keyboard user reach and operate every control, including custom ones like quizzes, maps, tabs and carousels?
- Is it always visible where keyboard focus is?
- Does the skip link actually skip anywhere?
- Does the alt text describe the image, or does it just exist?
- Can someone pause moving content?
- Does the page still work when the text is zoomed or the screen is narrow?

A homepage can return zero automated failures and still fail several of those. In this study, a third of the clean homepages did.

Five fixes that clear most of these failures

These are template-level changes. Each one fixes every page built on the template.

1. Name every icon link and button. Give social icons, search buttons, menu toggles and arrows an accessible name, and hide the decorative icon from screen readers.

```
<a href="https://www.facebook.com/yourcity" aria-label="City of Example on Facebook">
  <svg aria-hidden="true" focusable="false">…</svg>
</a>
```

2. Fix the brand colors once. Check every text color against its background at the WCAG ratio: 4.5:1 for normal text and 3:1 for large text and interface components. Darken accent colors used for text until they pass, and keep the bright version for decoration only.

3. Make focus visible everywhere, including the skip link. Remove any global `outline: none`, and use a focus style that shows up on both light and dark backgrounds. Make the skip link appear when focused.

```
:focus-visible { outline: 3px solid #ffffff; box-shadow: 0 0 0 6px #1a365c; }
.skip-link:not(:focus) { position: absolute; left: -9999px; }
.skip-link:focus { position: static; }
```

4. Keep closed menus out of the keyboard path. Hidden drop-down links shouldn't receive focus until the menu is open. Use `hidden` or `display: none` on closed menus, or the `inert` attribute, and open them with a real button that reports its state with `aria-expanded`.

5. Give moving content a pause button. Add a visible, keyboard-operable pause control to any background video or auto-advancing carousel, and don't autoplay at all when a visitor has asked their device to reduce motion.

Keyboard self-test (15 minutes, no tools)

Do this on your homepage, your most-used form and any page with an interactive feature. Put the mouse away.

1. **Load the page and press Tab once.** The first thing to get focus should be a "Skip to main content" link, and you should be able to see it.
2. **Press Enter on the skip link, then Tab again.** Focus should land in the main content, not back in the header or menu.
3. **Keep pressing Tab through the whole page.** At every stop, can you see where you are? If focus disappears, even for one link, that's a failure.
4. **Watch for invisible stops.** If you press Tab and nothing visible changes, focus may be on a hidden element, such as a closed menu or an off-screen slide.
5. **Operate every control with the keyboard.** Menus, tabs, accordions, carousels, maps, quizzes, video players and pop-ups. Enter and Space should activate buttons; Escape should close pop-ups; arrow keys often move within menus and tabs.
6. **Check that you can get out.** If Tab keeps cycling inside one widget and won't leave, that's a keyboard trap.
7. **Look for motion.** Anything that moves, blinks or auto-advances for more than five seconds needs a visible pause control.
8. **Zoom to 200 percent** (Ctrl or Cmd and the plus key). Text should grow, and the page should still work without sideways scrolling.
9. **On a phone, pinch to zoom.** If the page won't zoom, the site is blocking it.
10. **Write down what you found, with the date.** A dated record of testing and fixes is your best evidence of good-faith progress.

Sample contract language for web vendors

Many public websites are built and maintained by someone other than the people who edit them: a platform provider, a design firm, an agency or a hosting company. Your compliance plan has to reach that vendor, and the contract is where it starts.

Use this as a starting point for website, hosting, design and content contracts. It is not legal advice. Have your attorney and procurement office adapt it to your rules and to Arkansas's technology access clause requirement.

> **Accessibility.** Vendor shall ensure that all websites, web applications, mobile applications, digital documents and other electronic content it designs, develops, configures, hosts, maintains or provides under this Agreement ("Deliverables") conform to the Web Content Accessibility Guidelines (WCAG) 2.1 Level AA, or a later version that the Agency adopts in writing.
>
> **Testing.** Before acceptance of any Deliverable, and after any change to a template, theme, component or third-party integration, Vendor shall test conformance using both automated tools and manual testing, including keyboard-only operation and testing with at least one screen reader. Vendor shall provide written results identifying any nonconformance.
>
> **Documentation.** Vendor shall provide an Accessibility Conformance Report based on the current Voluntary Product Accessibility Template (VPAT) for each Deliverable, and shall update it when the Deliverable changes materially.
>
> **Remediation.** Vendor shall correct, at no additional cost to the Agency, any nonconformance in a Deliverable within thirty (30) days of notice, or within a schedule the Agency approves in writing for complex issues. Nonconformance in a Deliverable is a material breach if not corrected within that time.
>
> **Authoring.** Vendor shall configure content management systems and templates so that Agency staff can create conforming content, including required fields for text alternatives, heading structure and accessible tables, and shall provide basic accessibility training for Agency content editors.
>
> **Third-party components.** Vendor shall not add third-party scripts, widgets, embeds or overlays to a Deliverable without the Agency's written approval and evidence that the component conforms. Accessibility overlays shall not be used as a substitute for conforming code.
>
> **Warranty.** Vendor's accessibility obligations survive acceptance and continue for the term of this Agreement and any maintenance or hosting period.

What to do before April 2027

1. **Find out who owns your template.** Then get accessibility into that contract.
2. **Run an automated scan, and treat it as a starting list.** It's the quickest way to find unnamed icons, contrast and missing alt text.
3. **Do the keyboard self-test** on your homepage, your top forms and anything interactive.
4. **Fix the template first.** Icon names, brand colors, focus styles, menus and motion clear the most failures for the least work.
5. **Don't buy an overlay as a shortcut.** Put that budget into fixing the code.
6. **Inventory your documents.** PDFs, forms and meeting packets are covered too, and they're often the largest backlog.

7. **Keep a dated record** of what you tested, what you found and what you fixed.

How Had A Farm can help

Had A Farm builds tools for exactly this work. eiSEO scans sites for accessibility and search problems and turns the results into a prioritized fix list. ComplyField is built for ADA Title II programs: it tracks your sites and documents, records manual testing alongside automated results, and produces the dated progress record that a scan alone can't give you.

Had A Farm is an Arkansas company founded by a chief web architect and digital accessibility practitioner with more than 20 years at public universities and an academic medical center.

Methodology

Sites. 54 public-sector homepages in Central Arkansas (Pulaski, Faulkner, Saline, Lonoke and Garland counties, plus state government bodies headquartered in Little Rock), selected to cover the largest public bodies by sector. One site was left out because of a professional conflict of interest. Four could not be tested reliably and were excluded: two returned server errors to our test environment and two presented automated-browser challenges. Results cover the remaining 50.

Automated testing. Run on Nov. 4, 2026, using axe-core 4.10.2 in a headless Chromium browser at a 1280 by 900 pixel viewport. Each homepage was loaded, allowed to settle for up to 14 seconds, then tested against the WCAG 2.0 and 2.1 Level A and AA rule sets only (best-practice rules were excluded from failure counts). "Instances" counts every element that failed a rule; "critical" is axe-core's own impact rating. Results that axe-core marked as needing human review were not counted as failures. Three sites showed a browser-update banner triggered by our test browser's version string; we re-tested them with a current version string.

Keyboard testing. Run on the 12 homepages with zero automated failures and four others for comparison. A scripted browser pressed Tab through the first 35 focus stops of each page. At each stop it captured the focused element's area with and without focus; identical images mean there was no visible focus indicator. It also tested the skip link (visibility, whether its target exists, and where focus goes next), flagged focus landing on hidden or off-screen elements, and looked for clickable elements that can't receive keyboard focus. We calibrated the script against two sites tested by hand the day before, where it reproduced every keyboard finding except one that requires pressing Enter on a custom control. **We counted a keyboard failure only after confirming it with a screenshot or the element's on-page position.** Results interrupted by pop-ups, chat widgets or advertisements were treated as inconclusive.

Overlays and video. Overlays were counted only when the vendor's script was actually requested from the vendor's servers during page load. Background video was counted only when a visible video was confirmed playing, looping, with no native controls and no button named "pause" or "play" anywhere on the page.

Limits. Homepages only. No screen reader testing. Contrast over images and video was not measured. Third-party embeds were not tested separately. Websites change constantly, and results reflect the date tested. This report describes conformance problems, not legal compliance or liability, and it isn't legal advice.

Sources

- Jackson Lewis: DOJ extends public entities' compliance deadline
- EDUCAUSE Review: DOJ and HHS extend web accessibility deadlines to 2027 to 2028
- Ark. Code Ann. § 25-26-204, technology access procurement requirements
- W3C: Web Content Accessibility Guidelines (WCAG) 2.1