Audit Report: Foxway - We enable circular tech – We enable circular tech

Website: https://foxway.com/
Date: 01.11.2026
Audit Coverage: 100% — all sources returned data
Confidence: high

Pages Audited (1 of 1):

- https://www.foxway.com/en/

Summary of results

Overall Score: 48 / 100
Status: **Poor**

Site overall 48 is the mean of 1 page. Performance is the dominant drag: PSI mobile 54 / desktop 52, with lab LCP 6.5 s, FCP 6.54 s and Speed Index 7.11 s on mobile, and desktop TBT actually worse at 1.05 s. CrUX field data confirms real-user pain — mobile TTFB 2626 ms p75 (slow) and FCP 3005 ms p75 (slow) — consistent with the missing Cache-Control header, no page cache/CDN, and a 4.44 s document request. The 37.70 MB page weight is the single worst finding: 35.58 MB of it is three media files, dwarfing the 270 KB of images and 1.28 MB of scripts. Accessibility is comparatively healthy (95 mobile / 92 desktop) but still carries 2 serious axe violations (link-name, aria-command-name) plus heading-order and landmark issues. Security basics returned FAILED — AWSALB cookies lack Secure/HttpOnly/SameSite and CSP is absent, which matters here because the site has both a login surface and upload/user-content surface. W3C reports 84 HTML errors (duplicate IDs, h3 inside role=button, style blocks in body) and the page ships 42 scripts with 5 render-blocking, so the score lands in the Poor band despite clean SEO (100) and good CLS (0.000).

Per-page scores

 Poor · https://www.foxway.com/en/

Score	Performance	Accessibility	Best Practices	SEO	Security
48	54	95	96	100	FAILED

PageSpeed Insights — Mobile vs Desktop

*Lower is worse for Performance; higher is worse for LCP and CLS. Worse value is **bolded**.*

URL	Performance (M / D)	LCP (M / D)	CLS (M / D)
https://www.foxway.com/en/	54 / **52**	**6.54 s** / 2.91 s	0.000 / **0.002**

Optimization Checklist

3 of 8 passing — 3 pass · 2 warn · 3 fail

Item	Status	Detail
Page caching plugin / CDN active	**Fail**	No WordPress cache plugin marker or CDN edge cache detected on the document response.
Response compressed (gzip / brotli)	**Pass**	Document response is compressed with gzip.
Images lazy-loaded	**Fail**	None of the 14 raster images use loading="lazy".
Hero image eagerly loaded	**Pass**	Hero image is eagerly loaded (inferred from DOM order/size — Lighthouse LCP element unavailable).
Hero is a real (not a CSS background-image)	**Warn**	Hero element uses a CSS background-image (no image-set() variants), so the browser always loads the original asset regardless of viewport — there is no srcset equivalent. Move the hero to a real with srcset/sizes (or) so smaller viewports can fetch a smaller file.
Responsive images (srcset /)	**Fail**	10/14 raster images lack srcset and are not inside .
Reasonable number of image sizes	**Pass**	8 distinct srcset widths.

Item	Status	Detail
JS scripts not blocking in	Warn	1 render-blocking script in . Move to footer or add defer/async.

Fixes

Priority 1: Critical

Immediate action — impacts user experience, search rankings, or site safety.

1A. Cut the 35.6 MB of autoplay video/media weight `Performance`

- **Impact:** LCP, FCP, Speed Index, total page bytes
- **Problem:** Browser runtime shows 3 media files totalling 35.58 MB of a 37.70 MB page, and the hero mp4 (homepage-5-1.mp4) failed with net::ERR_ABORTED. Lab LCP is 6.5 s on mobile.
- **Solution:**
  - Transcode the hero video to a modern codec (AV1/H.264) at a much lower bitrate and target <2 MB.
  - Do not autoplay on mobile; gate behind a poster + play button:

```
<video preload="none" poster="/img/hero-poster.webp" muted
playsinline>
  <source src="/media/hero.av1.webm" type="video/webm">
  <source src="/media/hero.mp4" type="video/mp4">
</video>
```

  - Serve a still poster image as the LCP element on small viewports.
  - Fix the aborted request so the browser stops retrying the file.

1B. Enable document caching and a CDN edge cache `Performance`

- **Impact:** TTFB, LCP, FCP
- **Problem:** No `cache-control` header is set on the document, the optimized-web checklist fails "Page caching plugin / CDN active", and `cache-insight` estimates 567 KiB of savings. CrUX TTFB is 2626 ms p75 mobile (slow) and the document request took 4.44 s.
- **Solution:**
  - Add a caching plugin (WP Rocket / LiteSpeed Cache / W3C Total Cache) or front the site with a CDN.

- Send explicit cache headers on the HTML document:

```
Cache-Control: public, max-age=0, s-maxage=600, stale-while-revalidate=86400
```

- Cache immutable static assets for a year:

```
Cache-Control: public, max-age=31536000, immutable
```

- Bypass the page cache for logged-in users and cart/checkout paths.

1C. Defer or remove third-party scripts (reCAPTCHA, GTM, gtag, cookie banner) `Performance`

- **Impact:** TBT, LCP, FCP
- **Problem:** 42 scripts (5 render-blocking, 1.28 MB) include reCAPTCHA at 1.06 MB from gstatic with 169 KB + 162 KB unused, GTM/gtag at 521 KB, and a cookiechimp consent modal producing a 436 ms long task.
- **Solution:**
  - Load reCAPTCHA v3 only when a form is focused/submitted rather than on page load.
  - Consolidate the three Google tags (GTM-TRQB56V, G-1VPLWX8V3J, AW-11485841630) into a single GTM container.
  - Defer the cookiechimp widget and remove the render-blocking `<script src="https://cookiechimp.com/widget/5J4t3Gd.js">` from `<head>`.
  - Audit the 42 scripts for anything not needed on first paint.

1D. Set Secure, HttpOnly and SameSite on the AWSALB cookies `Security`

- **Impact:** Session confidentiality, CSRF exposure
- **Problem:** `AWSALB` is set with HttpOnly=false, Secure=false and SameSite=none; `AWSALBCORS` is HttpOnly=false. Security basics FAILED partly on "Cookies Secure + HttpOnly" (2 of 2 cookies lack flags).
- **Solution:** Configure the load balancer / application to emit:

```
Set-Cookie: AWSALB=...; Path=/; Secure; HttpOnly; SameSite=Lax
Set-Cookie: AWSALBCORS=...; Path=/; Secure; HttpOnly; SameSite=None
```

(SameSite=None requires Secure; use Lax where cross-site is not needed.)

1E. Add a strict Content-Security-Policy `Security`

- **Impact:** XSS mitigation on an authenticated, user-content site

- **Problem:** CSP is missing entirely (advanced checks "Content-Security-Policy present" and "CSP is strict" both fail). Site signals show a login surface and an upload surface, so XSS would be high-impact.
- **Solution:** Deploy a nonce-based policy rather than a flat allowlist:

```
Content-Security-Policy: default-src 'self'; script-src 'nonce-{random}' 'strict-dynamic' https:; object-src 'none'; base-uri 'none'; frame-ancestors 'self'; require-trusted-types-for 'script'
```

  - Roll out in Report-Only first, then enforce.
  - Note the existing console warning "Unrecognized feature: 'web-share'" — the Permissions-Policy header is malformed or absent; fix it alongside.

Priority 2: Important

Essential for compliance, user reach, and search visibility.

2A. Fix the two serious axe violations (link-name, aria-command-name) `Accessibility`

- **Impact:** WCAG 2.4.4, 4.1.2; screen-reader navigation
- **Problem:** axe reports 2 serious violations: two article image links (`article:nth-child(1) > .img--ratio\:4/3 > a` and `article:nth-child(3) > ...`) have no discernible text, and the language dropdown `a[role="button"][data-lang="English"]` has no accessible name.
- **Solution:**
  - Give the image links an accessible name:

```
<a href="/en/case-study/..."><img src="..." alt="Case study: circular IT for Nordic schools"></a>
```

  - Label the language switcher:

```
<a role="button" data-lang="English" aria-label="Change language, current: English">EN</a>
```

2B. Repair heading outline, landmarks and add a skip link `Accessibility`

- **Impact:** WCAG 1.3.1, 2.4.1; document outline
- **Problem:** The page has 2 `<h1>` elements, an h2→h6 skip after "Leading the way in circular tech", an axe `heading-order` and `landmark-unique` violation on `.menu-container` , a `region` violation, and no skip-to-content link.
- **Solution:**

- Keep exactly one `<h1>`; demote the slider `<h1 id="n2-ss-10item1">` to `<h2>`.
- Change the `<h6 id="n2-ss-10item5">` to `<h3>` so levels descend by one.
- Give the two `<nav>` landmarks unique `aria-label`s (e.g. "Main", "Social").
- Wrap `.dialog` and the reCAPTCHA iframe in a landmark.
- Add a skip link as the first focusable element:

```
<a class="skip-link" href="#main">Skip to content</a>
```

2C. Lazy-load below-fold images and add srcset to the 10 that lack it `Performance`

- **Impact:** LCP, total image bytes, initial payload
- **Problem:** None of the 14 raster images use `loading="lazy"`, 10/14 lack srcset and are not inside `<picture>`, and the hero is a CSS background-image (`div.globe-icon.desktop-icon`) so no viewport-specific variant can be fetched.
- **Solution:**
  - Add `loading="lazy" decoding="async"` to every image below the fold; keep the hero `eager` with `fetchpriority="high"`.
  - Add `srcset` / `sizes` to the 10 images missing them (the site already generates 8 widths: 300–1920).
  - Replace the CSS background hero with a real `<img>` + `srcset`:

```
<img src="/img/hero-960.webp" srcset="/img/hero-480.webp 480w,
/img/hero-960.webp 960w, /img/hero-1920.webp 1920w" sizes="100vw"
width="1920" height="900" alt="" fetchpriority="high">
```

2D. Eliminate console errors and failed requests `Best Practices`

- **Impact:** Best Practices score, reliability, third-party trust
- **Problem:** 6 console events and 8 failed requests were captured, including `net::ERR_CERT_COMMON_NAME_INVALID` on `tm.bnxtrk.com`, a CORS block on the reCAPTCHA stylesheet, and `net::ERR_ABORTED` on the homepage mp4. PSI flags `errors-in-console` as failing.
- **Solution:**
  - Remove or fix the `tm.bnxtrk.com` tracker — its TLS certificate does not match the hostname.
  - Stop the aborted media fetch by not autoplaying the hero video.

    - Investigate the reCAPTCHA CORS failure; it usually indicates a stale/blocked reCAPTCHA release or a consent-mode interaction.
    - Re-run the page and confirm zero console errors before shipping.

2E. Reduce the 84 W3C HTML validation errors `Best Practices`

- **Impact:** Markup correctness, accessibility tree, maintainability
- **Problem:** W3C reports 84 errors and 7 warnings, including h3 inside `role="button"` ×9, `meta name` not allowed ×7, `<style>` in body ×5, duplicate IDs (`accordion` , `text-block` , `clip0` , `social-links`), stray `</head>` / `</noscript>` tags, and `top: 15` missing a CSS unit.
- **Solution:**
    - Move the inline `<style class="wpcode-css-snippet">` blocks into the theme stylesheet or a proper enqueued file.
    - Replace `<h3>` inside `role="button"` slider layers with `<span>` + `aria-label` .
    - Make IDs unique (`accordion` , `text-block` , `clip0` , `social-links`).
    - Fix `top: 15;` → `top: 15px;` .
    - Remove the stray `</head>` and the `<img>` inside `<noscript>` in `<head>` .
    - Validate after each theme/plugin change in CI.

Priority 3: Best Practice

Recommended for long-term maintainability.

3A. Add Referrer-Policy, Permissions-Policy, COOP and CORP `Security`

- **Impact:** Defense-in-depth headers
- **Problem:** Referrer-Policy, Permissions-Policy, Cross-Origin-Opener-Policy and Cross-Origin-Resource-Policy are all missing; the console also warns about an unrecognized `web-share` Permissions-Policy feature.
- **Solution:**

```
Referrer-Policy: strict-origin-when-cross-origin
Permissions-Policy: camera=(), microphone=(), geolocation=(), web-share=()
Cross-Origin-Opener-Policy: same-origin
Cross-Origin-Resource-Policy: same-site
X-Permitted-Cross-Domain-Policies: none
```

3B. Fix vague link text, tap targets and structured data `SEO`

- **Impact:** SEO score, mobile usability

- **Problem:** PSI flags `tapTargets` and `structuredData` as failing SEO audits; the page repeats vague link text ("contact us" ×3, "about us" ×2, "news" ×2) and the single JSON-LD block has an unrecognized type.
- **Solution:**
  - Disambiguate repeated anchors with `aria-label` or visible context ("Contact us — sales", "Contact us — support").
  - Increase tap target size to at least 48×48 px with adequate spacing.
  - Replace the JSON-LD block with a valid `Organization` / `WebSite` schema and validate in the Rich Results Test.

3C. Stop disclosing the WordPress/polyfill version and server banner `Best Practices`

- **Impact:** Attack-surface reduction
- **Problem:** Security basics reports "WordPress version 3.15.0 is visible in the page" and the `Server: Apache` header discloses technology.
- **Solution:**
  - Remove `?ver=` query strings from enqueued assets (`wp_enqueue_script` with `null` version).
  - Strip version generators from `<head>` .
  - Set `ServerTokens Prod` and `ServerSignature Off` in Apache.