



https://meridianroasters.com

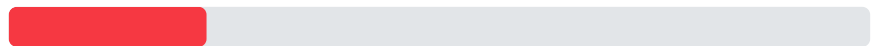
Report generated on Jul 11, 2026



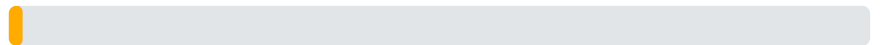
Average SEO score of top 100 sites: 74%

This website received an SEO score of **69 out of 100**, which is below the average score of **74**. However, there are **14 important issues** that need to be fixed to improve your website's ranking on search engines and enhance its overall performance.

14 Failed



1 Warnings



46 Passed



Issues to fix

HIGH

To provide a good user experience, **Google recommends** that sites should aim for a Cumulative Layout Shift score of 0.1 or less.

HIGH

To address URL canonicalization issues, it is recommended to select a primary URL for your webpage and set up redirects from all other variations to the preferred one.

HIGH

To improve the website experience for your visitors, it is recommended to eliminate any render-blocking resources on this webpage.

HIGH

To ensure that Search Engines can accurately identify the topic of this webpage, it is important to include the most common keywords in the title tag, meta description, and heading tags.

HIGH

Build quality backlinks from websites relevant to your topic/niche to improve your search engine rankings.

HIGH

Using images in a modern format can significantly reduce the file size and improve the loading speed of a webpage, providing a better user experience and potentially increasing engagement.

MEDIUM

Serve properly sized images to reduce page loading times and to improve user's experience.



- MEDIUM** Avoid using distorted images, as they can have a negative impact on the user experience.
- MEDIUM** By creating a custom 404 error page with helpful links and information, users are more likely to stay on the site and continue to explore.
- MEDIUM** Add a Google Analytics script to this website to help in diagnosing potential SEO issues by monitoring site visitors and traffic sources.
- LOW** Without an SPF record, spammers can easily spoof emails from this domain, potentially leading to compromised email security and deliverability issues.
- LOW** Consider adding the Strict-Transport-Security header to your webpage to ensure that web traffic is encrypted over HTTPS, mitigating the risk of man-in-the-middle attacks and other security threats.
- LOW** This website either lacks a favicon or it has not been referenced properly.

Common SEO issues

6 Failed 1 Warnings 15 Passed

! Meta Title Test

This webpage is using a title tag with a length of **68 characters**. While there's no target number of characters, titles should be descriptive and concise. We recommend using a title with a length **between 20 - 60 characters** in order to fit Google Search results that have a 600-pixel limit.

Text: Meridian Supply Co. — Small-Batch Specialty Coffee, Roasted to Order
Length: 68 characters

✓ Meta Description Test

This webpage is using a meta description tag.

Text: Single-origin and blended specialty coffee roasted to order in small batches. Ethically sourced beans, shipped within 24 hours of roasting. Free shipping over \$45.
Length: 163 characters



Google Search Results Preview Test

Desktop version

<https://meridianroasters.com/>

Meridian Supply Co. — Small-Batch Specialty Coffee, Roaste...

Single-origin and blended specialty coffee roasted to order in small batches.

Ethically sourced beans, shipped within 24 hours of roasting. Free shipping over...

Mobile version

<https://meridianroasters.com/>

Meridian Supply Co. — Small-Batch Specialty Coffee, Roasted...

Single-origin and blended specialty coffee roasted to order in small batches. Ethically sourced beans, shipped within 24 hours of...



✓ Social Media Meta Tags Test

This webpage is using social media meta tags.

Open Graph Meta Tags

og:type	website
og:url	https://meridianroasters.com/
og:title	Meridian Supply Co. — Small-Batch Specialty Coffee
og:description	Single-origin and blended specialty coffee roasted to order. Ethically sourced, shipped within 24 hours.
og:site_name	Meridian Supply Co.
og:image	https://meridianroasters.com/assets/og-image.jpg

Twitter Card Meta Tags

twitter:card	summary_large_image
twitter:title	Meridian Supply Co. — Small-Batch Specialty Coffee
twitter:description	Single-origin and blended specialty coffee roasted to order. Ethically sourced, shipped within 24 hours.
twitter:image	https://meridianroasters.com/assets/og-image.jpg

i Most Common Keywords Test

There is likely no optimal keyword density (search engine algorithms have evolved beyond keyword density metrics as a significant ranking factor). It can be useful, however, to note which keywords appear most often on your page and if they reflect the intended topic of your page. More importantly, the keywords on your page should appear within natural sounding and grammatically correct copy.

11 stock

11 cart

7 roasted

6 origin

6 colombia



✖ Keywords Usage Test

The most common keywords of this webpage are not distributed across the important HTML tags! Primary keywords should appear in title tag, meta description and heading tags to help Search Engines to properly identify the topic of this webpage.

Keyword	Title tag	Meta description	Headings
stock	✖	✖	✖
cart	✖	✖	✓
roasted	✓	✓	✖
origin	✖	✓	✖
colombia	✖	✖	✖

i Keywords Cloud Test



i Related Keywords Test

There are no indexed keywords for this domain!



Competitor Domains Test

Identifying competitor domains allows you to find domains that are competing with your website for search engine rankings on similar keywords. These domains typically target the same audience and have overlapping goals in terms of organic traffic.

Heading Tags Test

This webpage contains headings tags.

H1 tags

Antioquia

H2 tags

Shop the current harvest

Get the drop list.

Your cart

Robots.txt Test

Congratulations! Your site uses a "robots.txt" file.

<https://meridianroasters.com/robots.txt>

Sitemap Test

This website has a sitemap file.

<https://meridianroasters.com/sitemap.xml>

Image Alt Test

All "img" tags from this webpage have the required "alt" attribute.



Responsive Image Test

Not all images in this webpage are properly sized! This webpage is serving images that are larger than needed for the size of the user's viewport.



How to pass this test?

This test fails when one or more images on the page are served at dimensions significantly larger than the size at which the browser actually renders them. Fixing this issue means resizing the source files (or serving responsive variants) so the bytes downloaded match the pixels actually displayed. The result is faster page loads, lower data usage, and better Core Web Vitals scores.

Example

```

```

Where to make the change

- **Raw HTML:** use `srcset` and `sizes` to let the browser pick the right variant for the user's viewport. Always include intrinsic `width` and `height` attributes.
- **WordPress:** WordPress generates multiple image sizes automatically. Use the `the_post_thumbnail()` helper or the block editor's image block, both of which emit responsive `srcset` markup.
- **Shopify:** use the `image_url` Liquid filter with size parameters (for example `{{ image | image_url: width: 800 }}`) and pair with `image_tag` to emit a responsive image.
- **Wix or Squarespace:** both platforms generate responsive variants automatically; this issue is typically caused by uploading a deliberately oversized image.
- **Headless or framework sites:** use the framework's image component (for example `next/image` or `astro:assets`), which generates responsive variants and lazy-loads off-screen images.

Common causes and how to resolve them

- **Original camera or stock-photo file uploaded as-is:** a 6000-pixel-wide JPEG rendered in a 400-pixel column wastes most of the bytes. Resize the source or generate responsive variants.
- **No `srcset`:** the browser has no choice but to use the single source. Add multiple variants so mobile users get smaller files.
- **Hero images sized for ultra-wide displays:** if your largest layout is 1600 pixels, do not ship a 4000-pixel original.
- **Background images set in CSS without responsive sources:** use the `image-set()` CSS function or media queries to swap in different image files at different breakpoints.

Best practices

- **Generate variants at common widths:** 400, 800, 1200, and 1600 pixels covers most viewports cleanly.
- **Always set width and height:** intrinsic dimensions let the browser reserve space and avoid layout shift, helping CLS.
- **Lazy-load off-screen images:** add `loading="lazy"` to images below the fold so the browser only downloads them when needed.



- **Pair with modern formats:** resizing alone is good; resizing and serving WebP or AVIF compounds the savings.



✖ Image Aspect Ratio Test

Not all image display dimensions match the natural aspect ratio! Fix aspect ratio issues to avoid distorted images on this website!

How to pass this test?

This test fails when one or more images are rendered at an aspect ratio meaningfully different from the source file's native ratio, producing stretched or squashed visuals. Fixing this issue means either using `width` and `height` attributes that match the file's real dimensions or replacing the source with a correctly proportioned crop.

Example

```
<!-- Source file is 1600 x 900 (16:9). Set matching attributes: -->

```

Where to make the change

- **Raw HTML and CSS:** set the `width` and `height` attributes to the source file's native dimensions. Use CSS `height: auto` when scaling responsively so the browser preserves the ratio.
- **WordPress:** when inserting an image via the block editor, choose the image's intended size rather than overriding dimensions in custom CSS.
- **Shopify, Wix, or Squarespace:** use the platform's built-in image cropper so the displayed crop matches the source file you upload.
- **Headless or framework sites:** components like `next/image` require both `width` and `height`, which prevents distortion when the layout scales.

Common causes and how to resolve them

- **Hard-coded width and height that do not match the file:** recompute them from the source dimensions or remove them and rely on the file's intrinsic ratio.
- **Container has a fixed height that crops the image:** use CSS `object-fit: cover` if cropping is intentional, or change the container so the image keeps its natural ratio.
- **Wrong source file uploaded:** swap in a correctly proportioned crop rather than stretching the wrong file in CSS.

Best practices

- **Always set both `width` and `height`:** the browser uses these to reserve space and avoid layout shift before the image loads, which helps CLS.
- **Use `object-fit` when cropping is intentional:** `cover` fills the container without distortion; `contain` letterboxes to preserve the full image.
- **Standardize on a few aspect ratios:** picking one or two ratios per template makes it easier to crop assets consistently and avoid one-off distortions.



✔ Deprecated HTML Tags Test

This webpage does not use HTML deprecated tags.



Google Analytics Test

A Google Analytics script is not detected on this page. While there are several tools available to monitor your site's visitors and traffic sources, Google Analytics is a free, commonly recommended program to help diagnose potential SEO issues.



How to pass this test?

Google Analytics 4 is a free measurement platform that records visitor behavior, traffic sources, and conversion data. Installing it does not directly affect rankings, but the data it produces is essential for measuring SEO performance, identifying high-value pages, and prioritizing future optimization work. Fixing this issue means installing the GA4 tag site-wide and confirming events appear in your reports.

Example

```
<!-- Place inside <head> on every page -->
<script async src="https://www.googletagmanager.com/gtag/js?id=G-XXXXXXXXXX"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());
  gtag('config', 'G-XXXXXXXXXX');
</script>
```

Where to make the change

- **Raw HTML:** paste the GA4 snippet into the `<head>` of every page. A shared layout file is the cleanest place.
- **WordPress:** use a site-kit or analytics plugin, or paste the snippet into your theme's header file. Avoid duplicating the tag across multiple plugins.
- **Shopify:** add the GA4 measurement ID under Online Store, Preferences, Google Analytics, or install the official Google channel app.
- **Wix or Squarespace:** both platforms offer a GA4 measurement ID field in their analytics settings panel.
- **Headless or framework sites:** install via Google Tag Manager or use the framework's analytics integration so the tag fires on every route change.

Common causes and how to resolve them

- **Tag not installed at all:** install GA4 using the snippet above or the platform-specific path.
- **Universal Analytics tag still in place:** Universal Analytics stopped processing data on July 1, 2024. Replace any UA tag (`UA-...`) with a GA4 measurement ID (`G-...`).
- **Tag fires only on the homepage:** ensure the snippet is in a shared template loaded by every page, not pasted into one post.
- **Cookie banner blocks the tag before consent:** integrate Consent Mode v2 so GA4 receives anonymized signals before consent and full data after.
- **Tag duplicated:** two snippets on the same page double-count sessions. Audit with the Tag Assistant browser extension and remove duplicates.

Best practices

- **Verify with real-time reports:** after installing, open GA4's Realtime view and load the site to confirm events arrive.
- **Use Google Tag Manager:** for any site beyond the simplest, GTM gives you a single place to manage analytics, conversion tags, and third-party scripts without code changes.



- **Configure key events:** mark conversions (purchase, lead, signup) as key events so reports highlight what matters for SEO ROI.
- **Link to Search Console:** connecting GA4 to Search Console surfaces organic queries alongside on-site behavior in the same reports.



✖ Favicon Test

This website either doesn't have a favicon or this has not been referenced correctly!

How to pass this test?

A favicon is the small icon associated with the site, shown in browser tabs, bookmarks, history listings, and increasingly in search engine results. Fixing this issue means publishing a favicon at the conventional location and declaring it in the page `<head>` so browsers and search engines can find and display it.

Example

```
<head>
<link rel="icon" href="/favicon.ico" sizes="any">
<link rel="icon" href="/favicon.svg" type="image/svg+xml">
<link rel="apple-touch-icon" href="/apple-touch-icon.png">
</head>
```

Where to make the change

- **Raw HTML:** upload `favicon.ico` to the site root and add the `<link rel="icon">` tags in the `<head>`.
- **WordPress:** upload the icon under Appearance, Customize, Site Identity, Site Icon. WordPress will emit the correct link tags.
- **Shopify:** upload the icon under Online Store, Themes, Customize, Theme settings, Favicon.
- **Wix or Squarespace:** both expose a favicon upload field in the site or design settings.
- **Headless or framework sites:** place the favicon files in the public or static directory so they ship at the root paths.

Common causes and how to resolve them

- **No favicon at the root:** add at least `/favicon.ico`. Most browsers fall back to fetching this path even when no link tag is present.
- **Wrong MIME type:** if the favicon is served as `text/html` instead of `image/x-icon` or `image/svg+xml`, browsers will not render it. Check the response in the network tab.
- **Cached old icon after replacement:** browsers cache favicons aggressively. Append a query string (`?v=2`) to bust the cache.
- **Missing high-resolution variants:** modern devices need 180 x 180 (Apple), 192 x 192 (Android), and 512 x 512 (PWA) sizes for crisp rendering.

Best practices

- **Provide an SVG version:** SVG favicons scale crisply at any size and support light or dark scheme variants.
- **Include an Apple touch icon:** a 180 x 180 PNG named `apple-touch-icon.png` at the root makes the site look correct when added to an iOS home screen.
- **Use a consistent visual identity:** the favicon should match the brand mark so users recognize the site at a glance in tabs and bookmarks.
- **Test in both light and dark mode:** some browsers render the tab background differently. Make sure the icon stays legible in both.



❌ Backlinks Test

This domain has 14 backlinks from 10 referring domains and a rank of 0/1000 while the average rank of Top 100 sites is 895/1000.

✅ JS Error Test

There are no severe JavaScript errors on this webpage.

✅ Console Errors Test

This webpage doesn't have any warnings or errors caught by the Chrome DevTools Console.

✅ Charset Declaration Test

This webpage has a character encoding declaration.

```
Content-Type: text/html; charset=utf-8
```

Speed optimizations

3 Failed

0 Warnings

17 Passed

✅ HTML Page Size Test

The size of this webpage's HTML is 6.4 Kb and is under the average webpage's HTML size of 33 Kb. Faster loading websites result in a better user experience, higher conversion rates, and generally better search engine rankings.

✅ DOM Size Test

The Document Object Model (DOM) of this webpage has 396 nodes which is less than the recommended value of 1,500 nodes.

✅ HTML Compression/GZIP Test

This webpage is successfully compressed using br compression on your code. The HTML code is compressed from 28.14 Kb to 6.4 Kb (77% size savings). This helps ensure a faster loading webpage and improved user experience.



✓ Site Loading Speed Test

The loading time of this webpage (measured from N. Virginia, US) is around **0.53 seconds** and this is under the average loading speed which is **5 seconds**.

✓ JS Execution Time Test

The JavaScript code used by this webpage is executed in less than **2 seconds**.



✔ Page Objects Test

This webpage has less than 20 http requests. A higher number of http requests results in a user's browser needing to request a large number of objects from the server, which will ultimately slow down the loading of your webpage.

Content size by content type

Content type	Percent	Size
Image	84.2 %	910.35 Kb
Font	12.3 %	133.34 Kb
Javascript	2.2 %	23.32 Kb
Css	1.0 %	10.64 Kb
Html	0.3 %	3.57 Kb
Other	0.0 %	460 B
TOTAL	100%	1.06 Mb

Requests by content type

Content type	Percent	Requests
Image	42.1 %	8
Javascript	21.1 %	4
Font	15.8 %	3
Css	10.5 %	2
Html	5.3 %	1
Other	5.3 %	1
TOTAL	100%	19

Content size by domain

Domain	Percent	Size
meridianroasters.com	86.5 %	935.76 Kb
fonts.gstatic.com	12.3 %	133.34 Kb
static.cloudflareinsights.com	1.0 %	11.31 Kb



fonts.googleapis.com	0.1 %	1.27 Kb
TOTAL	100%	1.06 Mb

Requests by domain

Domain	Percent	Requests
meridianroasters.com	73.7 %	14
fonts.gstatic.com	15.8 %	3
fonts.googleapis.com	5.3 %	1
static.cloudflareinsights.com	5.3 %	1
TOTAL	100%	19

✓ CDN Usage Test

This webpage is serving all images, javascript and css resources from CDNs.



Modern Image Format Test

This webpage is not serving images in a modern format! Image formats like [JPEG 2000](#), [JPEG XR](#), and [WebP](#) often provide better compression than PNG or JPEG, which means faster downloads and less data consumption.



How to pass this test?

This test fails when images are served in legacy formats such as JPEG and PNG instead of modern alternatives like WebP or AVIF. Modern formats compress significantly better at equivalent visual quality, which means smaller transfers, faster page loads, and lower data usage for visitors on metered connections. Fixing this issue means generating modern variants and serving them to browsers that support them.

Example

```
<picture>
  <source srcset="/images/hero.avif" type="image/avif">
  <source srcset="/images/hero.webp" type="image/webp">
  
</picture>
```

Where to make the change

- **Raw HTML:** use the `<picture>` element with multiple `<source>` entries so the browser picks the best supported format.
- **WordPress:** WordPress 6.x and later supports WebP natively. Use a media optimization plugin to convert existing JPEGs and PNGs and serve modern formats automatically.
- **Shopify:** Shopify's image CDN can serve WebP automatically; using the `image_url` filter with `format` parameter or the `image_tag` helper handles negotiation.
- **Wix or Squarespace:** both platforms convert and serve WebP automatically when supported by the visitor's browser.
- **Headless or framework sites:** components like `next/image` and `astro:assets` generate AVIF and WebP variants at build time and serve the best format per request.

Common causes and how to resolve them

- **Only legacy formats published:** add a build step or image CDN that generates WebP and AVIF alongside the original.
- **Single-format `<img>` tag:** wrap the image in `<picture>` with `<source>` entries for AVIF and WebP, falling back to the JPEG.
- **Image CDN not negotiating format:** enable automatic format conversion or pass a `format=auto` query parameter (depends on provider).
- **SVG used where a raster image would be smaller (or vice versa):** illustrations and icons should be SVG; photos should be raster in a modern format. Mixing them up wastes bytes.

Best practices

- **AVIF first, WebP fallback, JPEG last:** AVIF compresses best, WebP is widely supported, JPEG remains the universal fallback.
- **Match quality to use case:** hero images can use higher quality settings; thumbnails can drop quality without visible loss.
- **Use an image CDN:** services like Cloudflare Images, Cloudinary, or Imgix handle resizing, format conversion, and caching in one step.



✓ Image Metadata Test

This webpage is not using images with large metadata.

✓ Image Caching Test

This website is using cache headers for images and the browsers will display these images from the cache.

✓ JavaScript Caching Test

This webpage is using cache headers for all JavaScript resources.

✓ CSS Caching Test

This webpage is using cache headers for all CSS resources.

✓ JavaScript Minification Test

All JavaScript files used by this webpage are minified.

✓ CSS Minification Test

All CSS resources used by this webpage are minified.



Render Blocking Resources Test

This webpage is using render blocking resources! Eliminating render-blocking resources can help this webpage to load significantly faster and will improve the website experience for your visitors.



How to pass this test?

This test fails when JavaScript or CSS resources block the browser from rendering visible content. Each render-blocking file delays the first paint, and on slower networks the cumulative impact can be severe. Fixing this issue means inlining the truly critical CSS, deferring or asynchronously loading non-essential scripts, and removing unused code.

Example

```
<head>
<!-- Inline only above-the-fold critical CSS -->
<style>/* critical styles here */</style>

<!-- Load full stylesheet asynchronously -->
<link rel="preload" href="/styles/main.css" as="style"
      onload="this.onload=null;this.rel='stylesheet'">

<!-- Defer non-critical scripts -->
<script src="/scripts/app.js" defer></script>
</head>
```

Where to make the change

- **Application code or templates:** identify the CSS and JavaScript needed for the initial render and load only those synchronously. Defer the rest.
- **Build pipeline:** use a critical CSS extractor (such as `critical` or `penthouse`) to generate the above-the-fold styles automatically.
- **WordPress:** performance plugins offer "delay JS execution" and "load CSS asynchronously" options that handle most of this without code changes.
- **Headless or framework sites:** use the framework's preload and code-splitting hints to control what blocks rendering.

Common causes and how to resolve them

- **Single large stylesheet blocking render:** inline critical CSS for the first viewport, load the full stylesheet asynchronously.
- **Synchronous third-party scripts in the head:** add `defer` or `async`, or move them to the end of the body.
- **Web fonts blocking text rendering:** add `font-display: swap` in `@font-face` rules so the fallback font shows immediately.
- **Render-blocking @import in CSS:** replace with separate `<link rel="stylesheet">` tags so the browser can fetch them in parallel.

Best practices

- Use `defer` for scripts that need the DOM: they execute after parsing but before `DOMContentLoaded`.
- Use `async` for fully independent scripts: they execute as soon as they download, in any order.
- **Preload the critical request chain:** tell the browser early about fonts and key assets with `<link rel="preload">`.



- **Audit with Lighthouse:** the Render-blocking resources audit lists every offending file with the estimated time saved by deferring it.

✓ URL Redirects Test

This URL doesn't have any redirects (which could potentially cause site indexation issues and site loading delays).

✓ Largest Contentful Paint Test

The Largest Contentful Paint duration of this webpage is 1.21 seconds. To provide a good user experience, [Google recommends](#) that sites should strive to have Largest Contentful Paint of 2.5 seconds or less.

Largest Contentful Paint element within the viewport:

```
x.id==='ant...")>
```



Cumulative Layout Shift Test

The CLS score of this webpage is **0.7534**. To provide a good user experience, [Google recommends](#) that sites should strive to have a CLS score of **0.1** or less.

DOM element which contributes the most to CLS score:

Text: MERIDIAN SUPPLY CO · DIRECT TRADE · ROASTED TO ORDER · LOT №07 · FEATURED COLOMB...

Html: `<main id="main">`

Score: 0.7534



How to pass this test?

Cumulative Layout Shift (CLS) measures the total amount that visible elements move around unexpectedly while the page loads. Layout shifts cause users to misclick and feel the page is unstable, and CLS is one of Google's Core Web Vitals and a direct ranking factor. Fixing this issue means reserving space for everything that loads after first paint so nothing pushes existing content around.

Where to make the change

- **Application code or templates:** add explicit **width** and **height** (or aspect-ratio CSS) to every image, video, and embed so the browser can reserve space before the asset loads.
- **CSS:** use **aspect-ratio** on responsive containers; use **min-height** on slots that load content asynchronously (ads, embeds, recommendations).
- **Cookie consent and overlays:** render banners as fixed overlays rather than inserting them at the top of the document.

Common causes and how to resolve them

- **Images without dimensions:** add **width** and **height** attributes (or set **aspect-ratio** in CSS) so the browser reserves the slot before the image loads.
- **Late-loading web fonts:** use **font-display: swap** with **size-adjust** and matched fallback metrics so swapping the font does not change line heights.
- **Aggressively injected cookie consent banners:** render them as a fixed overlay on top of the layout, not as a block element pushing content down.
- **Ads or embeds inserted without reserved space:** reserve a fixed-size container for each slot so the surrounding content does not jump.
- **Animations that move layout-affecting properties:** animate **transform** and **opacity** instead of **top**, **left**, or **height**.

Best practices

- **Target a CLS score of 0.1 or less at the 75th percentile:** the official Core Web Vitals threshold for a "good" experience.
- **Reserve space for everything:** if it loads after first paint, give it a fixed slot so it cannot push content.
- **Animate **transform** only:** transform-based animations do not trigger layout and therefore do not produce layout shift.
- **Test on mobile first:** small viewports amplify the perceived impact of layout shifts.



Server and security

2 Failed

0 Warnings

5 Passed

URL Canonicalization Test

<https://meridianroasters.com/> and <https://www.meridianroasters.com/> should resolve to the same URL, but currently do not.



How to pass this test?

✓ SSL Checker and HTTPS Test

This test fails when different URL variants of the site (apex versus www, HTTP versus HTTPS, with or without trailing slash) do not all resolve consistently to a single canonical version. When the same content is reachable at multiple URLs, search engines may split ranking signals across the variants. Fixing this issue

means picking one canonical form for the site and 301-redirecting every other variant to it. See the

Google Search Central canonicalization guide for the full specification.

✓ The certificate is not used before the activation date.

✓ The certificate has not expired.

Where to make the change

✓ The hostname "meridianroasters.com" is correctly listed in the certificate.

• **Server configuration:** add 301-redirects in your web server config (Nginx, Apache, IIS) or via `_redirects` on Netlify, `vercel.json` on Vercel, or Cloudflare Page Rules.

✓ The certificate should be trusted by all major web browsers.

• **CDN:** most CDNs offer redirect rules that fire before the origin is hit, which is the cheapest way to enforce canonicalization.

✓ The certificate was signed with a secure hash.

• **Application code:** as a backstop, applications can detect the wrong host or scheme and emit a 301 redirect to the canonical URL.

Common causes and how to resolve them

Server certificate

• **Both apex and www serve the same content:** pick one (most sites prefer apex today) and 301-redirect the other to it permanently.

Common Name

meridianroasters.com

• **HTTP and HTTPS both serve content:** redirect all HTTP requests to HTTPS, then add HSTS so browsers stop trying HTTP at all.

Subject Alternative Names (SANs)

meridianroasters.com

• **Trailing slash inconsistency:** pick one form (with or without) and redirect the other.

Not Valid Before

Wed, July 10th 2020, 11:46:08 am (UTC)

• **Uppercase or mixed-case URLs:** redirect uppercase paths to their lowercase canonical form.

Not Valid After

Thu, October 8th 2026, 12:46:06 pm (UTC)

• **Index page accessible at multiple paths:** for example `/index.html`, `/index.php`, and `/` all serving the same content. Redirect everything to `/`.

Signature Algorithm

ecdsaWithSha256

Best practices

Issuer

WE1

• **Use 301 (permanent) redirects:** they pass the strongest ranking signal and are cached aggressively by browsers.

Root certificate

• **Pair with `<link rel="canonical">`:** the canonical tag is a backup signal for cases the redirect cannot cover (parameterized URLs, syndicated content).

Common Name

WE1

• **Avoid chains:** redirect directly to the final URL in one hop.

Organization

Google Trust Services

• **Update internal links to canonical URLs:** linking to non-canonical variants triggers unnecessary redirects on every visit.

Location

US

Not Valid Before

Wed, December 13th 2023, 9:00:00 am (UTC)

Not Valid After

Tue, February 20th 2029, 2:00:00 pm (UTC)

Signature Algorithm

ecdsaWithSha384

Issuer

GTS Root R4



✓ Mixed Content Test (HTTP over HTTPS)

This webpage does not use mixed content - both the initial HTML and all other resources are loaded over HTTPS.

✓ HTTP2 Test

This webpage is using the HTTP/2 protocol.

✓ Plaintext Emails Test

This webpage does not include email addresses in plaintext.

✓ Unsafe Cross-Origin Links Test

This webpage is not using `target="_blank"` links without `rel="noopener"` or `rel="noreferrer"` attribute.

Mobile usability

0 Failed

0 Warnings

3 Passed

✓ Meta Viewport Test

This webpage is using a viewport meta tag.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

✓ Media Query Responsive Test

This webpage is using CSS media queries, which is the base for responsive design functionalities.



i Mobile Snapshot Test



Advanced SEO

3 Failed

0 Warnings

6 Passed

✓ Structured Data Test

This webpage is using structured data.



✖ Custom 404 Error Page Test

This website is not using a custom 404 error page! Default 404 error pages result in a poor experience - it can mislead users into thinking an entire site is down or broken, greatly increases the chance they leave the website entirely, and looks unprofessional. We recommend to have a custom 404 error page in order to improve the website's user experience by letting users know that only a specific page is missing/broken (and not the entire site), providing them helpful links, the opportunity to report bugs, and potentially [track the source of broken links](#).

How to pass this test?

This test fails when the site does not serve a branded, helpful page for missing URLs. A custom 404 keeps users on the site by offering navigation, search, and suggested links rather than dropping them at the browser's generic error screen. Fixing this issue means designing a 404 template, ensuring it returns the correct HTTP 404 status, and pointing your server at it.

Where to make the change

- **Server configuration:** point your web server at a custom 404 page. In Nginx, use `error_page 404 /404.html;`. In Apache, use `ErrorDocument 404 /404.html`.
- **WordPress:** create a `404.php` template in your theme. WordPress invokes it automatically for missing URLs.
- **Shopify:** edit the `404.liquid` template in the theme code editor.
- **Wix or Squarespace:** use the platform's built-in 404 page editor in site settings.
- **Headless or framework sites:** Next.js uses `not-found.js`, Astro uses `404.astro`, and similar frameworks all have a convention for the missing-route page.

Common causes and how to resolve them

- **No custom 404 at all:** create one. The minimum is the site header, navigation, a clear "page not found" message, and links to popular sections.
- **404 page returns HTTP 200:** a "soft 404" confuses search engines. Verify the response status with DevTools or curl, and configure the server to return 404 for missing routes.
- **404 page redirects to homepage:** redirecting users to the homepage hides broken URLs from analytics and search engines. Show a real 404 instead.
- **Generic browser error visible:** the server is not invoking the custom template. Check the configuration directive.

Best practices

- **Keep the site chrome:** show the header, footer, and navigation so users do not feel lost.
- **Offer next steps:** a search box, popular links, or category navigation help users get back on track.
- **Track 404s:** log missing-URL hits in analytics so you can spot broken inbound links worth fixing or redirecting.
- **Be clear about the error:** "Sorry, that page does not exist" is better than vague messaging that confuses users.



✓ Noindex Tag Test

This webpage does not use the noindex meta tag. This means that it can be indexed by search engines.

✓ Canonical Tag Test

This webpage is using the canonical link tag. This tag specifies that the URL: <https://meridianroasters.com/> is preferred to be used in search results. Please ensure that this specification is correct, as canonical tags are often hard-coded and may not always reflect the latest changes in a site's URL structure.

```
<link href="https://meridianroasters.com/" rel="canonical"/>
```

i Nofollow Tag Test

This webpage does not use the nofollow meta tag. This means that search engines will crawl all links from this webpage.

i Disallow Directive Test

Your robots.txt file includes a disallow command which instructs search engines to avoid certain parts of your website! You are advised to confirm if access to these resources or pages are intended to be blocked (e.g., if they contain internal-only content or sensitive information).

✓ Meta Refresh Test

This webpage is not using a meta refresh tag.



✖ SPF Records Test

This DNS server is not using an SPF record! SPF (Sender Policy Framework) allows administrators to specify which hosts are allowed to send mail from a given domain by creating a specific SPF record or TXT record in the Domain Name System (DNS). You can find more information about SPF records [here](#).

How to pass this test?

The Sender Policy Framework (SPF) DNS record lists the mail servers authorized to send email on behalf of your domain. Receiving servers use it to verify that messages claiming to come from your domain are legitimate, which significantly reduces the chance of email being marked as spam. Fixing this issue means publishing an SPF record in your DNS that includes every legitimate sender and ends with a strict policy.

Example

```
example.com. TXT "v=spf1 include:_spf.google.com include:mailgun.org ~all"
```

Where to make the change

- **DNS provider:** add a TXT record at the domain root with the SPF policy. Cloudflare, Route 53, Namecheap, GoDaddy, and similar all expose a TXT record editor.
- **Email providers:** Google Workspace, Microsoft 365, and most transactional email services publish their required SPF include directives in their setup documentation.

Common causes and how to resolve them

- **No SPF record at all:** add one. Even a minimal record (`v=spf1 include:_spf.google.com ~all`) is better than nothing.
- **Multiple SPF records:** SPF only allows one TXT record per domain. Merge multiple records into a single one with all required `include:` directives.
- **Policy too permissive:** ending with `+all` authorizes anyone to send as your domain. Use `~all` (soft fail) or `-all` (hard fail) instead.
- **Missing legitimate senders:** add an `include:` for every transactional email service (SendGrid, Mailgun, Postmark) and every marketing platform you use.
- **Too many DNS lookups:** SPF caps include lookups at ten. Flatten or consolidate `include:` chains if you exceed it.

Best practices

- **Pair SPF with DKIM and DMARC:** SPF alone is not enough. DKIM signs messages cryptographically; DMARC tells receivers what to do when SPF or DKIM fails.
- **Start with `~all`, move to `-all`:** soft fail lets you observe before strict enforcement. Move to hard fail once you are confident no legitimate sender is missing.
- **Test deliverability:** tools such as MXToolbox SPF check or Google Postmaster Tools surface SPF problems before users complain.
- **Audit annually:** as you add or remove email vendors, the SPF record needs to keep up.



✖ Ads.txt Validation Test

The request of ads.txt file has an unexpected Content-Type header: `text/html; charset=utf-8`. In order for this resource to be easily accessed by the DSPs and advertisers, its Content-Type header should be `text/plain` or `text/plain; charset=utf-8`.

How to pass this test?

The ads.txt file is a public list of the advertising vendors authorized to sell your inventory programmatically. A correctly formatted ads.txt protects against domain spoofing in the ad supply chain and is required for inclusion in many premium advertising marketplaces. Fixing this issue means publishing a valid ads.txt at the site root with current authorized seller entries.

Example

```
google.com, pub-0000000000000000, DIRECT, f08c47fec0942fa0
example-ssp.com, 12345, RESELLER
contact=ads@example.com
```

Where to make the change

- **Raw HTML or static site:** upload `ads.txt` to the site root so it is served at `https://example.com/ads.txt`.
 - **WordPress:** use a small plugin or upload the file via FTP. Some advertising plugins manage it automatically based on connected accounts.
 - **Shopify:** upload the file through the theme code editor as an asset, then expose it at the root via a route configuration.
 - **Cloudflare or similar CDN:** serve ads.txt from a worker if you do not have file-system access at the origin.
- Common causes and how to resolve them

- **File missing:** create one with at least your direct advertising relationships.
- **File at the wrong path:** ads.txt must be at the domain root, not in a subdirectory.
- **Wrong relationship value:** use `DIRECT` for direct seller relationships, `RESELLER` for authorized resellers. Other values are invalid.
- **Stale entries:** partners change; remove relationships you no longer have so spoofers cannot exploit them.
- **Returning HTML instead of plain text:** the file must be served as `text/plain`. A page that returns HTML or a 404 fails validation.

Best practices

- **Audit quarterly:** stale entries are the main source of ads.txt vulnerabilities.
- **Mirror app-ads.txt for mobile inventory:** apps use a parallel `app-ads.txt` file with the same syntax.
- **Add a contact line:** include a `contact=email@example.com` line so vendors can reach the right person about ad-quality issues.

