

# Accessibility Audit Report

[jeffrystein.com](https://jeffrystein.com)

<https://jeffrystein.com>

Technical Accessibility Findings & Remediation Report

Accessibility Risk Score

**25**

out of 100

Report Date: 2026-04-12  
Standard: WCAG 2.1 Level AA  
Prepared by: ADA Audit Report

Canonical audit: cacd084c-49ea-45f2-bb0a-6279df4510e2  
Pages scanned: 2

# Executive Summary

---



This audit identified 6 underlying accessibility issues affecting 44 instances across 2 pages on jeffrystein.com.

These are not independent failures — they are systemic issues originating from templates, components, or design tokens that propagate across multiple pages. Most sites resolve 70–90% of instances by fixing a small number of templates or components.

3 originate in site-controlled markup. 3 appear associated with third-party markup — verify ownership before assigning.

## Violations by Severity

Critical		2
Serious		42
Moderate		0
Minor		0

**Total: 44 violations across 6 rule categories**

## Engineering Impact Summary

This audit identified 6 systemic issues. Most originate from shared templates and components, meaning a small number of targeted fixes will resolve the majority of instances.

### Estimated remediation scope:

- 1 template-level fix — resolves ~71% of site-controlled instances
- 2 page-specific fixes — isolated to individual pages
- 3 vendor issues — requires vendor contact, not developer time

### Recommended execution order:

1. Fix template-level issues first (highest instance resolution per fix)
2. Fix component-level issues (shared across pages)
3. Address page-specific issues
4. Contact vendors for third-party issues

Key insight: You are fixing systems, not pages. Effort is significantly lower than instance counts suggest.

## Confidence Distribution

Confirmed	2
Likely Issue	3
Needs Manual Verification	1

## Ownership Distribution

Site-controlled / likely site-controlled	3
Likely third-party	3

## Top Priority Findings

Some findings below are verified technical failures. Others are likely issues or require manual review. These categories are separated throughout the report.

### Verified high-priority findings (2):

- Images must have alternative text (YouTube Embed Player) — 1 instance (CRITICAL)
- Elements must meet minimum color contrast ratio thresholds — 29 instances (SERIOUS)

### Third-party / vendor-attributed findings (3):

- Buttons must have discernible text (YouTube Embed Player) — 1 instance
- Elements must only use permitted ARIA attributes (YouTube Embed Player) — 1 instance
- Frames must have an accessible name (YouTube Embed Player) — 1 instance

## How This Audit Was Produced

This report combines automated accessibility testing (axe-core engine) with rule-level adjudication of findings from the rendered page output. Each finding is classified by severity, confidence, ownership, and priority. Automated testing detects approximately 30-40% of WCAG issues. This report is a technical remediation document and should be used as a prioritized engineering worklist, not as a complete determination of accessibility compliance or a legal opinion.

Accessibility risk score: The score is a weighted technical risk indicator, not a legal determination of ADA compliance. Critical issues carry a 15-point penalty per instance, serious issues 8 points, moderate issues 3 points, and minor issues 1 point. 60% of the final score reflects the weighted penalty total; 40% reflects the pass/fail ratio across all automated checks.

### What This Report Is

- WCAG 2.1 AA-aligned technical audit
- Prioritized remediation plan with developer-ready fixes
- Ownership-classified findings (your code vs. third-party)

### What This Report Is Not

- Not a legal opinion or compliance certification
- Not a substitute for manual accessibility testing
- Not a guarantee of ADA compliance

## What You Need to Do

Issues are grouped into four action buckets so you know what to address first and who owns each fix.

**Fix Now** 2 issue categories (1 admin, 1 developer)

Verified failures that block access or comprehension for assistive technology users.

**Review Before Assigning** 1 issue category (1 developer)

Findings that need manual verification, involve design tradeoffs, or have mixed confidence.

**Raise With Plugin / Vendor Provider** 3 issue categories (3 vendor)

Issues that appear associated with third-party or platform-generated markup.

1 issue likely come from shared templates — verify in code before assuming a single fix resolves all pages.

# Section 1: 3 Site-Controlled Issues

These 3 issue categories were observed in site-controlled markup. Some may come from shared templates or components — verify in code before estimating scope. Each step includes the fix, what to avoid, and how to verify.

## Step 1 of 3: Images rendered without alternative text

You or Your Team in WordPress

Confirmed (High)

CRITICAL | 1 instance | Single page

Add alt text to each `<img>` element: `alt="description of what the image shows."` Use `alt=""` (empty) only for purely decorative images like background patterns. Do not use `aria-label` for images — use the `alt` attribute.

### Root cause

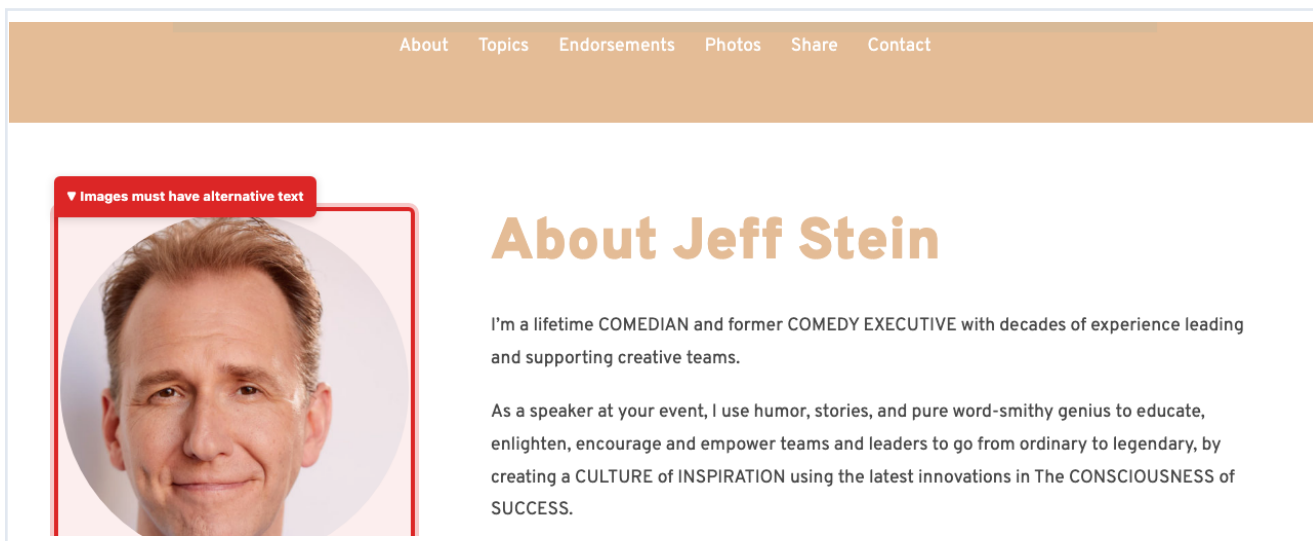
Origin: Content images added without alt attributes — may be template-level (logos, icons) or page-level (product/editorial images)

Template images (logo, hero) fix once for all pages. Content images require per-page attention.

WCAG: 1.1.1 | Level: AA | Priority: Must fix | Changes UI: No

Scope: Isolated to a single page.

### Where to find this on your site:



Example from: <https://jeffrystein.com/>

### Current code:

```

```

Element does not have an alt attribute

aria-label attribute does not exist or is empty

aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty

Element has no title attribute

Element's default semantics were not overridden with `role="none"` or `role="presentation"`

### Impact:

When the image conveys meaningful content that a screen reader user would miss.

### Exceptions:

- Purely decorative images (background patterns, spacers, visual separators) should have `alt=""` (empty), not descriptive text
- Images inside links where the link already has text do not need separate alt text
- CSS background images are not flagged by this rule

## Remediation:

In WordPress: Go to Media ! Library ! click the image ! fill in the "Alt Text" field on the right. For images already in pages: edit the page ! click the image ! fill in "Alt text" in the block settings.

Owner: Image alt text is editable in WordPress Admin without code changes. Each image has an alt text field.

## Fix pattern:

```
<!-- Add alt text that communicates the image's purpose in context -->
<!-- Descriptive images: alt="What the image shows or communicates" -->
<!-- Decorative images: alt="" (empty alt text) -->
<!-- Linked images: alt text should describe the link destination -->
```

### Avoid this bad fix:

Do not generate alt text from file names, URLs, or SKUs (e.g., "IMG\_2847.jpg" or "product-blue-widget"). If the image is decorative, use alt="" (empty). Do not write alt text for decorative images. Do not use aria-label on images — always use the alt attribute.

### How to verify:

DevTools: Inspect image ! check alt attribute value is meaningful. Screen reader: navigate to image and verify announcement. Automated: re-run axe-core on the page.

### Expected outcome:

Screen readers announce image descriptions. No visual change. Improves SEO image indexing as a side benefit.

## Step 2 of 3: Theme color tokens fail minimum contrast ratio

You or Your Developer

Confirmed (High)

SERIOUS | 29 instances | Likely shared template — observed on 2 pages

Update CSS color variables in your theme to meet WCAG 2.1 AA contrast ratio of 4.5:1 for normal text.

### Root cause

Origin: CSS color variables or inline styles in theme/template

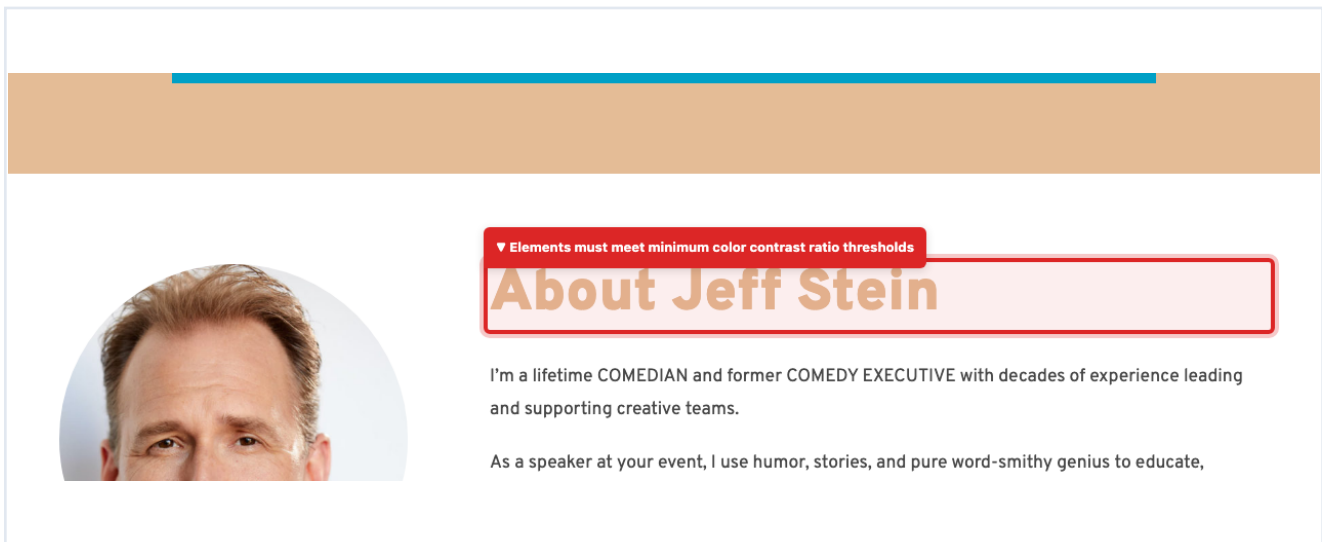
**This is not 29 separate issues — this is one underlying problem propagating across 29 instances.**

Typically 2-4 color pairings in the theme CSS cause all instances. Fix the variables, not each element.

WCAG: 1.4.3 | Level: AA | Priority: Must fix | Changes UI: Yes

Scope: Likely part of a shared template or component — appears on most scanned pages.

### Where to find this on your site:



▼ Elements must meet minimum color contrast ratio thresholds

## About Jeff Stein

I'm a lifetime COMEDIAN and former COMEDY EXECUTIVE with decades of experience leading and supporting creative teams.

As a speaker at your event, I use humor, stories, and pure word-smithy genius to educate,

Example from: <https://jeffrystein.com/>

### Current code:

```
<h2>About Jeff Stein</h2>
```

Element has insufficient color contrast of 1.75 (foreground color: #e4bc96, background color: #ffffff, font size: 42.0pt (56px), font weight: bold). Expected contrast ratio of 3:1

Affected pages: <https://jeffrystein.com/>, <https://jeffrystein.com/sample-page/>

### Impact:

When text is difficult to read for users with low vision, aging eyes, or in bright ambient light.

### Exceptions:

- Inactive UI elements, placeholder text, and purely decorative text are exempt
- Large text (18pt+ or 14pt+ bold) only requires 3:1 ratio
- Logos and brand names have no contrast requirement

### Remediation:

Locate: WordPress: Appearance ! Theme File Editor ! search style.css for the color values. Or check Customize ! Colors/Typography.  
Owner: Text color values are defined in your theme CSS. A developer must update the color variables.

### Fix pattern:

```
<!-- Current: #e4bc96 on #ffffff = 1.75:1 ratio (requires 3:1) -->  
<!-- Constraint: foreground color must meet 3:1 ratio against #ffffff -->  
<!-- Select a compliant value from your design system. Do not use arbitrary colors. -->
```

#### Avoid this bad fix:

Do not suggest arbitrary replacement colors without checking they work with the site's design system. State the required ratio and let the developer choose compliant colors.

#### How to verify:

DevTools: Use color contrast checker in Accessibility panel. Manual: verify text is readable at arm's length. Automated: re-run axe-core contrast check.

#### Expected outcome:

Text becomes readable for users with low vision and in bright environments. No layout changes. Colors change to meet minimum contrast ratio.

## Step 3 of 3: Link component lacks accessible naming

You or Your Developer

Verify Manually (Med)

SERIOUS | 11 instances | [Single page](#)

Verify first: Give each link a clear accessible name. Preferred sources (in order): (1) visible text inside the <a> tag, (2) alt text on an <img> inside the link, (3) aria-label as a last resort when no visible text exists.

Confidence mix in this bucket: 4 likely | 7 needs manual review

### Root cause

Origin: Links containing only icons, images with empty alt, or dynamically generated anchors without accessible text

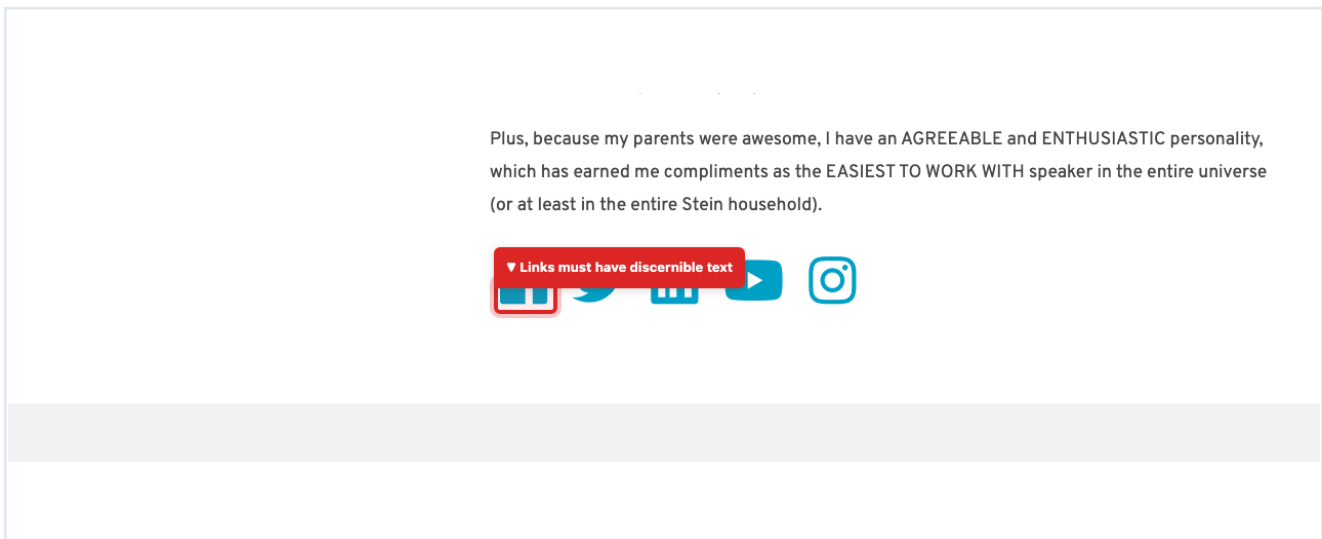
**This is not 11 separate issues — this is one underlying problem propagating across 11 instances.**

Icon links and social media links are usually template-level. Product card links may be component-level.

WCAG: 2.4.4 | Level: AA | Priority: Must fix | Changes UI: No

Scope: Isolated to a single page.

### Where to find this on your site:



Example from: <https://jeffrystein.com/>

### Current code:

```
<a href="https://www.facebook.com/JeffsWorldShow/"> <i class="fab fa-facebook fa-3x" aria-hidden="true"></i>  
</a>
```

Element is in tab order and does not have accessible text

Element does not have text that is visible to screen readers

aria-label attribute does not exist or is empty

aria-labelledby attribute does not exist, references elements that do not exist or references elements that are empty

Element has no title attribute

**Impact:**

When a screen reader user encounters the link and hears "link" with no indication of where it goes.

**Exceptions:**

- Links that are not in tab order (tabindex="-1") or are visually hidden duplicates of accessible links may be acceptable
- Check if the link has a computed accessible name from nested content not visible in the HTML snippet

**Remediation:**

Locate: Use your browser DevTools: right-click the unlabeled link !' Inspect. In WordPress: Appearance !' Theme File Editor !' search template files (.php) for matching HTML.

Owner: Link accessible names come from the HTML structure. A developer must update the template code.

*Verify this issue in the rendered DOM before applying any fix. Accepted remediation patterns are described in the developer guidance above.*

**Avoid this bad fix:**

Do not generate aria-labels from URL paths or slugs. Do not add aria-label when visible link text already provides the name. Prefer visible text over hidden labels.

**How to verify:**

DevTools: Inspect link !' Accessibility tab !' check "Name" field. Screen reader: Tab to link and verify announcement describes the destination.

**Expected outcome:**

Screen readers announce link destinations. No visual change unless visible text is added.

# Section 2: 3 Issues to Raise With App Vendors

---

The following issues appear associated with third-party or platform-generated markup based on DOM patterns (class names, data attributes, script signatures). Verify ownership in the rendered DOM and source before assigning remediation responsibility. If confirmed, contact each vendor listed below.

## 1. YouTube Embed Player

3 total instances | 3 issue types

### Issues found:

- Some buttons on your website have no accessible name (the label a screen reader announces).
- An element on your website has an accessibility label that isn't allowed on that type of element.
- An embedded video or widget on your website has no title or description.

### Copy and paste this email to their support team:

```
Subject: Accessibility (ADA/WCAG) Issues in Your YouTube Embed Player on Our Website

Hi YouTube Embed Player Support,

We recently conducted a WCAG 2.1 Level AA accessibility audit of our website (jeffrystein.com) and identified several accessibility violations originating from your app/widget on our site.

The specific issues are:
- Some buttons on your website have no accessible name (the label a screen reader announces) (1 instances across 1 pages)
- An element on your website has an accessibility label that isn't allowed on that type of element (1 instances across 1 pages)
- An embedded video or widget on your website has no title or description (1 instances across 1 pages)

These issues affect 1 pages on our site and create barriers for visitors who use screen readers and other assistive technology.

Can you let us know:
1. Are you aware of these accessibility issues?
2. Do you have a timeline for an accessibility update?
3. Are there any settings or configuration options we can adjust to improve accessibility?

We'd appreciate a response within the next two weeks. Thank you for your attention to this important issue.

Best regards,
[Your Name]
jeffrystein.com
```

# WCAG Success Criteria Failed

---

The following WCAG 2.1 success criteria were found to be violated. Each criterion is a specific, testable requirement from the Web Content Accessibility Guidelines.

## 1.1.1 — Non-text Content (Level A)

User Impact: High

All images, icons, and non-text content must have text alternatives (alt text) that describe their purpose.

Why it matters: Screen reader users rely entirely on alt text to understand images. Without it, they encounter blank gaps where content should be.

Remediation: Add descriptive alt attributes to all <img> elements. Use alt="" for purely decorative images. Ensure complex images have long descriptions.

## 1.4.3 — Contrast (Minimum) (Level AA)

User Impact: High

Text and images of text must have a contrast ratio of at least 4.5:1 (3:1 for large text).

Why it matters: Low contrast text is the #2 most common accessibility issue on the web. It affects users with low vision, aging eyes, and anyone in bright sunlight.

Remediation: Increase contrast between text and background colors. Use tools like WebAIM Contrast Checker. Large text (18pt+) needs only 3:1 ratio.

## 2.4.4 — Link Purpose (In Context) (Level A)

User Impact: High

The purpose of each link can be determined from the link text alone or from the link together with its context.

Why it matters: Screen reader users often navigate by links. "Click here" and "Read more" links without context are meaningless.

Remediation: Use descriptive link text that makes sense out of context. Avoid "click here", "read more", "learn more" without additional context via aria-label.

## 4.1.2 — Name, Role, Value (Level A)

User Impact: High

All UI components must have accessible names, roles, and states communicated to assistive technology.

Why it matters: Custom widgets (dropdowns, modals, tabs) built with divs and spans are invisible to screen readers without ARIA attributes.

Remediation: Use native HTML elements when possible. For custom widgets, add appropriate ARIA roles, labels, and state attributes.

# Appendix: Per-Instance Repair Checklist

Every instance found by the audit is listed below. Use this as a checklist to locate, fix, and verify each issue individually.

## button-name — 1 instance on 1 page [App / Plugin Vendor]

### #1 Button missing accessible name[App vendor]

Page: /

```
<button class="ytmVideoInfoLink ytmVideoInfoChannelAvatar"><img alt="thumbnail-image" class="ytCoreImageHost ytmVideoI...
```

No accessible name. Add visible text or aria-label to the button.



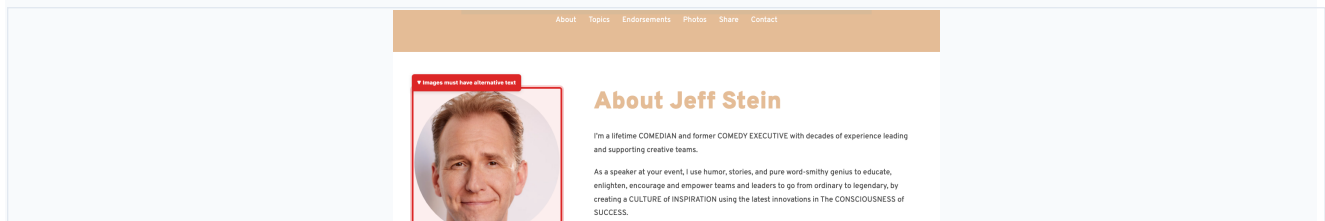
## image-alt — 1 instance on 1 page [You or Your Team in WordPress]

### #1 Missing image alt text[WordPress admin]

Page: /

Image type: Content

src: https://jeffrystein.com/wp-content/uploads/2020/01/Jeffrey\_Stein\_DHOFF\_115157-e1578667265700-512x512.jpg



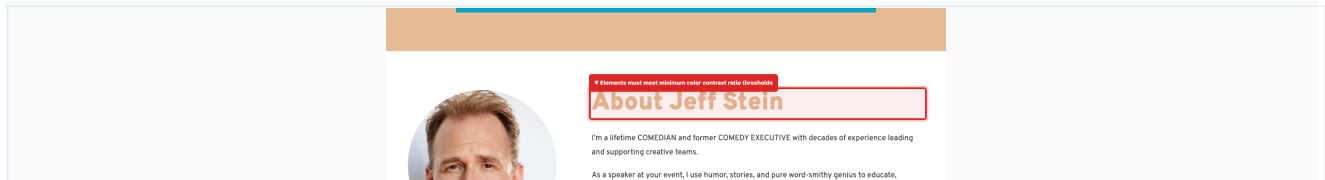
## color-contrast — 29 instances on 2 pages [You or Your Developer]

### #1 Insufficient color contrast[Developer]

Page: /

```
<h2>About Jeff Stein</h2>
```

#e4bc96 on #ffffff — Measured: 1.75:1; Required: 3:1

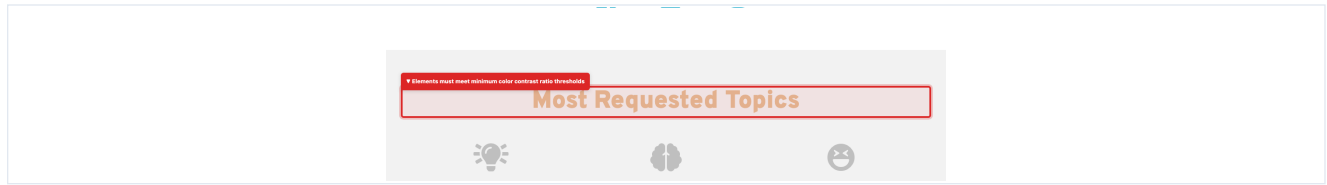


## #2 Insufficient color contrast(Developer)

Page: /

<h2>Most Requested Topics</h2>

#e4bc96 on #f2f2f2 — Measured: 1.57:1; Required: 3:1

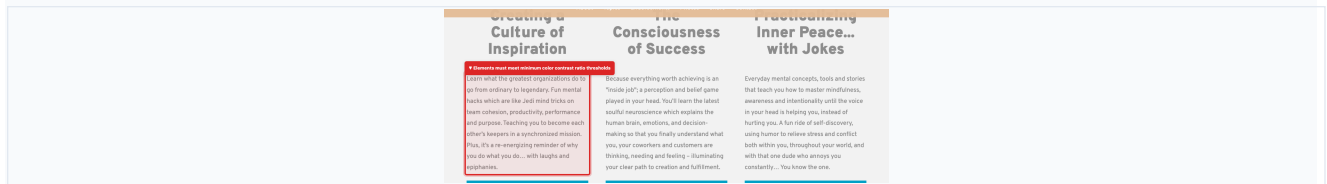


## #3 Insufficient color contrast(Developer)

Page: /

<p>

#8c8c8c on #f2f2f2 — Measured: 3:1; Required: 4.5:1

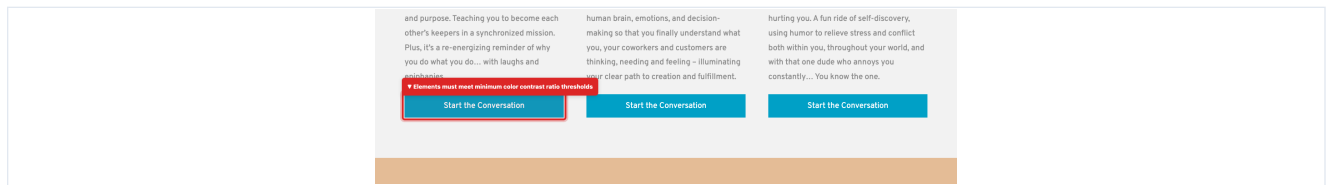


## #4 Insufficient color contrast(Developer)

Page: /

<a class="button" href="#nt\_contact\_widget-2"> Start the Conversation </a>

#ffffff on #00a0c6 — Measured: 3.06:1; Required: 4.5:1

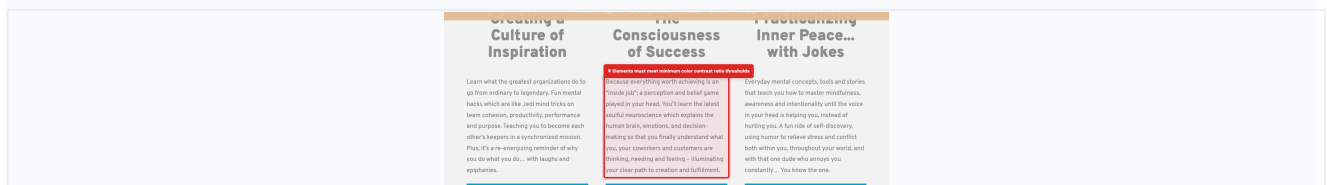


## #5 Insufficient color contrast(Developer)

Page: /

<p>

#8c8c8c on #f2f2f2 — Measured: 3:1; Required: 4.5:1



## #6 Insufficient color contrast(Developer)

Page: /

<a class="button" href="#nt\_contact\_widget-2"> Start the Conversation </a>

#ffffff on #00a0c6 — Measured: 3.06:1; Required: 4.5:1

## #7 Insufficient color contrast(Developer)

Page: /

<p>

#8c8c8c on #f2f2f2 — Measured: 3:1; Required: 4.5:1

### #8 Insufficient color contrast[Developer]

Page: /

```
<a class="button" href="#nt_contact_widget-2"> Start the Conversation </a>
#ffffff on #00a0c6 — Measured: 3.06:1; Required: 4.5:1
```

### #9 Insufficient color contrast[Developer]

Page: /

```
<h2>Endorsements</h2>
#ffffff on #e4bc96 — Measured: 1.75:1; Required: 3:1
```

### #10 Insufficient color contrast[Developer]

Page: /

```
<p>Jeff is awesome at working the crowd and blowing minds, which is why I keep hiring him for as many events as possib...
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #11 Insufficient color contrast[Developer]

Page: /

```
<p class="endorsement-byline">Cameron Graham Corporate Event Coordinator, MC, Speaker</p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #12 Insufficient color contrast[Developer]

Page: /

```
<p>Everything Jeff shares is wisdom and wit. Our audiences absolutely love him!</p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #13 Insufficient color contrast[Developer]

Page: /

```
<p class="endorsement-byline">Erika Feresten Community Leader, Executive Coach</p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #14 Insufficient color contrast[Developer]

Page: /

```
<p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #15 Insufficient color contrast[Developer]

Page: /

```
<p class="endorsement-byline">Mike Bonin, Los Angeles City Councilman, 11th District</p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #16 Insufficient color contrast[Developer]

Page: /

```
<p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #17 Insufficient color contrast[Developer]

Page: /

```
<p class="endorsement-byline">Renee Marshall Education Advocate, Renee Marshall Education</p>
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #18 Insufficient color contrast[Developer]

Page: /

```
<p>My staff tells me the day exceeded expectations. They unanimously agreed it was productive and also fun with flexib...
#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1
```

### #19 Insufficient color contrast[Developer]

Page: /

```
<p class="endorsement-byline">Ted Lieu, Member of Congress, CA 33rd District</p>
```

#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1

### #20 Insufficient color contrast[Developer]

Page: /

```
<p>Jeff is the most positive human being Ive ever met For the first time in my life, I really feel truly hopeful about...</p>
```

#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1

### #21 Insufficient color contrast[Developer]

Page: /

```
<p class="endorsement-byline">Julie Dove Actor, Director, Producer</p>
```

#878787 on #ffffff — Measured: 3.59:1; Required: 4.5:1

### #22 Insufficient color contrast[Developer]

Page: /

```
<h2>Spread the Word!</h2>
```

#e4bc96 on #ffffff — Measured: 1.75:1; Required: 3:1

### #23 Insufficient color contrast[Developer]

Page: /

```
<h2>Contact Me</h2>
```

#ffffff on #e4bc96 — Measured: 1.75:1; Required: 3:1

### #24 Insufficient color contrast[Developer]

Page: /

```
<p>Jeff@JeffryStein.com</p>
```

#ffffff on #e4bc96 — Measured: 1.75:1; Required: 4.5:1

### #25 Insufficient color contrast[Developer]

Page: /

```
<input type="submit" value="Send It!">
```

#ffffff on #e4bc96 — Measured: 1.75:1; Required: 4.5:1

### #26 Insufficient color contrast[Developer]

Page: /

```
<a href="https://notablethemes.com/themes/podium/" target="_blank" rel="noopener">Podium Theme by Notable Themes</a>
```

#00a0c6 on #ffffff — Measured: 3.06:1; Required: 4.5:1

### #27 Insufficient color contrast[Developer]

Page: /sample-page/

```
<h1 class="entry-title">Sample Page</h1>
```

#e4bc96 on #ffffff — Measured: 1.75:1; Required: 3:1

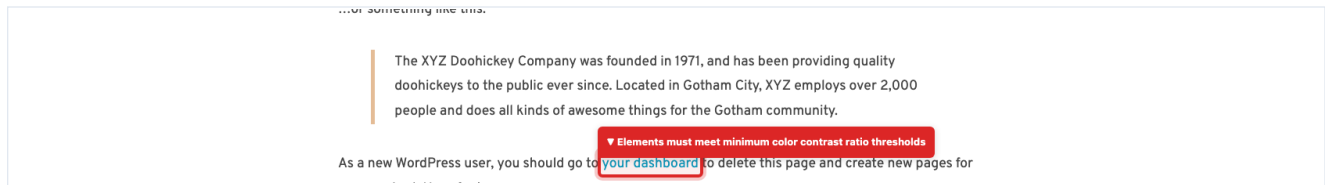


## #28 Insufficient color contrast[Developer]

Page: /sample-page/

```
<a href="http://jeffrystein.com/wp-admin/">your dashboard</a>
```

#00a0c6 on #ffffff — Measured: 3.06:1; Required: 4.5:1

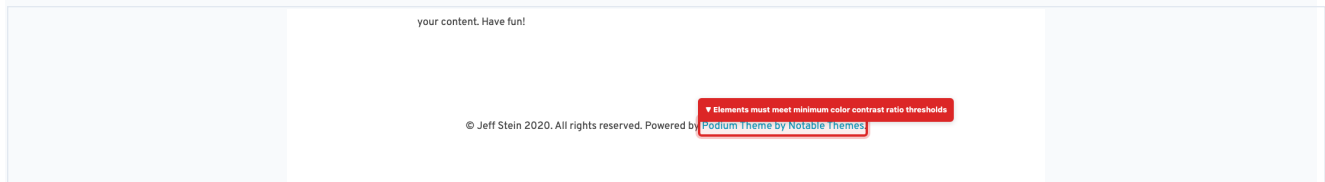


## #29 Insufficient color contrast[Developer]

Page: /sample-page/

```
<a href="https://notablethemes.com/themes/podium/" target="_blank" rel="noopener">Podium Theme by Notable Themes</a>
```

#00a0c6 on #ffffff — Measured: 3.06:1; Required: 4.5:1



## link-name — 11 instances on 1 page [You or Your Developer]

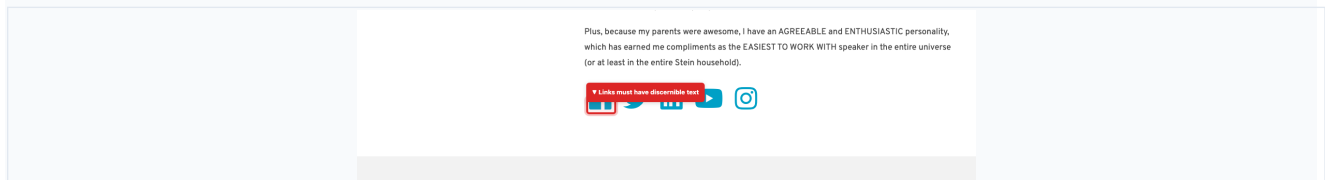
### #1 Link missing accessible name[Developer]

Page: /

Link target: <https://www.facebook.com/JeffsWorldShow/>

```
<a href="https://www.facebook.com/JeffsWorldShow/"> <i class="fab fa-facebook fa-3x" aria-hidden="true"></i> </a>
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.



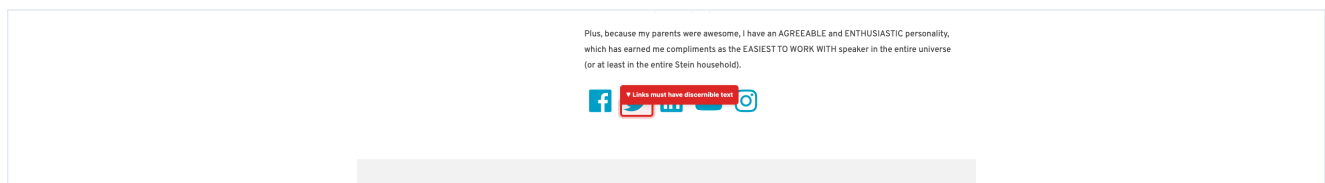
### #2 Link missing accessible name[Developer]

Page: /

Link target: <https://twitter.com/JeffsWorldShow>

```
<a href="https://twitter.com/JeffsWorldShow"> <i class="fab fa-twitter fa-3x" aria-hidden="true"></i> </a>
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.



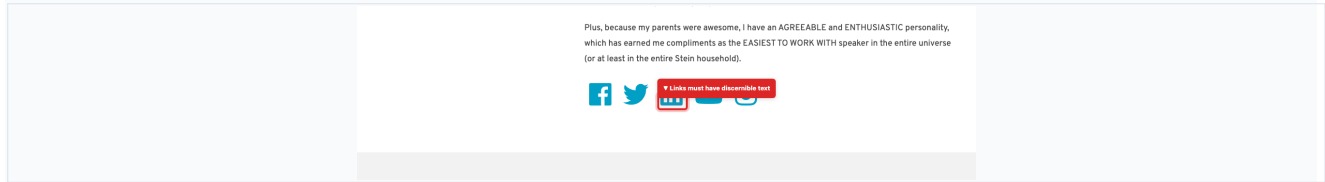
### #3 Link missing accessible name [Developer]

Page: /

Link target: <https://www.linkedin.com/in/jeff-stein-1136441/>

```
<a href="https://www.linkedin.com/in/jeff-stein-1136441/"> <i class="fab fa-linkedin fa-3x" aria-hidden="true"></i> </a>
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.



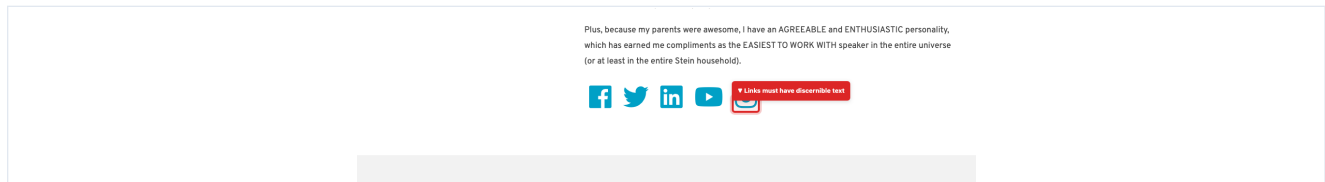
### #4 Link missing accessible name [Developer]

Page: /

Link target: <https://www.instagram.com/jeffworldshow/>

```
<a href="https://www.instagram.com/jeffworldshow/"> <i class="fab fa-instagram fa-3x" aria-hidden="true"></i> </a>
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.



### #5 Link missing accessible name [Developer]

Page: /

Link target: <https://jeffrystein.com/wp-content/uploads/2020/01/Jeff-2.png>

```
<a class="fancybox" href="https://jeffrystein.com/wp-content/uploads/2020/01/Jeff-2.png" rel="photo-gallery">
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

### #6 Link missing accessible name [Developer]

Page: /

Link target: [https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry\\_Stein\\_DHOFF\\_115200.jpg](https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry_Stein_DHOFF_115200.jpg)

```
<a class="fancybox" href="https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry_Stein_DHOFF_115200.jpg" rel="photo...">
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

### #7 Link missing accessible name [Developer]

Page: /

Link target: [https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry\\_Stein\\_DHOFF\\_115612.jpg](https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry_Stein_DHOFF_115612.jpg)

```
<a class="fancybox" href="https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry_Stein_DHOFF_115612.jpg" rel="photo...">
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

### #8 Link missing accessible name [Developer]

Page: /

Link target: [https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry\\_Stein\\_DHOFF\\_115190.jpg](https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry_Stein_DHOFF_115190.jpg)

```
<a class="fancybox" href="https://jeffrystein.com/wp-content/uploads/2020/01/Jeffry_Stein_DHOFF_115190.jpg" rel="photo...">
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

### #9 Link missing accessible name [Developer]

Page: /

Link target: <https://www.facebook.com/sharer/sharer.php?u=https://jeffrystein.com>

```
<a href="https://www.facebook.com/sharer/sharer.php?u=https://jeffrystein.com" target="_blank"> <i class="fab fa-faceb...">
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

### #10 Link missing accessible name [Developer]

Page: /

Link target: <https://twitter.com/intent/tweet?url=https://jeffrystein.com&text=Jeff Stein: Create a Culture of Inspiration... with humor.&>

```
<a href="https://twitter.com/intent/tweet?url=https://jeffrystein.com&text=Jeff Stein: Create a Culture of Inspira...">
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

## #11 Link missing accessible name [Developer]

Page: /

Link target: <http://www.linkedin.com/shareArticle?mini=true&url=https://jeffrystein.com>

```
<a href="http://www.linkedin.com/shareArticle?mini=true&url=https://jeffrystein.com" target="_blank"> <i class="fa...
```

No accessible name. Add visible text, alt text on the linked image, or aria-label.

## aria-prohibited-attr — 1 instance on 1 page [App / Plugin Vendor]

### #1 Invalid ARIA attribute [Elementor (Page Builder)]

Page: /

```
<div class="html5-video-player ytp-hide-controls ytp-exp-bottom-control-flexbox ytp-modern-caption ytp-livebadge-color...
```

Element has a prohibited ARIA attribute.



## frame-title — 1 instance on 1 page [App / Plugin Vendor]

### #1 Iframe missing title [App vendor]

Page: /

```
<iframe src="https://www.youtube.com/embed/tgCDDRIG7g0" frameborder="0" allow="accelerometer; autoplay; encrypted-medi...
```

Embedded frame has no title attribute.



# Page-by-Page Results

---

**<https://jeffrystein.com/>**

Violations: 41 (2C / 39S / 0M / 0m) | Passed: 47 | Scan type: quick

**<https://jeffrystein.com/sample-page/>**

Violations: 3 (0C / 3S / 0M / 0m) | Passed: 30 | Scan type: quick

# Recommendations & Next Steps

---

## 1. Fix Theme and Template Issues First

1 issue(s) appear to originate from shared templates or components. Fixing these at the source resolves the issue across multiple pages at once, making this the highest-impact starting point.

## 2. Update Color Contrast in Design System

Low contrast affects the largest population of any accessibility barrier, including users with low vision, older adults, and anyone in bright sunlight. Update your theme color variables to meet the 4.5:1 ratio for normal text and 3:1 for large text.

## 3. Audit Images Before Writing Alt Text

Before adding alt text, categorize each image: product images need descriptive text, decorative images need alt="", linked images need text describing the destination, and logos need the brand name. Do not auto-generate alt text from file names.

## 4. Escalate Third-Party Issues to Vendors

3 issue category(s) appear to originate from third-party apps or platform code. Use the email templates in Section 2 to contact each vendor. Track responses and consider alternatives if vendors do not commit to fixes.

## 5. Manually Verify Flagged Findings

1 finding(s) are marked "Verify Manually" because automated scanning alone cannot confirm them. Have a developer inspect these in the rendered page using DevTools accessibility panel, keyboard navigation, and a screen reader (VoiceOver, NVDA, or JAWS).

## 6. Run Screen Reader and Keyboard Checks on Critical Pages

Automated scans catch 30-40% of accessibility issues. For your highest-traffic pages (homepage, product pages, checkout, contact), test with a screen reader and keyboard. Focus on: can a user complete the task without seeing the screen?

## 7. Add Accessibility Checks to Your Development Process

Prevent new issues by adding axe-core or eslint-plugin-jsx-a11y to your build pipeline. Include accessibility acceptance criteria in feature requirements. Re-scan after fixes to verify resolution.

## Verification Notice

While these issues are detected using industry-standard automated tools, we recommend:

- Verifying key findings in a browser using the DevTools Accessibility panel
- Testing critical user flows using only a keyboard (Tab, Enter, Escape)
- Spot-checking with a screen reader (VoiceOver on Mac, NVDA on Windows, or TalkBack on Android)
- Re-scanning after fixes are applied to confirm resolution

Some accessibility requirements cannot be fully evaluated through automation and require manual testing with assistive technologies. Findings marked "Verify Manually" should be inspected in the rendered page before assigning remediation work.

## Methodology, Limits, and Legal Disclosures

Automated testing detects approximately 30-40% of WCAG 2.1 accessibility barriers. Many requirements, including keyboard navigation, screen reader compatibility, visual adjustments, and core user flow testing, can only be verified through manual review with assistive technologies. This report identifies issues detectable by automated scanning and should be used as a prioritized remediation worklist, not as a complete assessment of accessibility compliance.

This report is a technical remediation document provided for informational and remediation-planning purposes only. It does not constitute legal advice or a legal opinion regarding ADA or WCAG compliance.

Where a finding appears associated with platform, vendor, or injected markup, ownership attribution is provisional and based on DOM patterns (class names, data attributes, script signatures). Ownership should be verified in the rendered DOM and page source before assigning remediation responsibility.