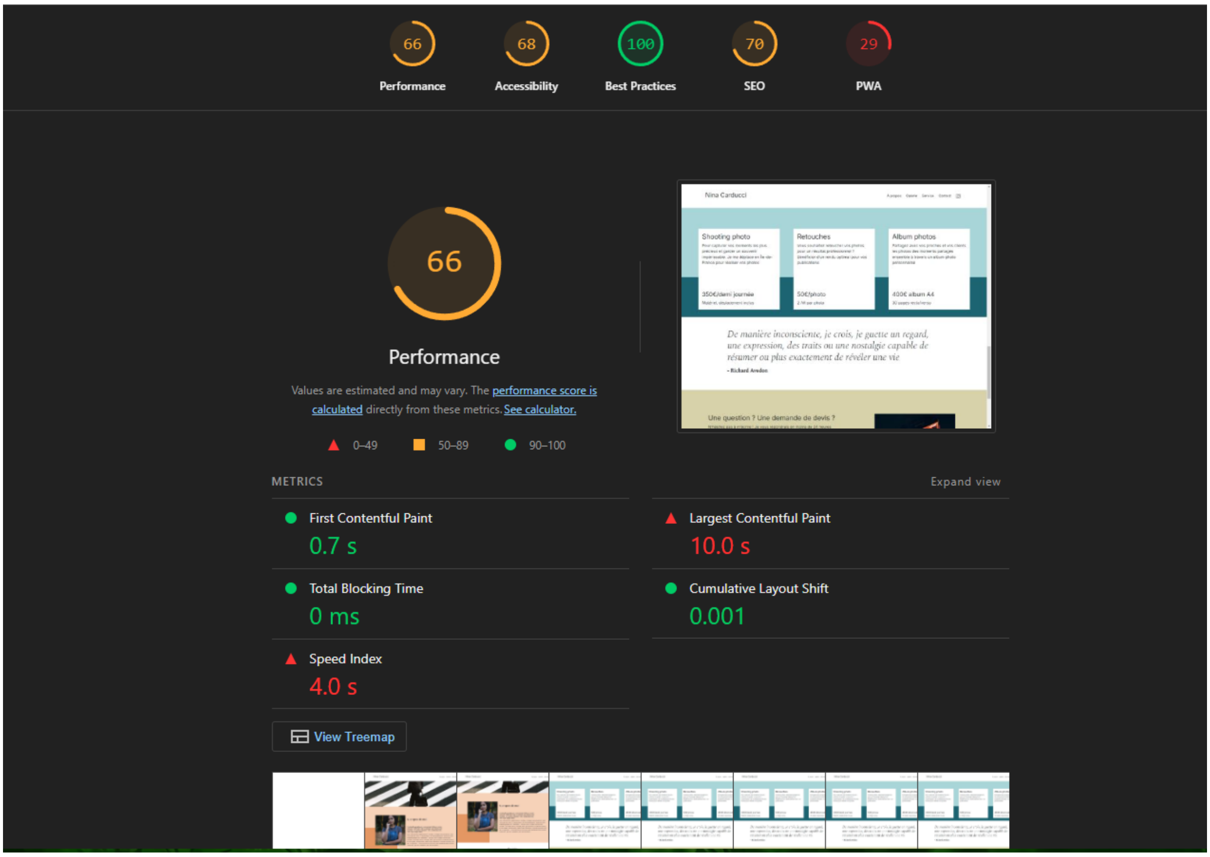
Intervention report

Nina Carducci

I - Lighthouse Score

Lighthouse score before optimisation



66 Performance

68 Accessibility

100 Best Practices

70 SEO

29 PWA

66

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● First Contentful Paint
0.7 s

▲ Largest Contentful Paint
10.0 s

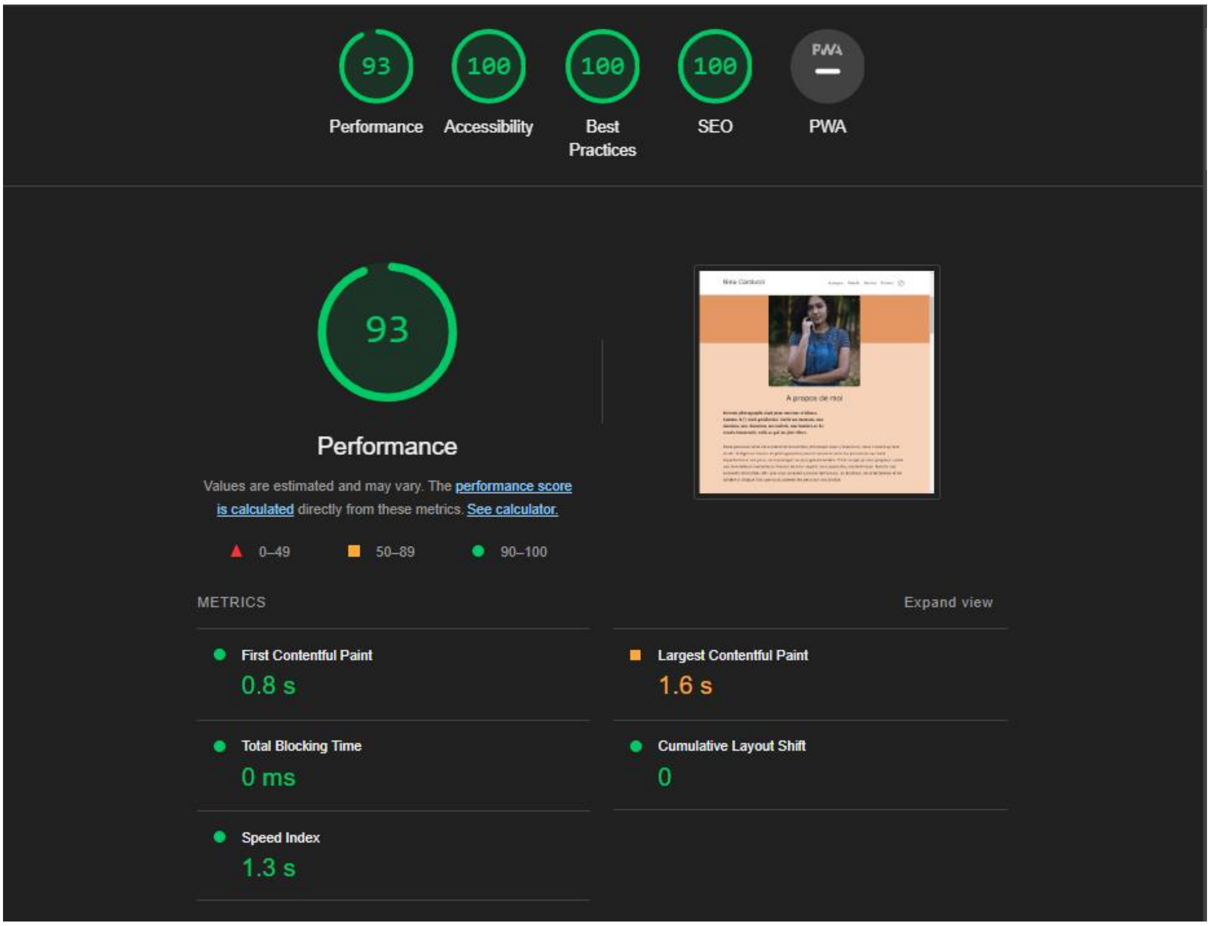
● Total Blocking Time
0 ms

● Cumulative Layout Shift
0.001

▲ Speed Index
4.0 s

View Treemap

Score Lighthouse after optimisation



93 Performance

100 Accessibility

100 Best Practices

100 SEO

PWA

93

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● First Contentful Paint
0.8 s

■ Largest Contentful Paint
1.6 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

● Speed Index
1.3 s

II - Details of optimisations and interventions carried out

1 – Performance (Images)

Performance (66)

Identified issues :

- Largest Contentful Paint (LCP): 10.0s (should be <2.5s)

- Speed Index: 5.5s

- Total Blocking Time : 0 ms (good)

- Cumulative Layout Shift : 0.001 (good)

- Image elements lack explicit width and height

Possible Fixes :

Serve images in next-gen formats (e.g., WebP) - **Convert JPG/PNG images to WebP**

- **Compress and resize images**

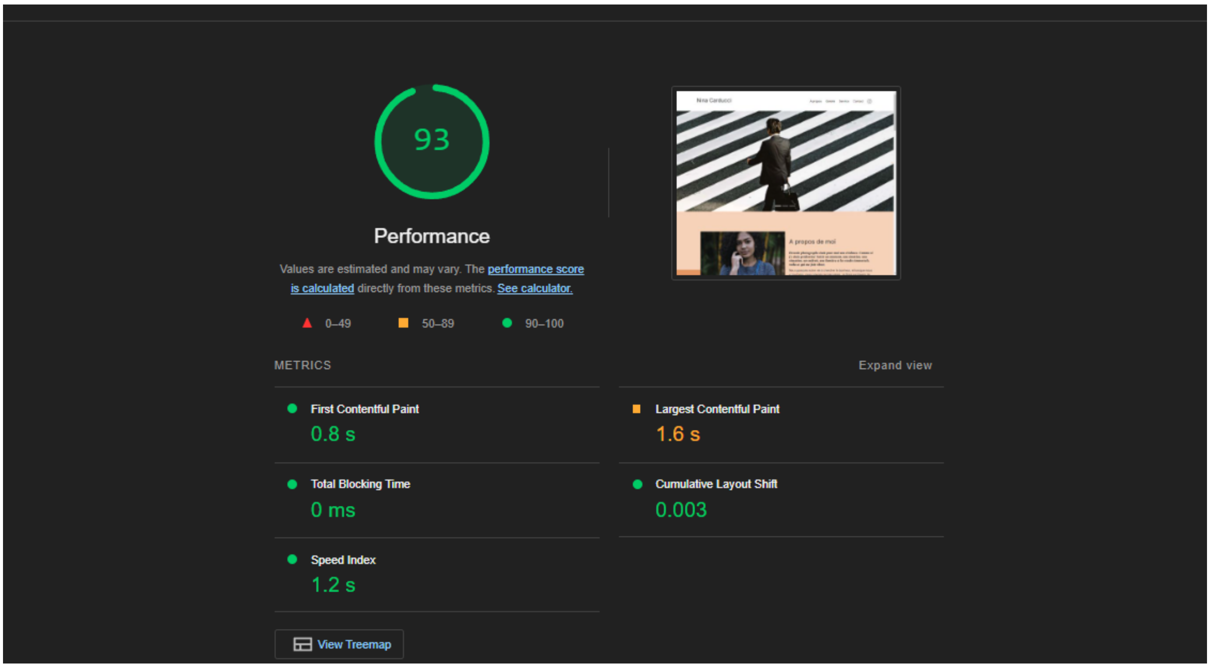
- Eliminate render-blocking resources

- Minify/Remove unused JS and CSS

- Text compression and image encoding improvements

- Reduce unused scripts/styles

- Add width and height attributes to images



93

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

Expand view

● First Contentful Paint
0.8 s

■ Largest Contentful Paint
1.6 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0.003

● Speed Index
1.2 s

Summary

The Lighthouse audit highlighted a **high Largest Contentful Paint (LCP)** time of over **4 seconds**, severely impacting the Performance score (initially 65–73).

- Assumed cause: Oversized images in the **gallery modal**.

- Compressed and replaced those images, but the LCP score didn't significantly improve.

- Re-analyzed the Lighthouse audit and discovered the true LCP culprit was **a 2.9MB image in the carousel** at the top of the page — a hero image rendered above the fold.

- Also noted that .png images were flagged as heavy and not using modern formats.

Solution Applied

- **Compressed** the carousel images and converted them to. webp (modern image format).

- **Replaced** heavy .png files (e.g., profile and camera images) with optimized. webp equivalents.

- **Resized heavy loaded** gallery thumbnail images and drastically reduced the image size.

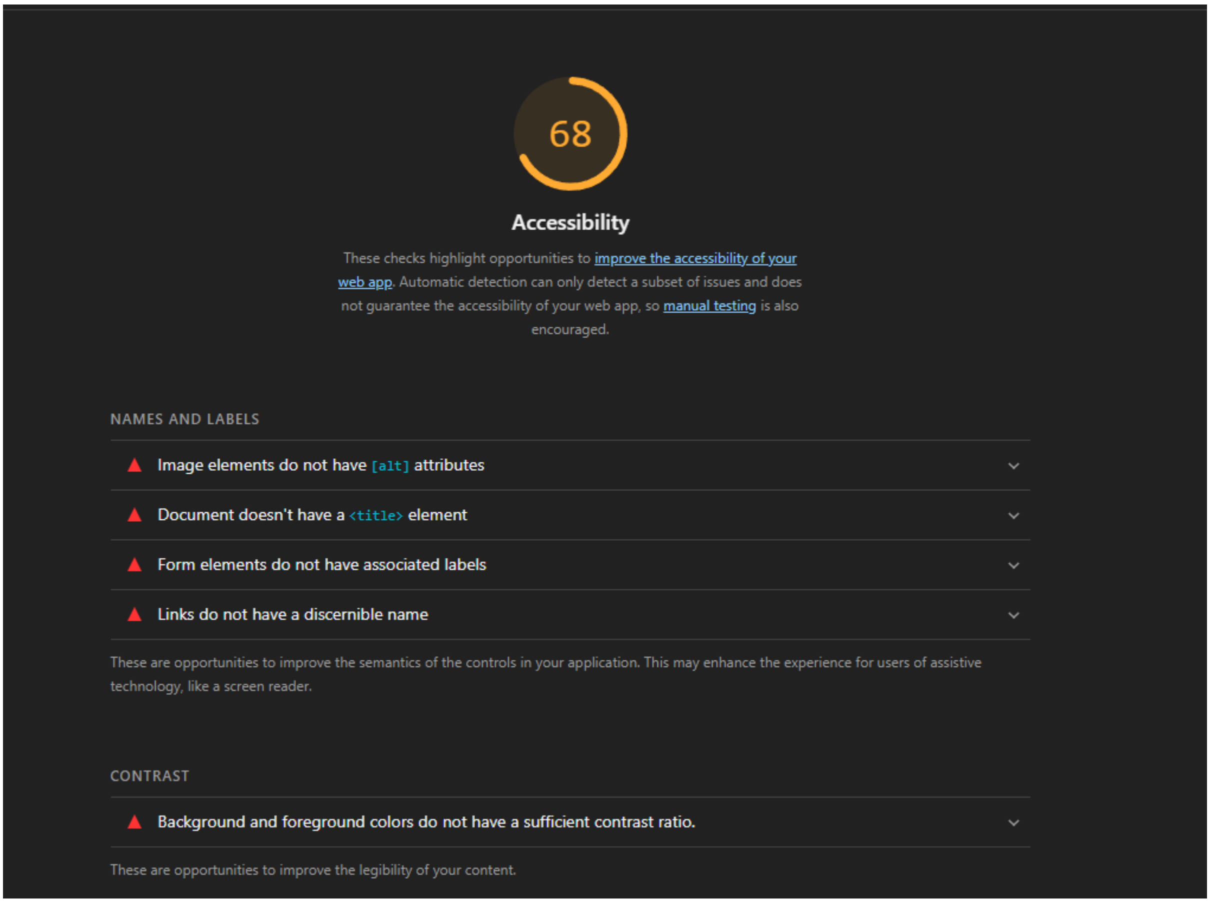
- No code or layout modifications were required — only asset-level changes.

Outcome

- **Performance** score improved to **93.**
- **LCP** reduced to **1.6s.**
- **All other metrics (FCP, Speed Index, TBT, CLS) entered the green zone.**
- The problem identification had some challenges despite the fix being minimal - yet impactful, demonstrating the power of proper image optimization for performance.

2 – Accessibility

Accessibility (68)



Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

NAMES AND LABELS

- ▲ Image elements do not have `[alt]` attributes
- ▲ Document doesn't have a `<title>` element
- ▲ Form elements do not have associated labels
- ▲ Links do not have a discernible name

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

- ▲ Background and foreground colors do not have a sufficient contrast ratio.

These are opportunities to improve the legibility of your content.

Identified issues :

- Form elements lack labels

- Links have no discernible name

- Low contrast between text and background –

Fixes :

- Form Accessibility:

Added <label> elements associated with all form inputs (for="..." and id="...") to ensure screen readers can properly announce field names.

- Heading Structure:

Changed the main page title from <h3> to <h1> to follow semantic order. No layout change was required as styling was already handled by CSS.

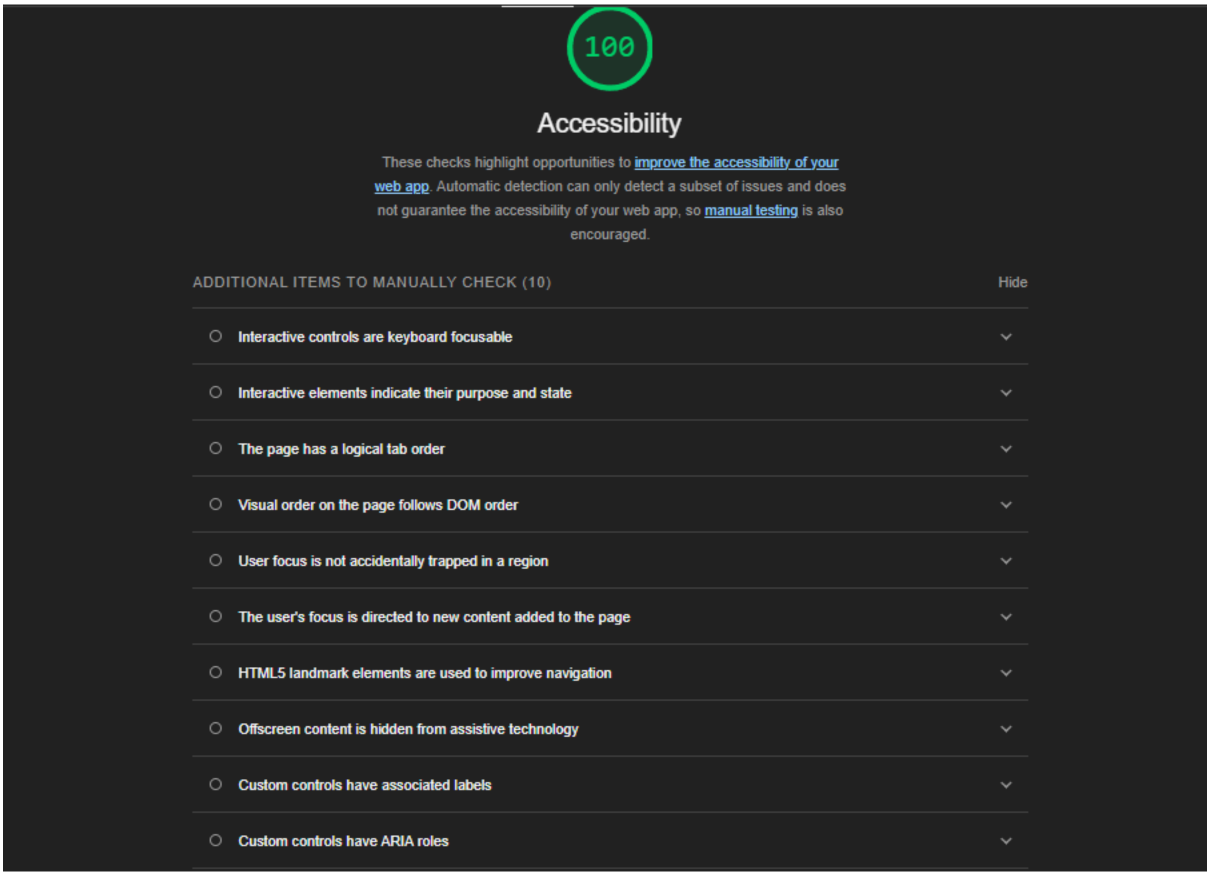
- Landmark Roles:

Wrapped primary page content in a <main> element and the menu in a <nav> element, improving keyboard navigation and screen reader structure.

- Contrast Improvements:

Adjusted background and text colors of buttons and UI elements (especially in the gallery filter tags) to ensure sufficient color contrast ratios, resolving the final WAVE contrast alerts.

Result: Lighthouse Accessibility score hit 100.



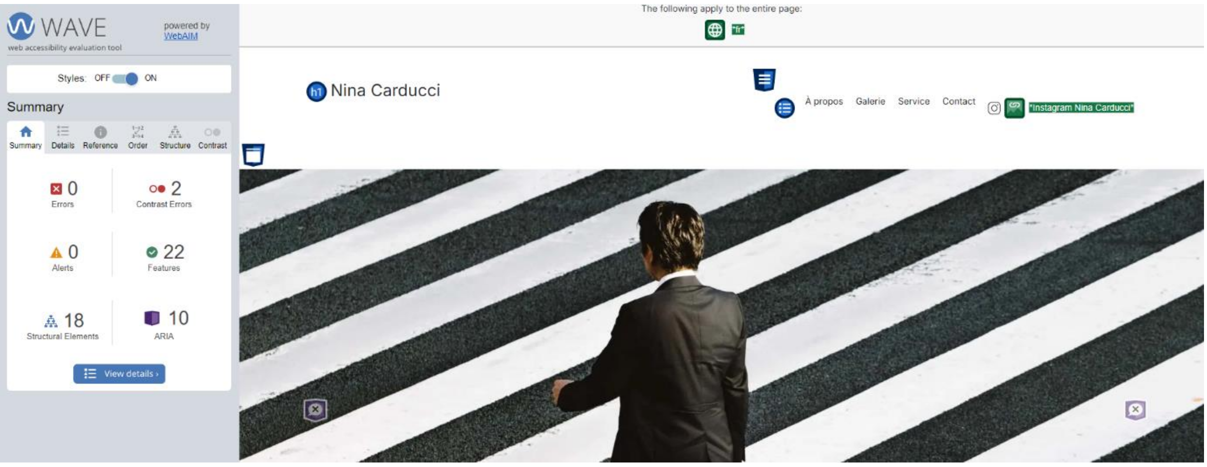
Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Hide

- ○ Interactive controls are keyboard focusable
- ○ Interactive elements indicate their purpose and state
- ○ The page has a logical tab order
- ○ Visual order on the page follows DOM order
- ○ User focus is not accidentally trapped in a region
- ○ The user's focus is directed to new content added to the page
- ○ HTML5 landmark elements are used to improve navigation
- ○ Offscreen content is hidden from assistive technology
- ○ Custom controls have associated labels
- ○ Custom controls have ARIA roles

Wave screenshot after Accessibility updates



0 Errors

2 Contrast Errors

0 Alerts

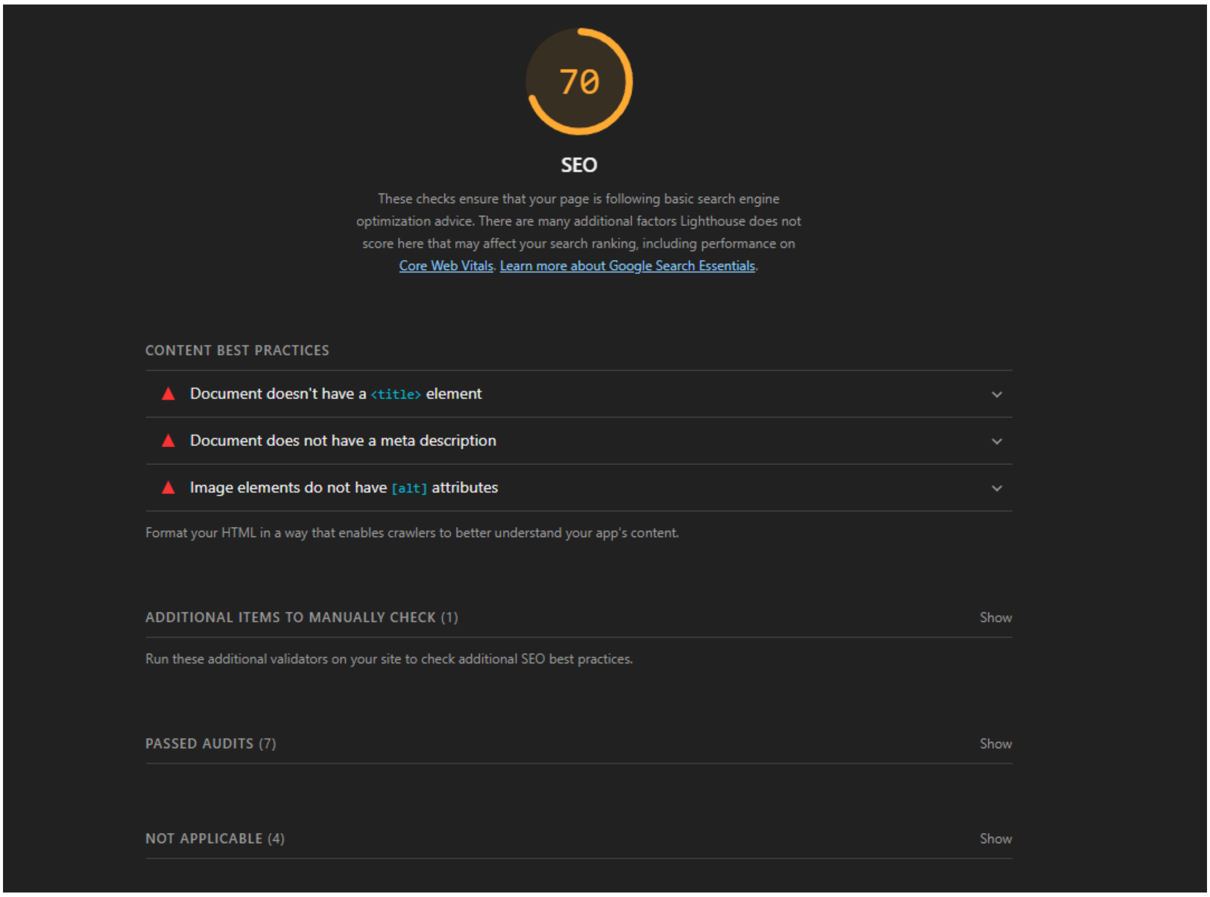
22 Features

18 Structural Elements

10 ARIA

3 – SEO

SEO (70)



70

SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.

CONTENT BEST PRACTICES

▲ Document doesn't have a `<title>` element

▲ Document does not have a meta description

▲ Image elements do not have `[alt]` attributes

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Show

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (7) Show

NOT APPLICABLE (4) Show

Identified issues :

- No `<title>` element

- No meta description

- Images missing `alt` attributes - Add `alt`

Fixes :

- Add missing `<title>` and `<meta name="description">`

- attributes to all `<img>` tags

- Declared the page language with <html lang="fr">

- Improved semantic structure

Improvements :

- Enhancement on search visibility: By optimizing the website for **local search,** we **improve chances** of appearing in relevant **local search results.**

- Implemented Open Graph tags for Facebook, LinkedIn ang general sharing.

- Twitter Card Metadata

Enables rich link previews when the site is shared on Twitter.

- Structured Data (Local SEO)

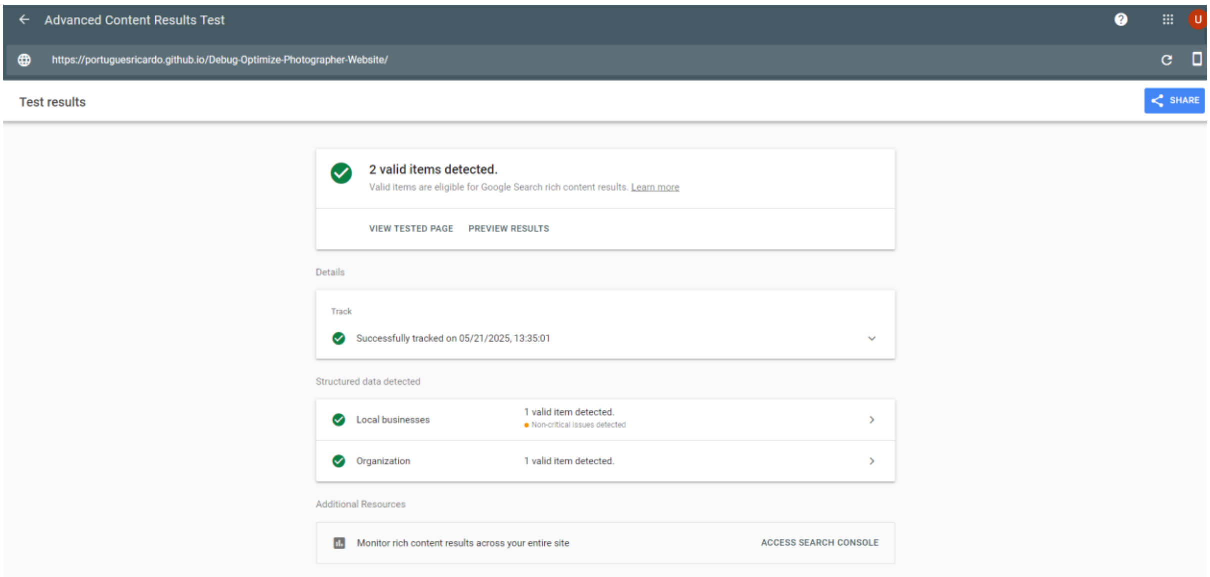
- Added Schema.org microdata to the <head> of the HTML to describe the business:

  @type: LocalBusiness

  Name, description, address, phone, email, and website included

  This allows search engines to understand the nature and location of the business.

Result: Google Rich Result Test



https://portuguesricardo.github.io/Debug-Optimize-Photographer-Website/

Test results

2 valid items detected.

Valid items are eligible for Google Search rich content results. Learn more

Details

Track

Successfully tracked on 05/21/2025, 13:35:01

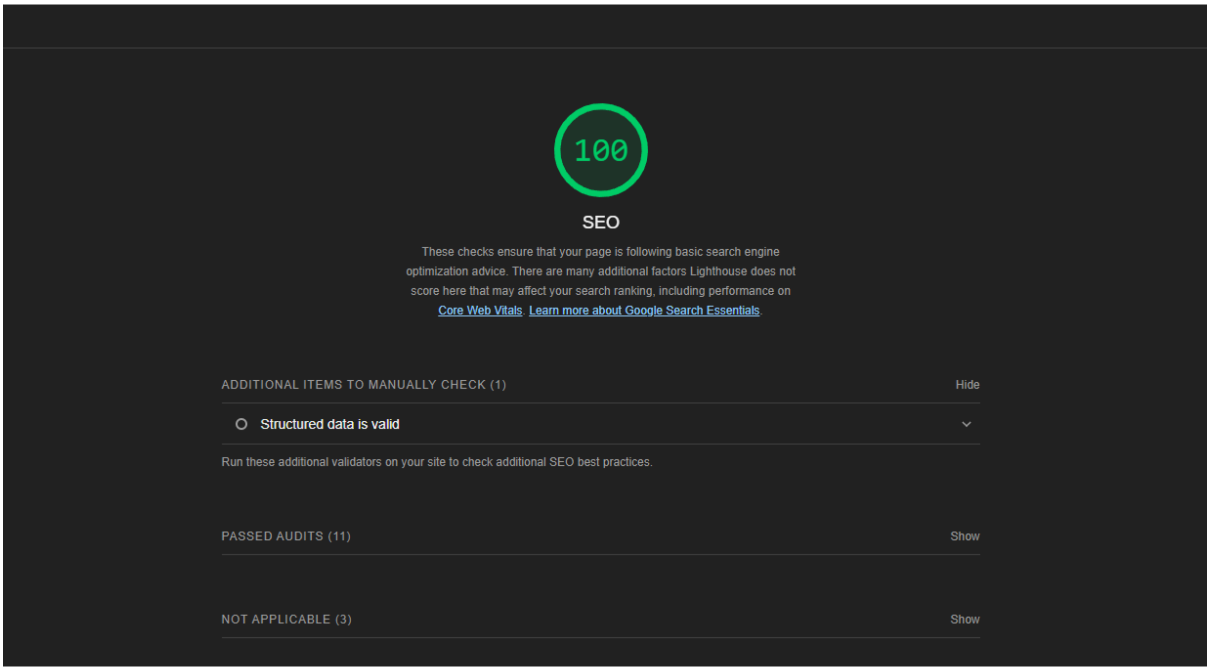
Structured data detected

Local businesses	1 valid item detected. ● Non-critical issues detected
Organization	1 valid item detected.

Additional Resources

Monitor rich content results across your entire site

Result: Lighthouse SEO score hit 100.



SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Hide

○ Structured data is valid

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (11) Show

NOT APPLICABLE (3) Show

IV - Details of additional work carried out at the customer's request

1. **“The gallery’s previous and next image navigation doesn’t work.”**

Outcome

- Both modal navigation arrows now work correctly.

- All logic is properly scoped using the correct `lightboxId`.

2. **“When you change filters to display images, the selected category is not visible. The category should normally have a gold background. The same as the default filter.”**

Outcome

- Clicking a filter now highlights it in brown (previous gold).

-.active-tag is both applied and styled correctly.

-JavaScript and CSS are now in sync.

V - Acceptance log

1 – Debugging Summary: Gallery Modal Navigation Bug

Context

The modal image gallery on the photographer's website opened correctly, but the navigation arrows (next/previous) did not update the image.

Goal

Fix the non-working left (previous) and right (next) arrow buttons in the modal gallery.

Tools Used

- Chrome DevTools

- Sources tab (Debugger)

- Console tab

- Event Listener Breakpoint

Step-by-Step Debugging Process

1. Set Breakpoints for Mouse Clicks

- In DevTools > Sources > Event Listener Breakpoints, I enabled `Mouse > click`.

- Clicked the navigation arrows while the modal was open.

- Observed the debugger pausing inside unrelated Bootstrap carousel logic (`bootstrap.bundle.js`) — a red herring.

2. Identified the Correct Script: `maugallery.js`

- Located functions: `prevImage()` and `nextImage()`.

- Set breakpoints inside these to trace logic.

3. Console Logging for Diagnostics

- Added `console.log()` statements to check:

- If arrow clicks triggered the functions.

- What image was considered "active".

- What the modal `src` was, and whether it matched any gallery items.

4. Discovered Root Cause

- Image comparison logic failed:

JS

```
 if ($(this).attr("src") === $(".lightboxImage").attr("src")
```

- This failed because it compared relative vs. absolute paths.

5. Fixed Matching Logic

- Replaced with:

JS

```
 if (this.src === document.querySelector(`#${lightboxId} .lightboxImage`).src)
```

6. Final Fixes

- Also corrected a bug in `prevImage()`:

- The function used a `prev` variable but mistakenly tried to set the image using `next`.

- Fixed by updating:

JS

```
$(`#${lightboxId} .lightboxImage`).attr("src", $(prev).attr("src"));
```

Outcome

- **Both modal navigation arrows now work correctly.**

- **All logic is properly scoped using the correct `lightboxId`**

2- Debugging Summary: Filter Highlight Not Showing Context

The filter buttons (e.g., All, Weddings, Concerts) correctly filtered images when clicked. However, the selected filter button was not showing the expected **gold highlight** to indicate it was active.

Goal

Fix the missing **visual feedback** so users can clearly see which filter is currently selected.

Tools Used

- **Chrome DevTools:**

Elements tab – to inspect filter elements and confirm class changes
Styles panel – to verify applied CSS

- **Code review:**

style.css
maugallery.js

Debugging Process

1. Inspected Filter Behavior

- Clicked filter buttons and inspected them in DevTools
- Verified that the class active-tag **was added correctly** via JavaScript

2. Checked for Matching CSS

- Looked for styles targeting .active-tag — **none found**
- Found existing styles for:

Css

```
.nav-pills .nav-link .active {
  background-color: #BEB45A;
}
```

But **JavaScript wasn't adding .active**, it added **.active-tag.**

3. Identified the Mismatch

- JavaScript applies **.active-tag**
- CSS only styles **.active**
- Result: no visual highlight, even though logic was working

Solution Implemented

CSS Fix

Added this to style.css:

css

CopyEdit

```
.nav-pills .nav-link.active-tag {
  background-color: #BEB45A;
  color: white;
  border-radius: 5px;
}
```

This kept the JavaScript unchanged and made the UI behave correctly.

Outcome

- Clicking a filter now highlights it in gold
- **.active-tag** is both applied and styled correctly
- JavaScript and CSS are now in sync

This was bug resolved by tracing through DevTools and aligning JavaScript with CSS. Classic case of **functionality without styling** due to mismatch.

Appendix

Full Lighthouse audit report

Final audit performed on:

https://portuguesricardo.github.io/Debug-Optimize-Photographer-Website/

Scores achieved:

- **Performance: 95**
- **Accessibility: 100**
- **Best Practices: 100**
- **SEO: 100**

📎 Full report available at:

https://portuguesricardo.github.io/Debug-Optimize-Photographer-Report/