

Rapport d'intervention

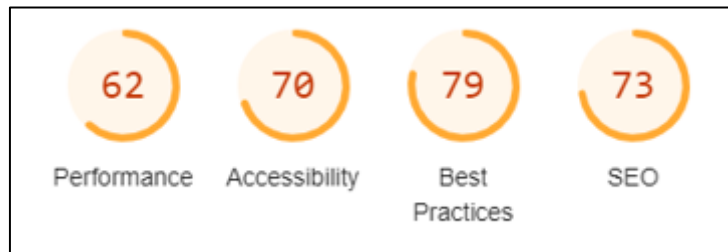
Nina Carducci

Sommaire:

I - Comparatif avant et après optimisation	2
II - Détails des optimisations effectuées	3
III - Accessibilité du site	4-5
IV - Détails de réalisations additionnelles à la demande du client	6
V - Cahier de recette	6
Annexe	7-14
Rapport complet de l'audit Lighthouse.	

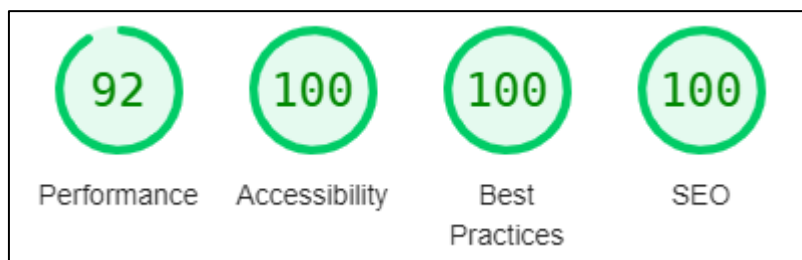
I- Comparaison du score lighthouse avant et après les optimisations

Le score **lighthouse** du site de nina avant les optimisations:



Comme on peut le voir les scores sont loin de 100, qui est le score le plus élevé et qui indique que le site ne présente aucun dysfonctionnement.

Le score **lighthouse** du site de nina après les optimisations:



Comme vous pouvez le voir on est au-dessus de 90 pour la totalité des champs que ce soit la performance, accessibility, best practises et SEO.

II- Détails des optimisations effectuées

1. J'ai modifié le HTML, les éléments de la balise <head>et la sémantique.

- J'ai ajouté la balise <title> et la balise méta<description> qui sont des éléments que chaque page doit contenir.
- Création d'un <footer> qui est essentiel dans une page.
- Ajout de l'attribut lang=fr dans la balise html.
- J'ai lié les labels aux inputs dans le formulaire en donnant la même valeur à l'attribut "for" et à l'attribut "id".
- Ajout de la balise <Header>, <Footer>, <Main>, <Nav>
- réorganisation des balises h1, h2, h3 ... pour une sémantique juste.

2. J'ai apporté des modifications aux images sur le site:

- j'ai modifié le format des fichiers image qui était de format png et jpg pour les mettre en webp plus rapide en chargement.
- J'ai compressé les fichiers pour qu'il soit moins lourds, moins de données.
- J'ai ajouté des balises alt là où elles étaient absentes dans le code html.
- J'ai précisé les balises width et height dans le code html afin de supprimer l'error "Cumulative Layout Shift".
- Ajout de l'attribut "loading: lazy" sur certaines images qui permet d'améliorer les performances du site.
- Ajout dans le code css de l'attribut "max-width: 100%" qui permet aux images de ne pas déborder de leur contenant.

3. J'ai apporté une modification au code CSS:

J'ai résolu le problème de contraste qu'il y avait entre la couleur du background du site et le background doré qui apparaît lorsque l'on clique sur l'un des éléments du filtre. Le ratio n'était pas bon, j'ai donc fait le choix de mettre le ratio le plus élevé 7:1. Pour éliminer cette alert.

4. Mise en place des balises méta pour les réseaux sociaux:

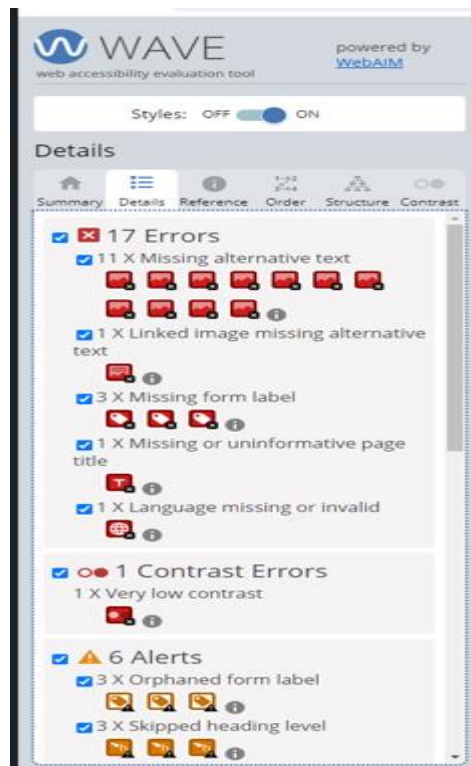
Ajout des balises méta pour les **réseaux sociaux**: Facebook et X anciennement twitter. Voir dans le code HTML, L 4-19

5. Mise en place du référencement local avec le **rich snippet**:

Voir dans le code HTML, L 23-53

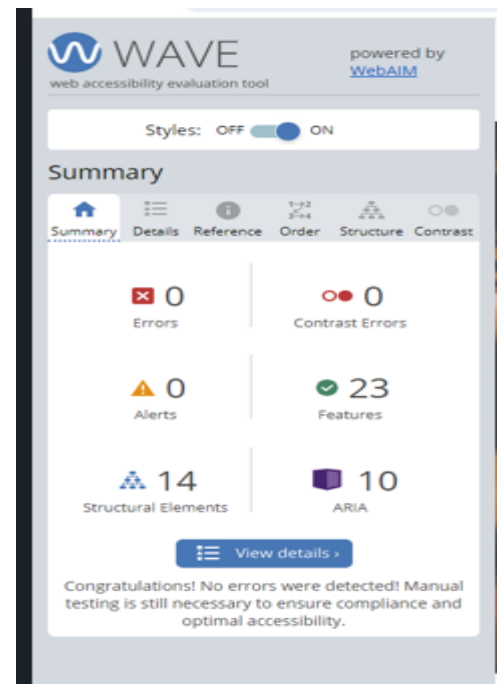
III- Accessibilité du site

Voici le rapport **Wave** avant les optimisations:




Et voici le rapport Wave après les optimisations:

Comme vous pouvez le voir le test wave indique qu'il n'y a plus d'erreurs détectées. Toutes les erreurs ou alertes ont été levées.




Résultats du test

**2 éléments valides détectés**
Les éléments valides peuvent apparaître dans les résultats enrichis de la recherche Google. [En savoir plus](#)


[AFFICHER LA PAGE TESTÉE](#) [PRÉVISUALISER LES RÉSULTATS](#)

Données structurées détectées

 Commerces et services à proximité

1 élément valide détecté
● Problèmes non critiques détectés


>

 Organisation

1 élément valide détecté
● Problèmes non critiques détectés

>

Ressources supplémentaires

 Surveiller les résultats enrichis pour l'intégralité de votre site

[ACCÉDER À LA SEARCH CONSOLE](#)

Voici une capture d'écran du **rich snippet** ou “extrait enrichi” en français qui correspond à un petit descriptif présent dans les résultats google en dessous du lien qui pointe vers notre site web. Cela permet d'indiquer à l'internaute les grandes lignes de notre site et de faciliter sa

compréhension. Le résultat du test indique bien que notre référencement a été fait correctement. Il est bien indiqué commerce et service de proximité et notre organisation est spécifié aux internautes. Les informations que j'ai mis sur le rich snippet dans mon code html sont visibles par les internautes.

IV- Détails de réalisations additionnelles à la demande du client.

Demande 1:

À la demande du client, on avait pour mission de débbuguer le site, lorsque la souris passait devant les éléments du filtre, il devait y avoir un rectangle doré qui apparait pour entourer le mot. De plus lorsque l'on sélectionnait une image, on devait pouvoir faire défiler les images par simple clique sur le symbole suivant ou précédent.

J'ai résolu le premier bug en ajoutant à la ligne 274 la class active à l'élément sélectionné car il était supprimé précédemment mais pas ajouté de nouveau pour le nouvel élément.

```
$(this).addClass("active active-tag");
```

Demande 2:

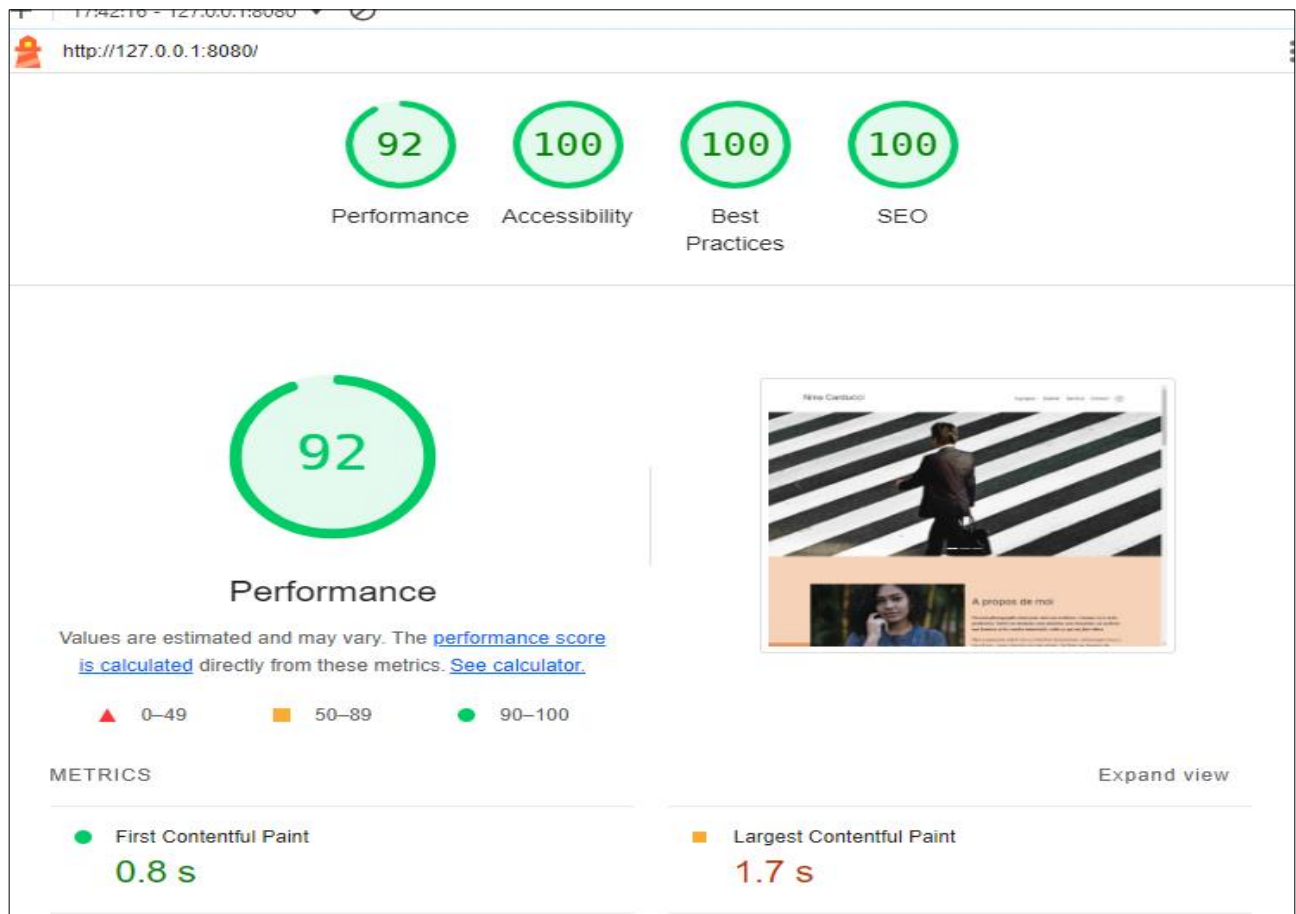
J'ai résolu le second bug en modifiant les fonctions nextImage et prevImage, à l'intérieur de ces fonctions la variable next n'était pas correctement indexée. Pour la variable next de la fonction nextImage, il fallait `next = imageCollection[index + 1] || imageCollection[index.length - 1]` et pour la fonction prevImage: `next = imageCollection[index - 1] || imageCollection[0]`.

V - Cahier de recette

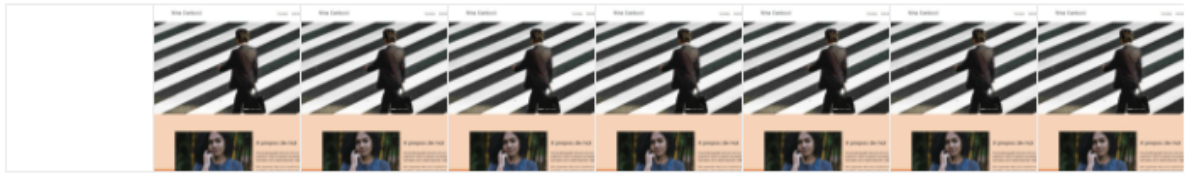
Détail des fonctionnalités débuggées et de leur statut :

ID	Action	Résultat initial	Résultat attendu	Statut	Remarques et commentaires
1	Débugger la fonctionnalité.	Lorsque la souris passe et sélectionne un des éléments du filtre, le filtrage a lieu mais aucune autre action se produit.	Lorsque la souris passe et sélectionne l'un des éléments du filtre, un rectangle doré doit entourer cet élément	Résolu	J'ai ajouté la class "active" à l'élément sélectionné.
2	Débugger la fonctionnalité.	Lorsque on sélectionne une image et que l'on souhaite voir l'image suivante ou la précédente ça ne fonctionne pas.	Lorsque on clique sur l'icon suivante ou précédente, l'image suivante ou précédente doit être affiché.	Résolu	J'ai modifié les fonctions nextImage et prevImage.

Annexe:



 [View Treemap](#)



Show audits relevant to: [All](#) [FCP](#) [LCP](#) [TBT](#) [CLS](#)

DIAGNOSTICS

- ▲ Properly size images — Potential savings of 2,611 KiB
- ▲ Enable text compression — Potential savings of 373 KiB
- ▲ Largest Contentful Paint element — 1,720 ms
- ▲ Minify JavaScript — Potential savings of 79 KiB
- ▲ Reduce unused CSS — Potential savings of 203 KiB
- ▲ Minify CSS — Potential savings of 39 KiB
- ▲ Eliminate render-blocking resources — Potential savings of 580 ms

METRICS

[Expand view](#)

● First Contentful Paint
0.8 s

■ Largest Contentful Paint
1.7 s

 [View Treemap](#)



Show audits relevant to: [All](#) [FCP](#) [LCP](#) [TBT](#) [CLS](#)

DIAGNOSTICS

- ▲ Properly size images — Potential savings of 2,611 KiB
- ▲ Enable text compression — Potential savings of 373 KiB
- ▲ Largest Contentful Paint element — 1,720 ms
- ▲ Minify JavaScript — Potential savings of 79 KiB
- ▲ Reduce unused CSS — Potential savings of 203 KiB
- ▲ Minify CSS — Potential savings of 39 KiB
- ▲ Eliminate render-blocking resources — Potential savings of 580 ms

■ **Serve static assets with an efficient cache policy** — 18 resources found

■ **Avoid enormous network payloads** — Total size was 3,216 KiB

○ **Avoid large layout shifts** — 1 layout shift found

○ **Initial server response time was short** — Root document took 10 ms

○ **Avoids an excessive DOM size** — 140 elements

○ **Avoid chaining critical requests** — 10 chains found

○ **JavaScript execution time** — 0.1 s

○ **Minimizes main-thread work** — 0.5 s

○ **Minimize third-party usage** — Third-party code blocked the main thread for 130 ms

○ **Avoid long main-thread tasks** — 1 long task found

More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.



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

- ☐ 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

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

PASSED AUDITS (23)

Show

NOT APPLICABLE (34)

Show





Best Practices

TRUST AND SAFETY

- ☐ Ensure CSP is effective against XSS attacks



GENERAL

- ☐ Detected JavaScript libraries



PASSED AUDITS (14)

Show

NOT APPLICABLE (2)

Show



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



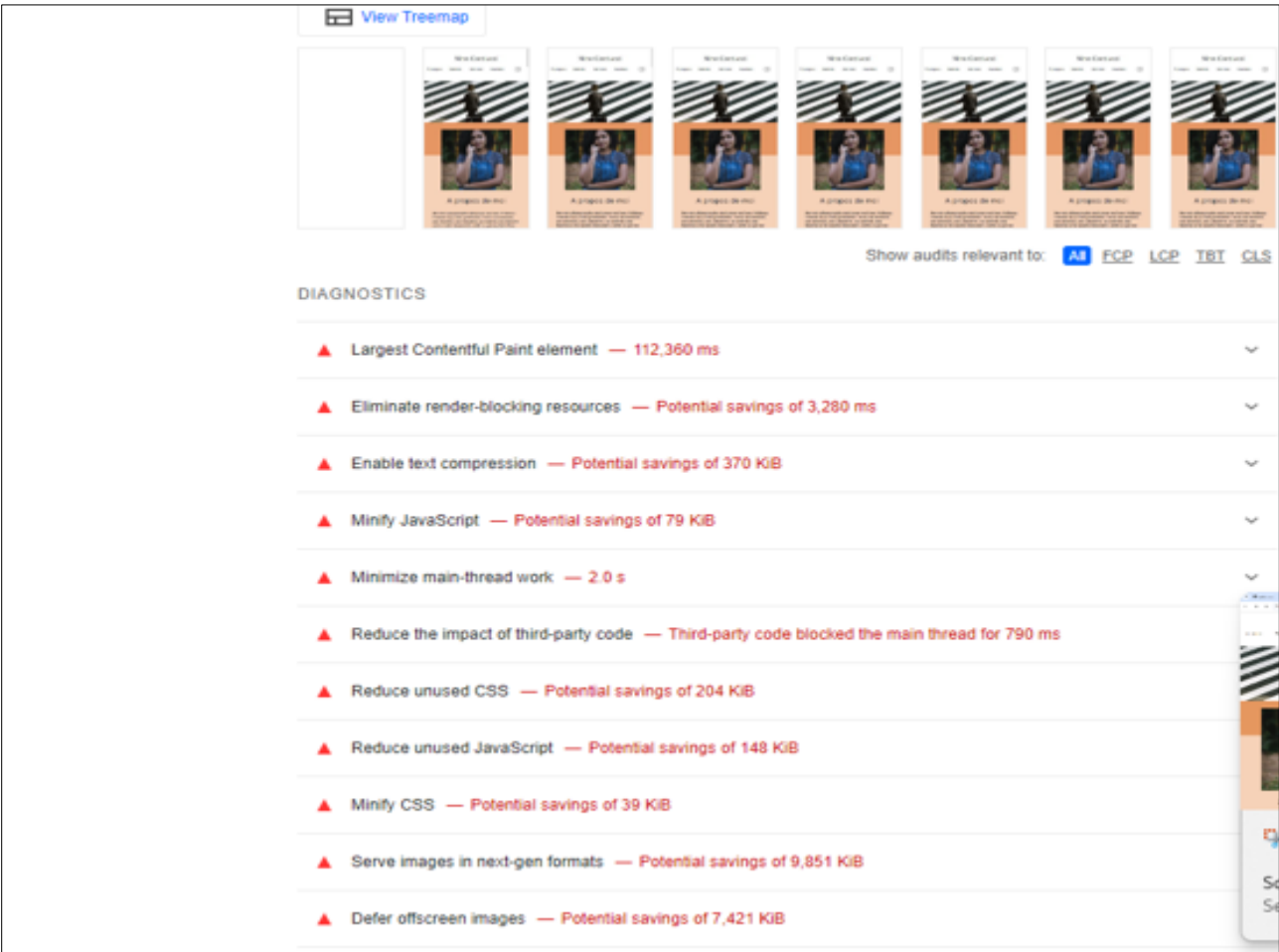
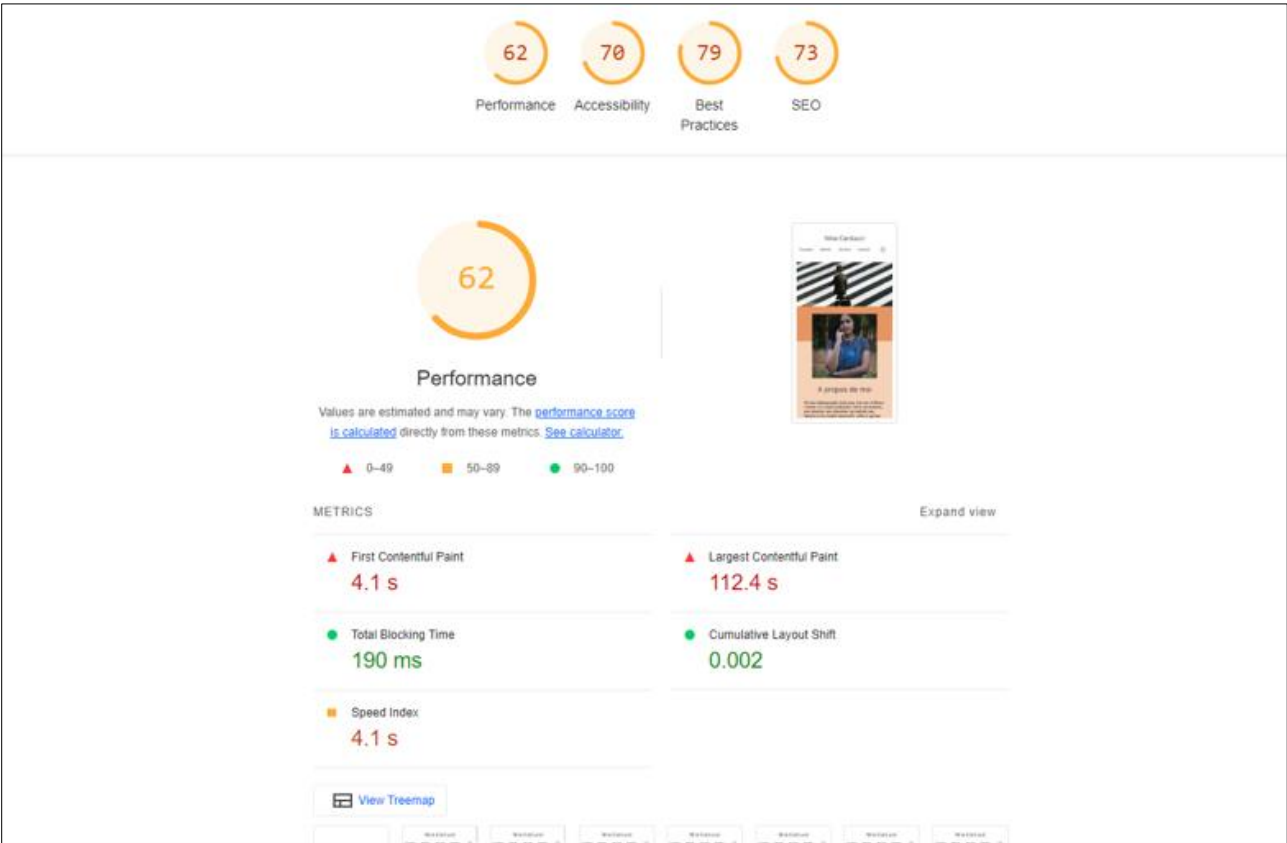
Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (8)

Show

NOT APPLICABLE (2)

Show



<div> <div>62</div> <div>78</div> <div>79</div> <div>73</div> </div>		
▲	Defer offscreen images — Potential savings of 7,421 KiB	▼
▲	Use HTTP/2 — 21 requests not served via HTTP/2	▼
▲	Properly size images — Potential savings of 21,920 KiB	▼
▲	Efficiently encode images — Potential savings of 1,955 KiB	▼
■	Image elements do not have explicit <code>width</code> and <code>height</code>	▼
■	Serve static assets with an efficient cache policy — 20 resources found	▼
■	Avoid enormous network payloads — Total size was 30,701 KiB	▼
○	JavaScript execution time — 0.9 s	▼
○	Avoid long main-thread tasks — 3 long tasks found	▼
○	Avoid large layout shifts — 2 layout shifts found	▼
○	Initial server response time was short — Root document took 0 ms	▼
○	Avoids an excessive DOM size — 132 elements	▼
○	Avoid chaining critical requests — 10 chains found	▼
More information about the performance of your application. These numbers don't directly affect the Performance score.		
PASSED AUDITS (15)		Show

