

# Rapport d'intervention

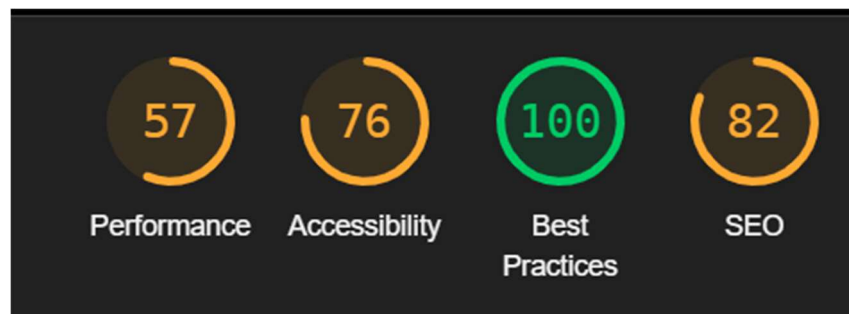
Nina Carducci

<b>I - Comparatif avant et après optimisation</b>	<b>2</b>
<b>II - Détails des optimisations effectuées</b>	<b>2</b>
1 - Les images	2
2 - ...	2
<b>III - Accessibilité du site</b>	<b>2</b>
<b>IV - Détails de réalisations additionnelles à la demande du client</b>	<b>2</b>
1 - ...	2
<b>Annexe</b>	<b>3</b>
Rapport complet de l'audit Lighthouse	3

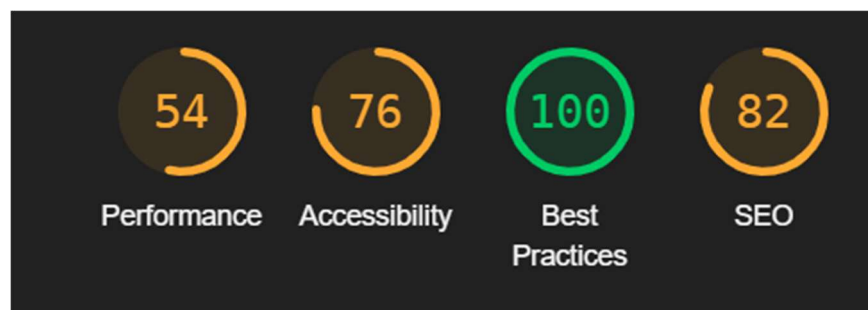
# I - Score Lighthouse

## Score Lighthouse avant optimisation

### Desktop

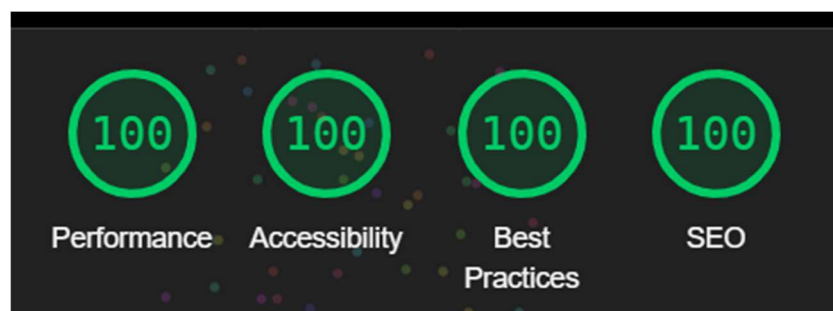


Mobile :

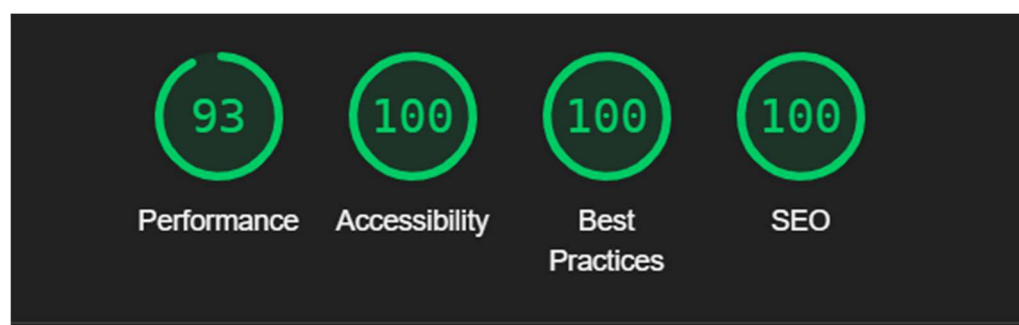


## Score Lighthouse après optimisation

### Desktop



Mobile :



## II - Détails des optimisations et interventions effectuées

### 1 - Les images

Le projet comporte originellement 15 images pour un poids total de 31,3 MB. Nous avons effectué les modifications suivantes aux images :

- Conversion des images au format WebP pour une meilleure compression, des temps de chargement accélérés et une réduction des données.
- Simplification des noms de fichiers pour plus de concision.
- Ajout des attributs ALT manquants pour renforcer l'accessibilité.
- Inclusion de l'attribut TITLE pour aider les personnes ayant des difficultés visuelles à mieux comprendre le contenu.
- Définition explicite des dimensions (width et height) dans le code HTML pour éviter les décalages de mise en page ("Cumulative Layout Shift").
- Ajout d'une règle CSS fixant la largeur max-width= 100 % avec height : auto,
- Mise en œuvre de l'attribut "loading: lazy" pour un chargement progressif des images au fur et à mesure du défilement.
- Utilisation des attributs "srcset" et "sizes" pour les images du carrousel, améliorant ainsi leur réactivité et l'affichage en fonction des écrans..

Après les modifications, le poids total des images est de 3 MB, soit un gain de 90,4 %.

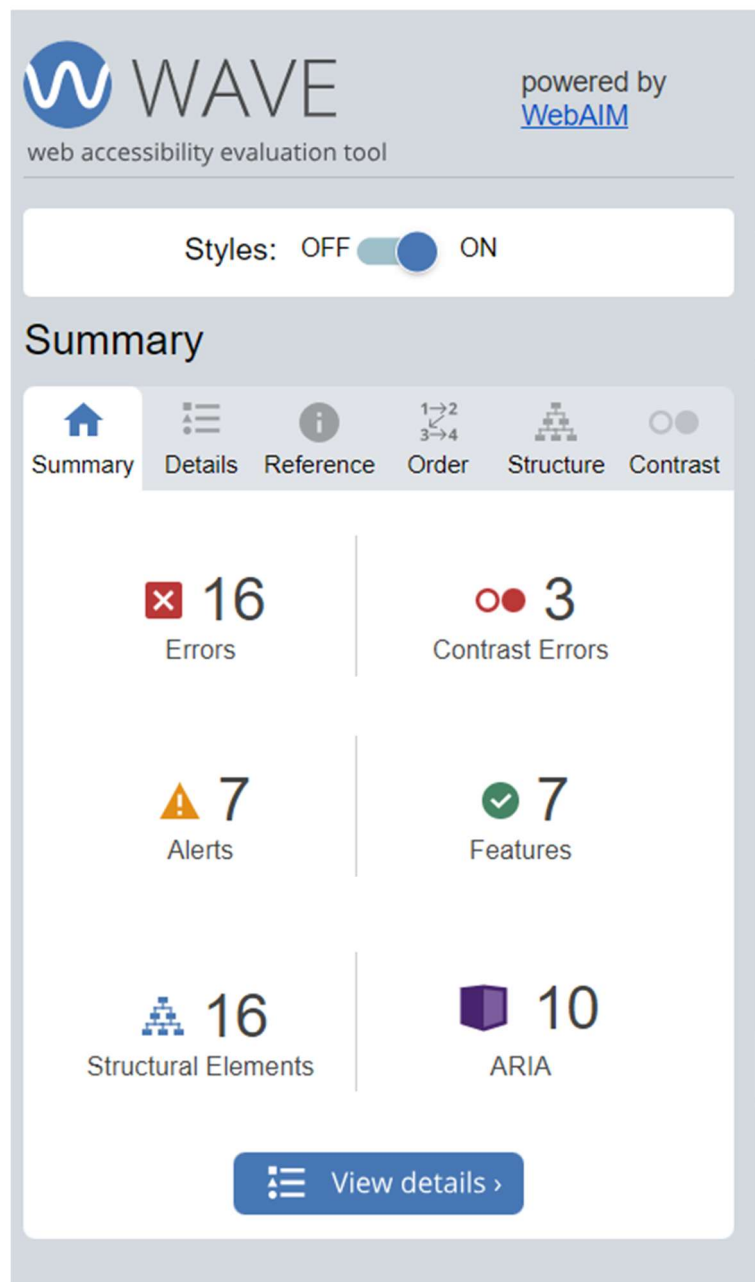
### 2 - HTML et aux fichiers associés :

- Ajout de l'attribut lang="fr" dans la balise HTML pour indiquer la langue principale du site.
- Déplacement de l'importation des Google Fonts dans le fichier CSS pour une meilleure organisation.
- Ajout des Meta Tags :
  - Intégration des balises Meta Open Graph pour les réseaux sociaux comme Facebook.
  - Ajout des Meta Tags spécifiques à Twitter (désormais X) pour optimiser le partage sur cette plateforme.
  - Utilisation de l'outil *opengraph.xyz* pour générer automatiquement les Meta Tags nécessaires.
- Intégration des données structurées au format JSON-LD, avec l'aide de *technicalseo.com*, pour améliorer le référencement local.
- Déplacement des balises script de l'en-tête vers le bas de la page, juste avant la balise </body>, afin d'accélérer le chargement initial du site.
- Ajout des favicons dans l'en-tête pour un affichage correct dans les navigateurs.
- Suppression du dossier local de Bootstrap.
- Remplacement de Bootstrap local par les liens CDN, en incluant la dernière version des fichiers CSS et JS.

### III - Accessibilité du site

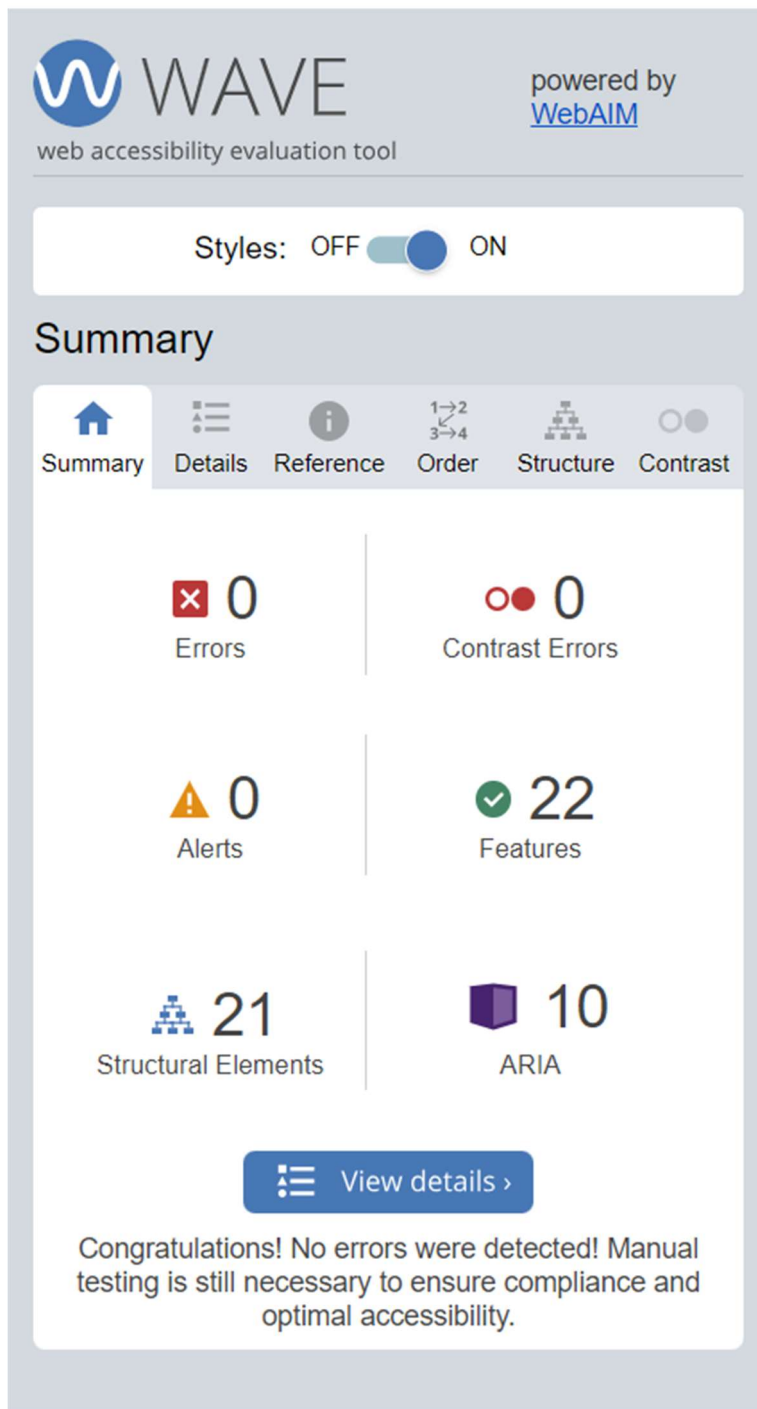
#### Accessibilité avant optimisation

*[Insérer une capture de la page du site avec l'extension Wave avant optimisation]*



#### Accessibilité après optimisation

*[Insérer une capture de page du site avec l'extension Wave après optimisation]*



## Modifications effectuées

*[Lister les modifications faites pour valider l'accessibilité]*

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

### 1. Structure HTML :

- **Balise h1** : Une seule balise h1 est maintenant présente sur chaque page pour améliorer la hiérarchie des titres et optimiser le SEO.
- **Formulaire** : Ajout de l'attribut `for=""` pour chaque élément `label` dans le formulaire, garantissant que chaque champ soit correctement lié à son label.

- **Footer :**
  - Ajout d'une balise footer pour structurer la fin de la page.
  - Ajout de contenu dans le footer, notamment des informations personnelles telles que l'adresse et les coordonnées.

## 2. CSS :

- **Optimisation du CSS :** Vérification des règles existantes pour supprimer les styles inutiles.
  - Suppression d'une règle CSS vide : `#gallery {}`.
- **Amélioration du contraste :** La couleur de fond change maintenant lors de la sélection d'un filtre dans le projet pour assurer un meilleur contraste et une meilleure lisibilité.
- **Création d'un footer personnalisé :** Le CSS a été ajusté pour le style du footer ajouté précédemment.

## 3. Minification :

- **Minification du fichier CSS :** Le fichier *style.css* a été minifié, créant ainsi un fichier optimisé *style.min.css*.
- **Minification du fichier JavaScript :** Le fichier *maugallery.js* a également été minifié, générant un fichier *maugallery.min.js*.
- **Suppression de Bootstrap local :** Le dossier Bootstrap local a été supprimé, et les liens et scripts Bootstrap sont maintenant chargés via un CDN, permettant une meilleure performance.

## 4. Accessibilité :

- **Accessibilité avant et après optimisation :** L'accessibilité du site a été améliorée
- **Amélioration du contraste des filtres :** La couleur des filtres a été ajustée pour offrir un meilleur contraste visuel.

## 5. Réalisations supplémentaires à la demande du client :

- **Footer avec informations de contact :** Un footer a été réalisé, contenant les informations de contact, l'adresse et d'autres détails pertinents.
- **Correction du formulaire :** Le formulaire a été corrigé pour être pleinement responsive, améliorant son comportement sur les différents types d'appareils.

## 6. Recette et débogage :

- **Boutons de filtres de catégorie :**
  - **Problème initial :** Le bouton de catégorie "TOUS" gardait bien sa couleur de fond, mais en sélectionnant une autre catégorie, la couleur ne se transférait pas sur le bouton cliqué.
  - **Résultat attendu :** Le background doit changer du bouton actif à celui sélectionné.
  - **Résolution :** Une correction dans le fichier *maugallery.js* a été effectuée, ajoutant la classe `.active` pour permettre le changement de background.
- **Navigation entre les photos de la galerie via modales :**
  - **Problème initial :** Les modales ne permettaient pas de naviguer entre les images.
  - **Résultat attendu :** Permettre de passer d'une image à l'autre avec les flèches dans les modales.
  - **Résolution :** Les fonctions `prevImage()` et `nextImage()` ont été ajustées avec les lignes :
    - `index = i + 1;`
    - `index = i - 1;`

## 7. Annexe :

- Un audit complet via Lighthouse a été réalisé pour vérifier la performance du site sur ordinateur.

Ces modifications visent à optimiser les performances, l'accessibilité, la structure, et l'ergonomie du site, tout en répondant aux besoins spécifiques du client.

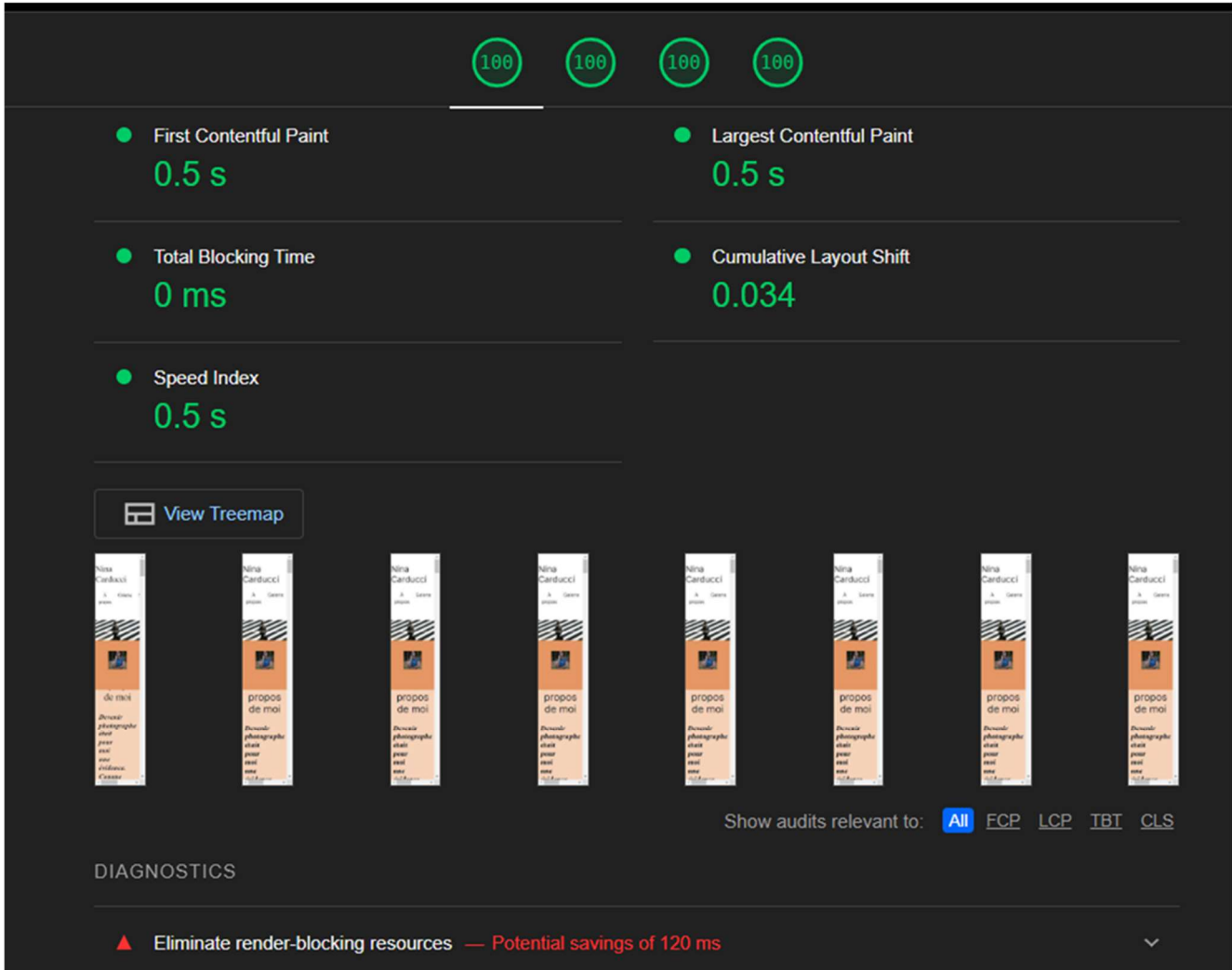
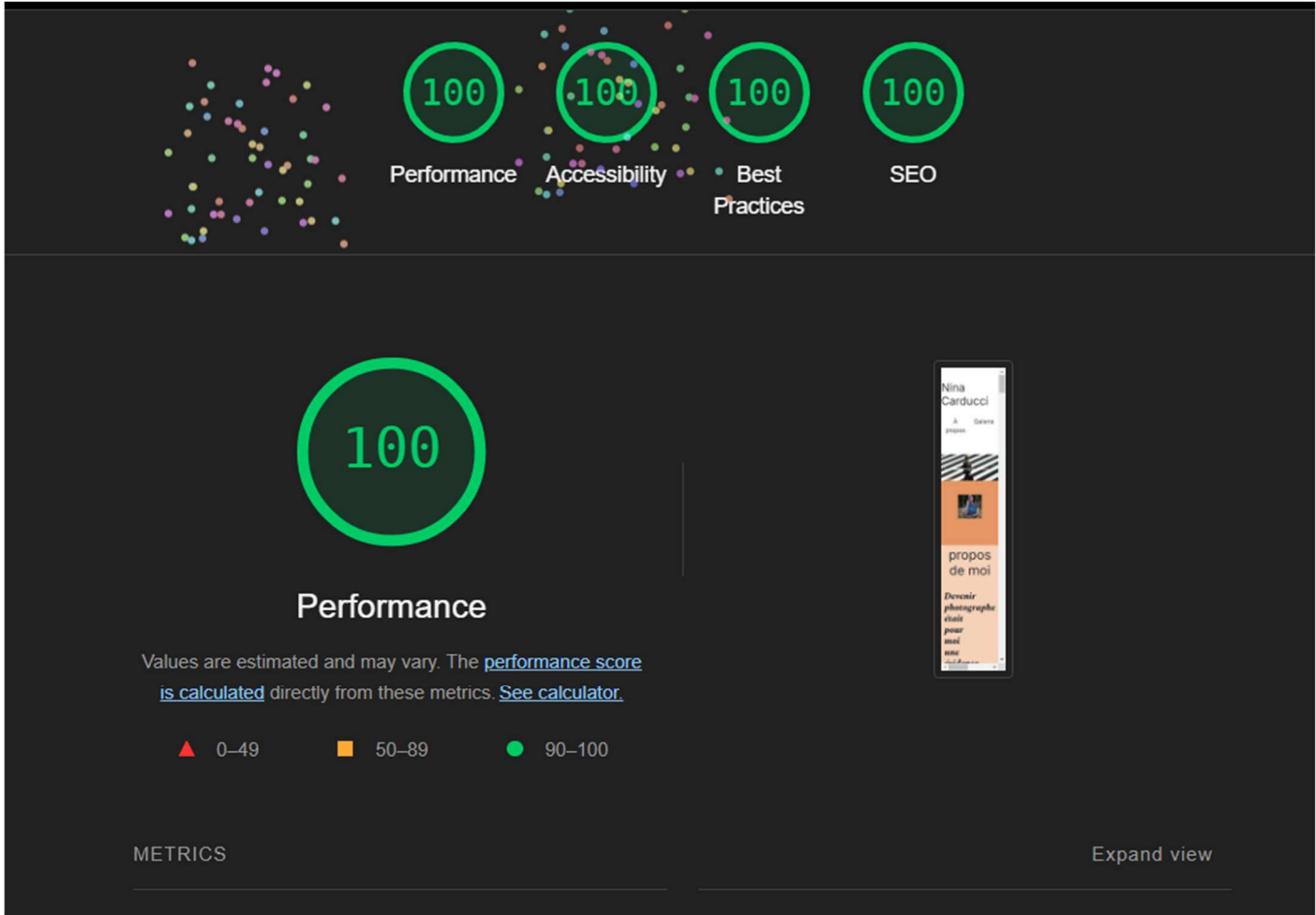
## V - Cahier de recette

Détail des fonctionnalités débuggées et de leur statut :

ID	Action	Résultat initial		Statut	Remarques et commentaires
1	Bouton de filtres de catégorie	Le bouton "TOUS" gardait bien sa couleur de fond, mais en sélectionnant une autre catégorie, la couleur disparaissait et ne se transférait pas sur le bouton cliqué.	La couleur de fond doit changer entre le bouton actif et celui sélectionné.	Résolu	Correction dans le fichier <i>maugallery.js</i> en ajoutant la classe <code>.active</code> , permettant ainsi le changement de couleur du bouton sélectionné.
2	Navigation entre les photos de la galerie via les modales	Impossible de naviguer entre les images via les flèches des modales.	Pouvoir passer d'une image à l'autre en utilisant les flèches de navigation.	Résolu	Les fonctions <i>prevImage()</i> et <i>nextImage()</i> ont été corrigées en modifiant l'index avec les lignes : <code>index = i + 1;</code> <code>index = i - 1;</code>

## Annexe

### Rapport complet de l'audit Lighthouse





100

100

100

100

○ Minimizes main-thread work — 0.4 s

○ Minimize third-party usage — Third-party code blocked the main thread for 0 ms

○ Largest Contentful Paint element — 510 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.

PASSED AUDITS (24)

Show

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)

Show

100

100

100

100

▲ Eliminate render-blocking resources — Potential savings of 120 ms

■ Serve static assets with an efficient cache policy — 7 resources found

■ Properly size images — Potential savings of 66 KiB

■ Reduce unused CSS — Potential savings of 21 KiB

○ Avoid large layout shifts — 1 layout shift found

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

○ Avoids enormous network payloads — Total size was 275 KiB

○ Avoids an excessive DOM size — 165 elements

○ Avoid chaining critical requests — 10 chains found

○ JavaScript execution time — 0.1 s

○ Minimizes main-thread work — 0.4 s

○ Minimize third-party usage — Third-party code blocked the main thread for 0 ms

100100100100

encouraged.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Show

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

100

Best Practices

TRUST AND SAFETY

○

Ensure CSP is effective against XSS attacks

▼

100100100100

Best Practices

TRUST AND SAFETY

○

Ensure CSP is effective against XSS attacks

▼

GENERAL

○

Detected JavaScript libraries

▼

PASSED AUDITS (14)

Show

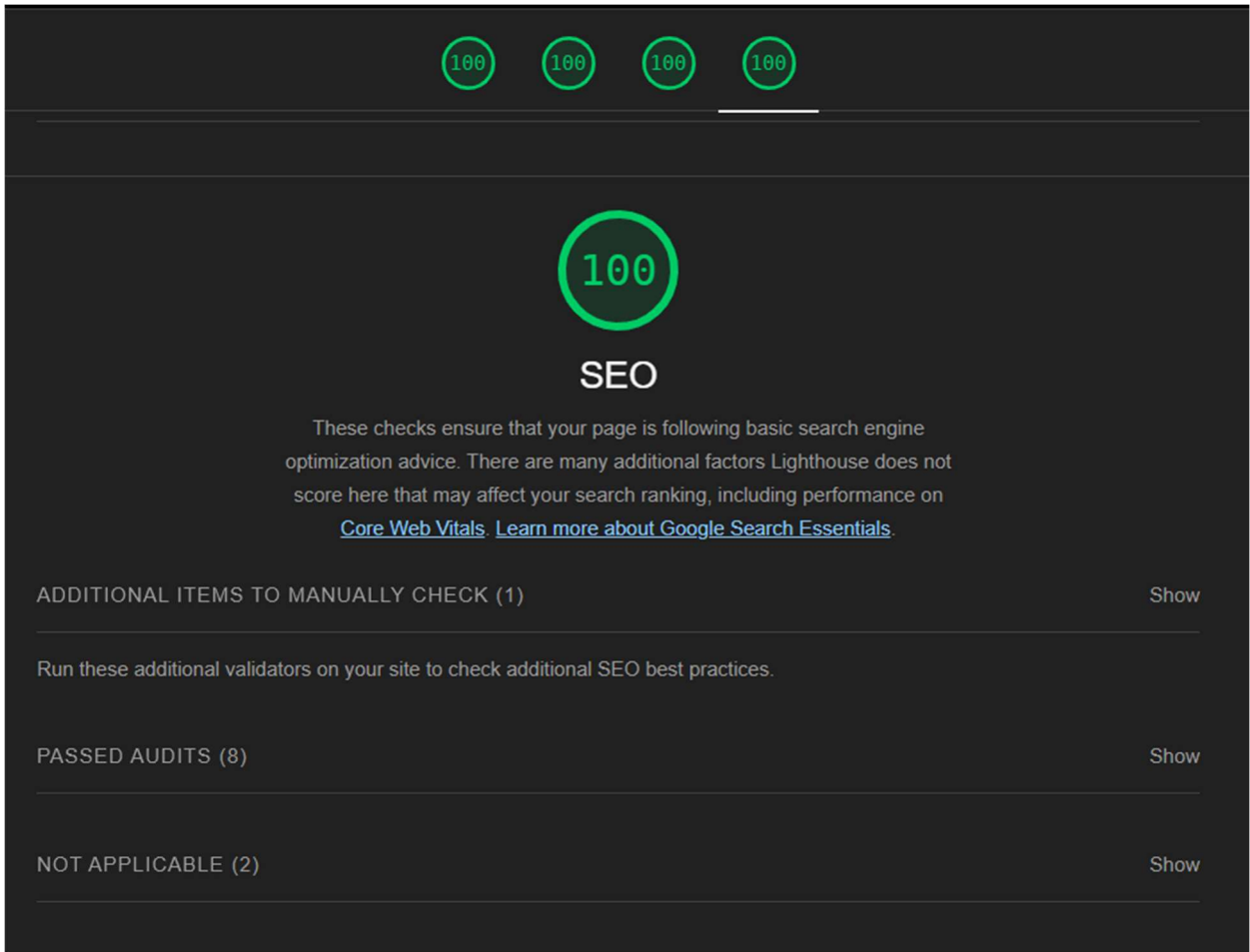
NOT APPLICABLE (2)

Show

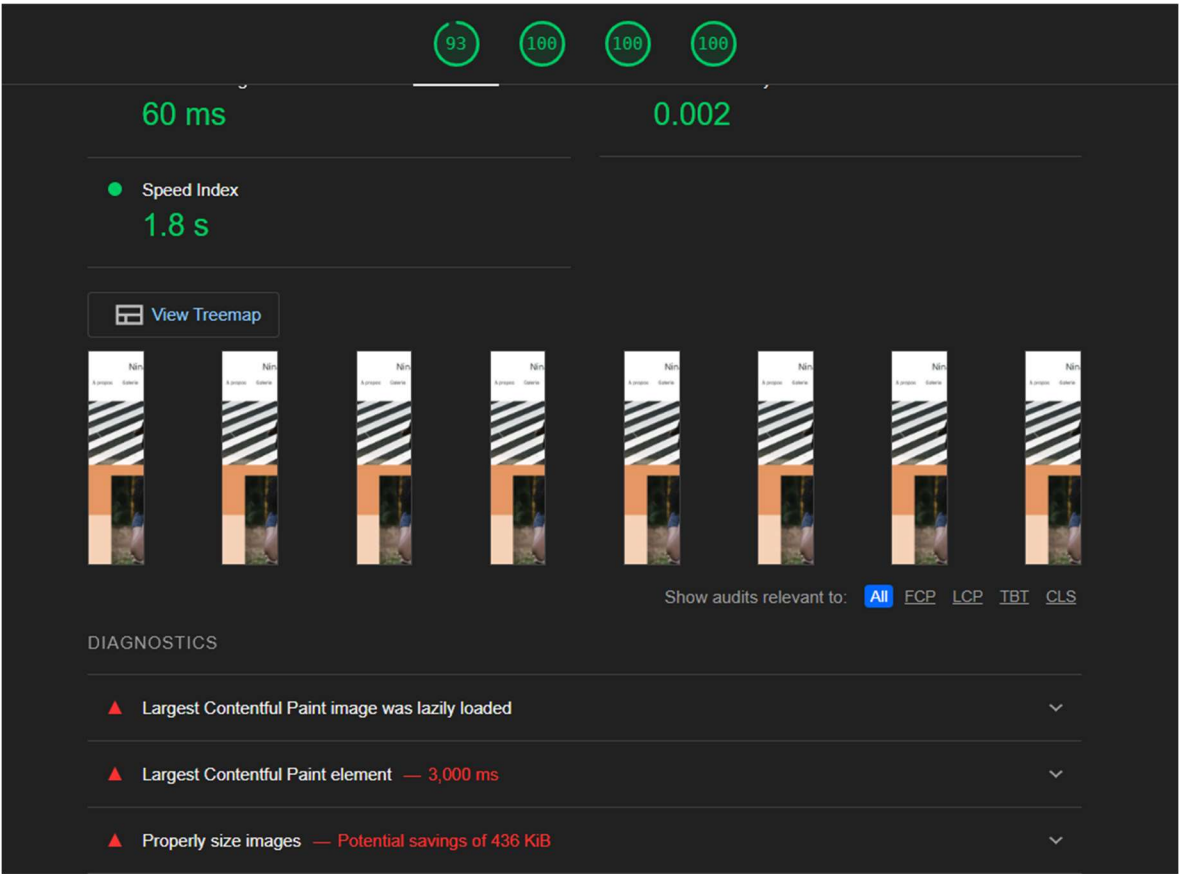
100

SEO

These checks ensure that your page is following basic search engine



Mobile :



93			100	100	100
▲	Properly size images	— Potential savings of 436 KiB	▼		
▲	Eliminate render-blocking resources	— Potential savings of 700 ms	▼		
▲	Reduce unused CSS	— Potential savings of 33 KiB	▼		
■	Minify JavaScript	— Potential savings of 16 KiB	▼		
■	Serve static assets with an efficient cache policy	— 9 resources found	▼		
■	Reduce unused JavaScript	— Potential savings of 31 KiB	▼		
○	Avoid large layout shifts	— 2 layout shifts found	▼		
○	JavaScript execution time	— 0.3 s	▼		
○	Minimizes main-thread work	— 1.7 s	▼		
○	Minimize third-party usage	— Third-party code blocked the main thread for 70 ms	▼		
○	Avoid long main-thread tasks	— 5 long tasks found	▼		
○	Initial server response time was short	— Root document took 10 ms	▼		

93

100

100

100

○	Initial server response time was short	— Root document took 10 ms	▼
○	Avoids enormous network payloads	— Total size was 691 KiB	▼
○	Avoids an excessive DOM size	— 165 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.

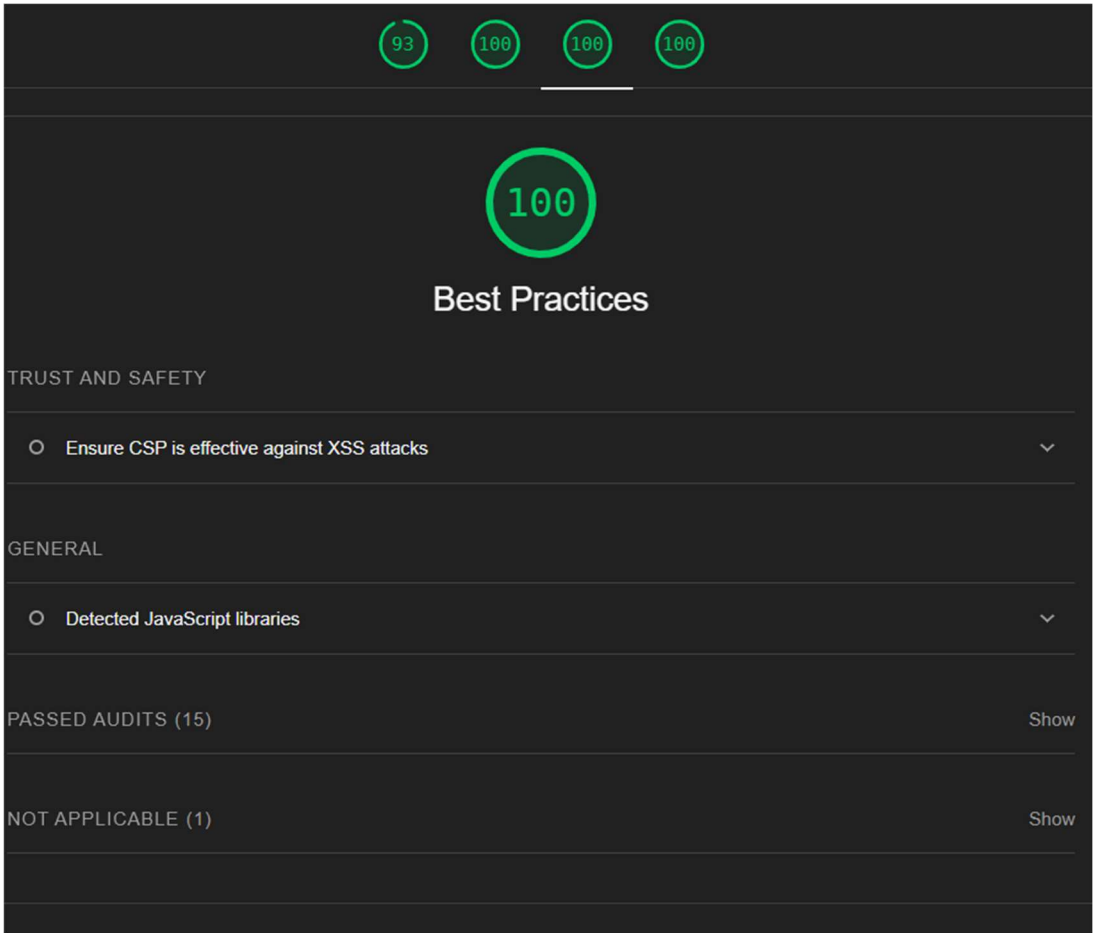
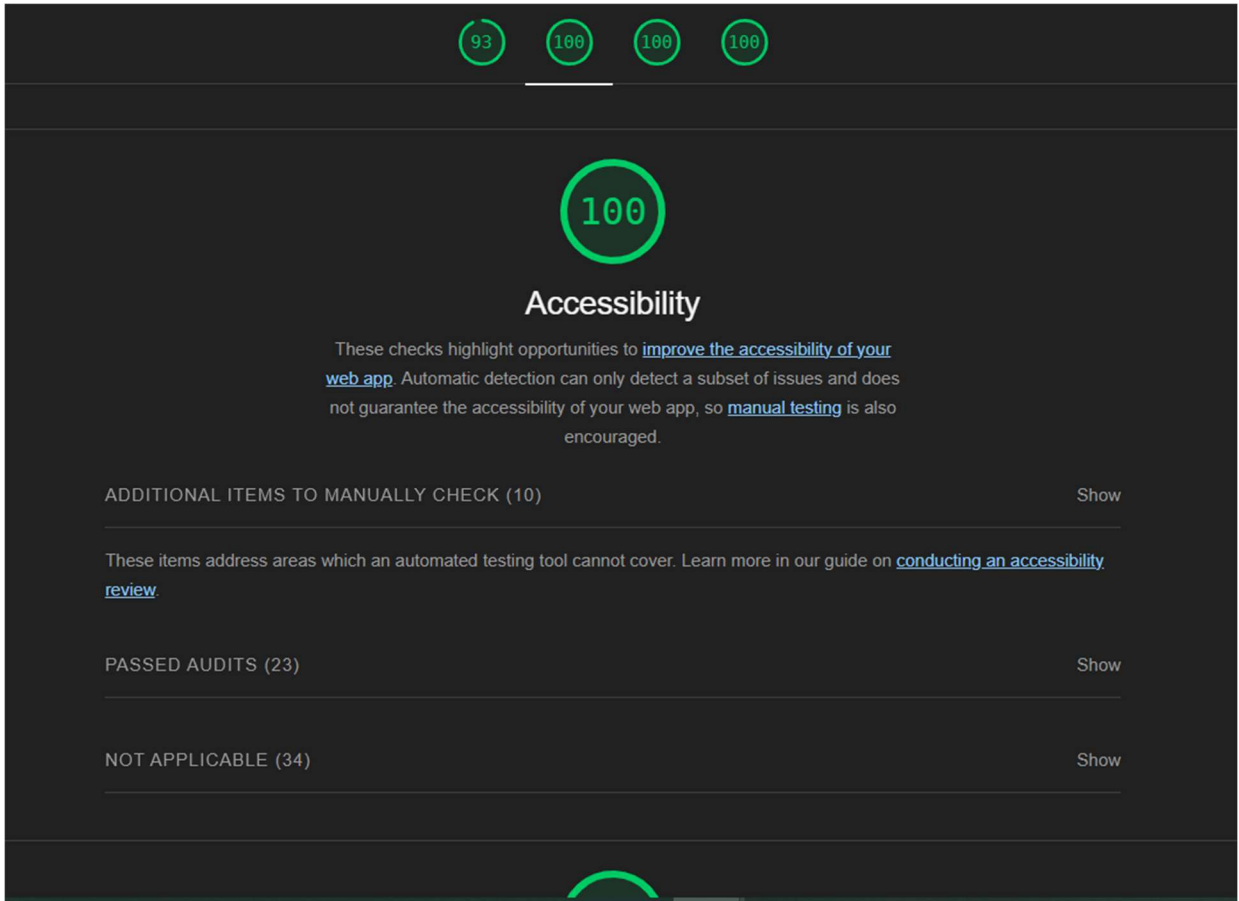
PASSED AUDITS (21)

Show

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.



93

100

100

100

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)

Show

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

## PASSED AUDITS (8)

Show

## NOT APPLICABLE (2)

Show

Captured at Sep 25, 2024

Emulated Moto G Power with

Single-page session