Rapport d’intervention

NINA\_CARDUCCI

[**I - Comparatif avant et après optimisation**](#_n8jvrnml41oj) **2**

[**II - Détails des optimisations effectuées**](#_75j88ale97cb) **2**

[1 - Les images](#_uxfyskso5n4s) 2

[2 - …](#_xkbpxkl6umhk) 2

[**III - Accessibilité du site**](#_r7gkf09frlj5) **2**

[**IV - Détails de réalisations additionnelles à la demande du client**](#_s89pup9bbtic) **2**

[1 - …](#_gt5hgt2h0fn6) 2

[**Annexe**](#_w04kirgfeg7j) **3**

[Rapport complet de l’audit Lighthouse](#_3m019n8dyixe) 3

# I - Score Lighthouse

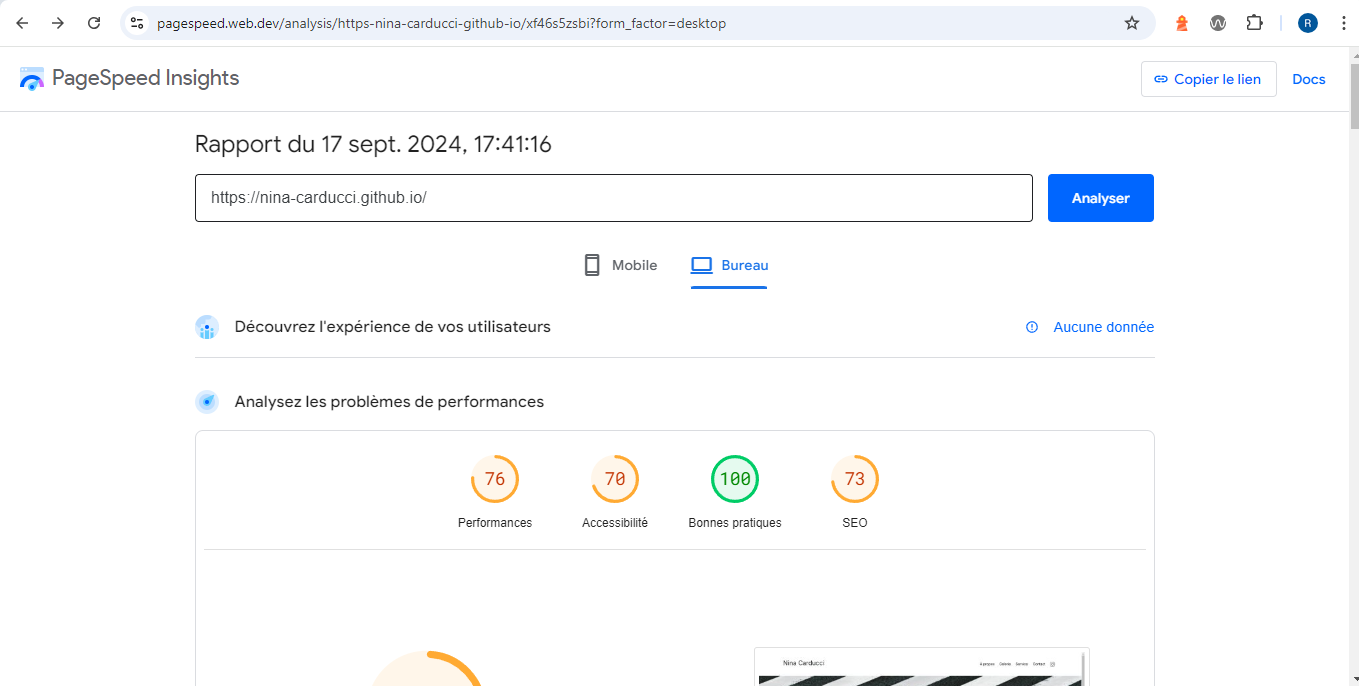
Lighthouse est un outil de mesure et d’analyse développé par Google pouvant permettre aux utilisateurs d’examiner et de déterminer la performance, l’accessibilité et l’optimisation du SEO du site ainsi que des bonnes pratiques générales exploités lors du développement de l’application.

## 

## Score Lighthouse avant optimisation

Il m’appartient de présenter une capture des scores Lighthouse avant optimisation pour le site de Nina. Le lien du site est : [*https://nina-carducci.github.io/*](https://nina-carducci.github.io/)*.*

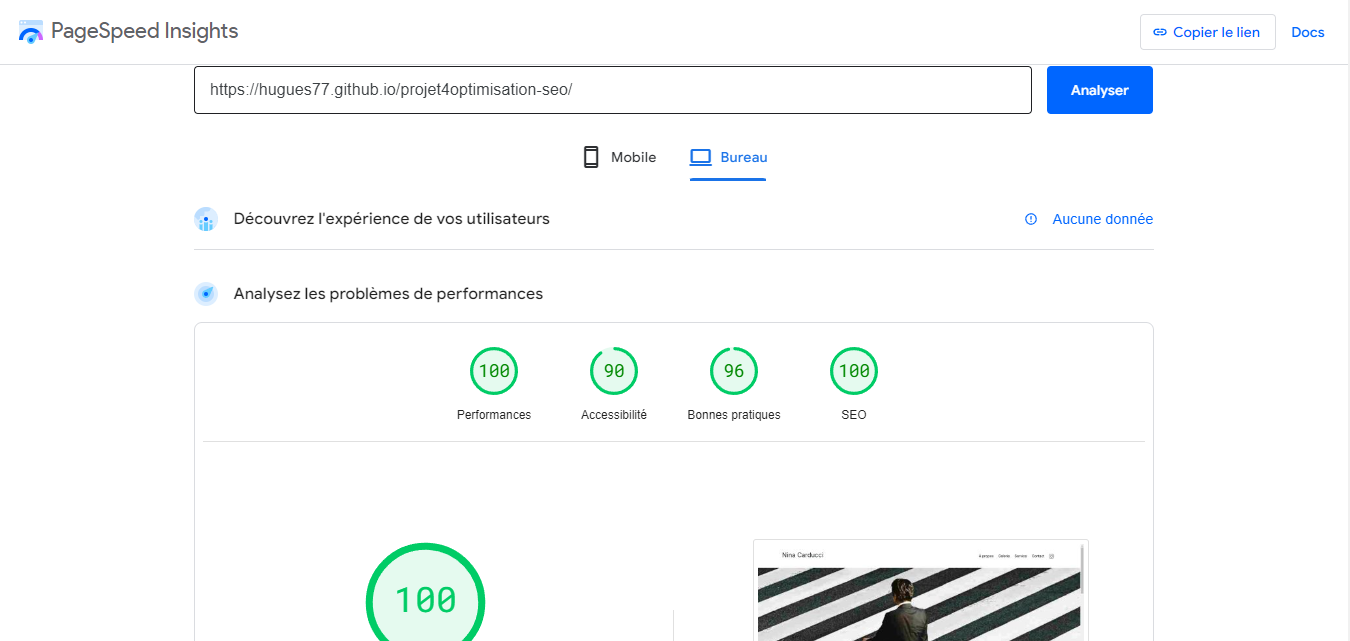
En voici le résultat :



## Score Lighthouse après optimisation

Après mon intervention, mon expertise m’a permis d’améliorer le résultat en utilisant plusieurs techniques sémantiques et beaucoup des test pour arriver dans ce merveilleuxrésultat. Voici la capture des scores Lighthouse après optimisation.

Le lien du site est le suivant : <https://hugues77.github.io/projet4optimisation-seo/>



# II - Détails des optimisations et interventions effectuées

## 1 - Les images

Le projet comporte originellement ***15*** images pour un poids total de ***29.4*MB**. Nous avons effectué les modifications suivantes aux images :

* Ajuster la résolution des images via le logiciel Photoshop c’est-à-dire réduire la taille de chaque image en vue de faire correspondre chacune à la taille du contenant et de l’exporter pour le WEB, par exemple : les 3 images de carrousels ayant la résolution d’avant de 6000px/2775px, 5653px/2613px et 4540px/2100px et je les ai tous ramener à 1900px/879px et les autres images de la gallery y compris celle de nina ont pu avoir aussi un ajustement au niveau de la résolution image.

Ce qui a conduit automatiquement à la diminution des poids des images ; A ce stade, mes 15 images venaient de totaliser un poids total de 989Ko

* Compression des images via le site <https://imagecompressor.11zon.com/fr/image-compressor/> pour obtenir encore un poids incroyable pouvant alléger mes pages web avec des qualités d’images non loin de la réalité.

Le poids final de mon dossier image à chuter jusqu’à atteindre **147Ko**, c’est tellement magique !

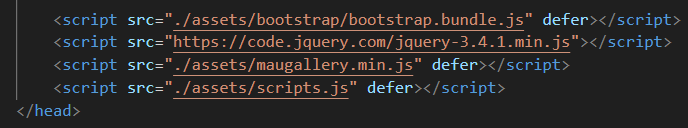
* Pour gagner un temps record lors de chargement de mes images c’est-à-dire réduire le temps de chargement initial de la  
  page bien que compresser, je fais le choix d’utiliser l’attribut **loading="lazy"** dans la balise **<img>** pour l’ensemble des images de la gallery

Après les modifications, le poids total des images est de **0.147MB**, soit un gain de ***99.5*%.**

## 2 – Minifier les fichiers

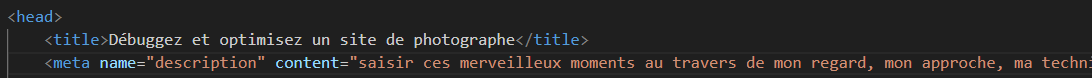
Après constatation de mes fichiers, j’ai résolu de minifier tous mes fichiers sources via ce lien du site : <https://www.minifier.org/>

* Minifier le fichier CSS (style.css)
* Minifier le fichier JS (maugallery.js)
* Utilisation du fichier minifier du bootstrap (bootstrap.min.css)
* Insérer **defer** à chaque script JS pour réduire le temps de chargement initial



## 3 – Gérer le référencement naturel SEO

* Ajout de la balise **title** pour le titre de la page principale (Index.html)
* Ajout de la balise **meta description** pour permettre aux moteurs de recherche de bien afficher la description du site de Nina.



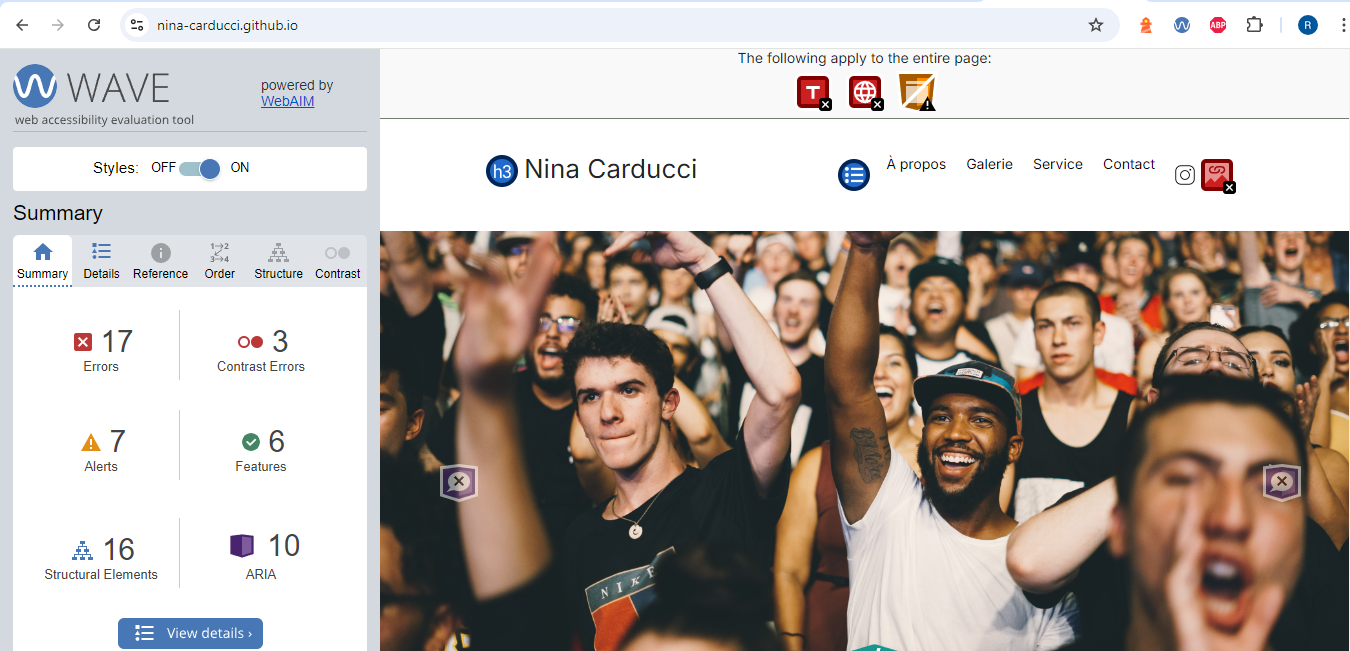
* Formaliser le contenu par des balises sémantiques en ajoutant **header**, **main** et **footer** pour chaque partie respective
* Hiérarchisez et structurer les différents titres (headings) de la page c’est-à-dire du H1 jusqu’à H6

# III - Accessibilité du site

WAVE est l’acronyme de Web Accebility Evaluation  un outil gratuit d'évaluation de l'accessibilité du Web qui identifie les moyens de rendre une page Web plus accessible aux personnes handicapées (mal voyants par exemple) . il joue sur une forme plus visuelle et interactive pour lister des erreurs du code

## Accessibilité avant optimisation

Ayant testé l’accessibilité du site au travers l’outil wave j’obtiens un résultat très controverser avec **17** erreurs et **3** erreurs de contraste, je vous présente une capture de la page du site avec l’extension Wave avant optimisation.

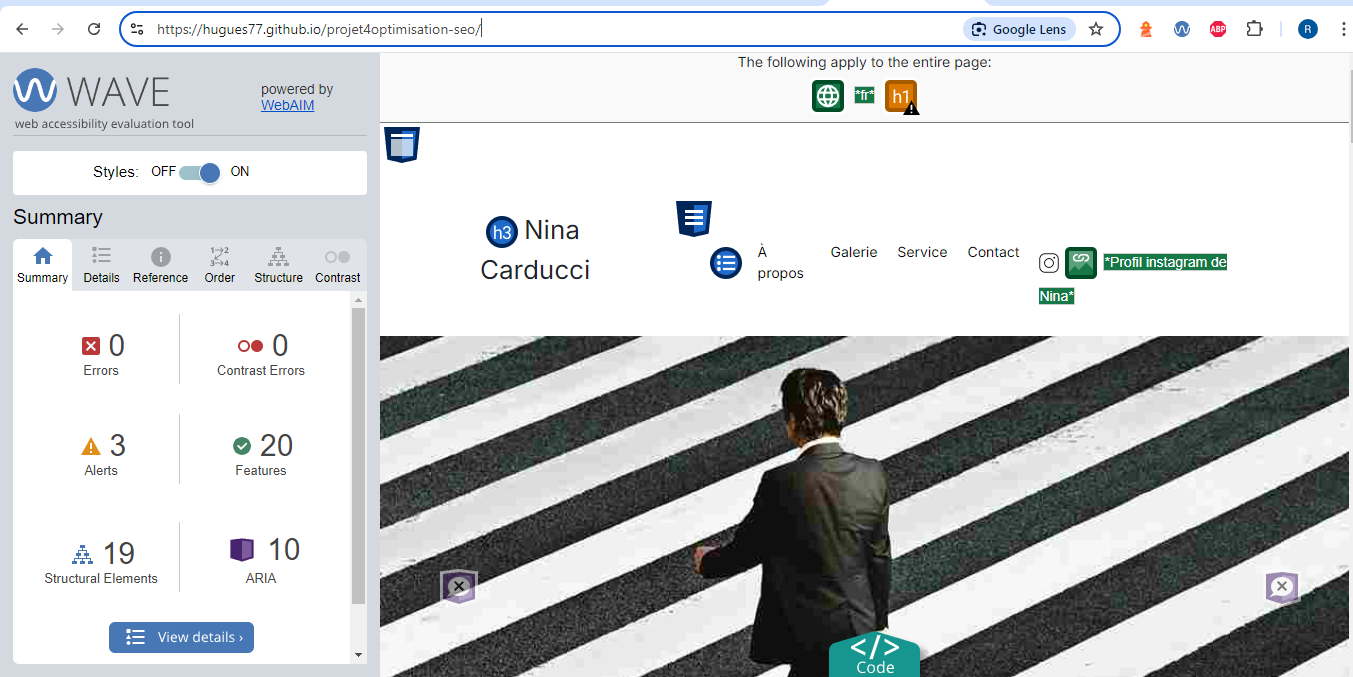


## Accessibilité après optimisation

Mon intervention sur le projet du coté accessibilité, je réalise beaucoup des modifications et ceci m’a amené à atterrir à un résultat beaucoup plus agréables dont **0** erreurs et **0** erreurs contraste

*En voici le capture d’écran ; toujours avec le lien fourni précédemment*

[*https://hugues77.github.io/projet4optimisation-seo/*](https://hugues77.github.io/projet4optimisation-seo/)

**

## Modifications effectuées

1 – Audit Wave

Après analyse du site via l’outil wave, le panneau latérale gauche m’a fourni l’ensemble des erreurs et des erreurs liés aux contraste du site, des alertes, et autres

Et au milieu du site, j’ai l’ensemble des icônes qui font référence aux erreurs trouvés dans le code ; entre autres :

* La langue du site n’était pas définie
* Les différents textes alternatives dans toutes nos balises **<img>** c’est-à-dire dans nos images du site
* Il manquait des balises label dans notre formulaire du contact du site
* La structure du site n’était pas non plus respectée
* Les couleurs utilisés dans le site n’étaient pas adaptés pour le contraste du site

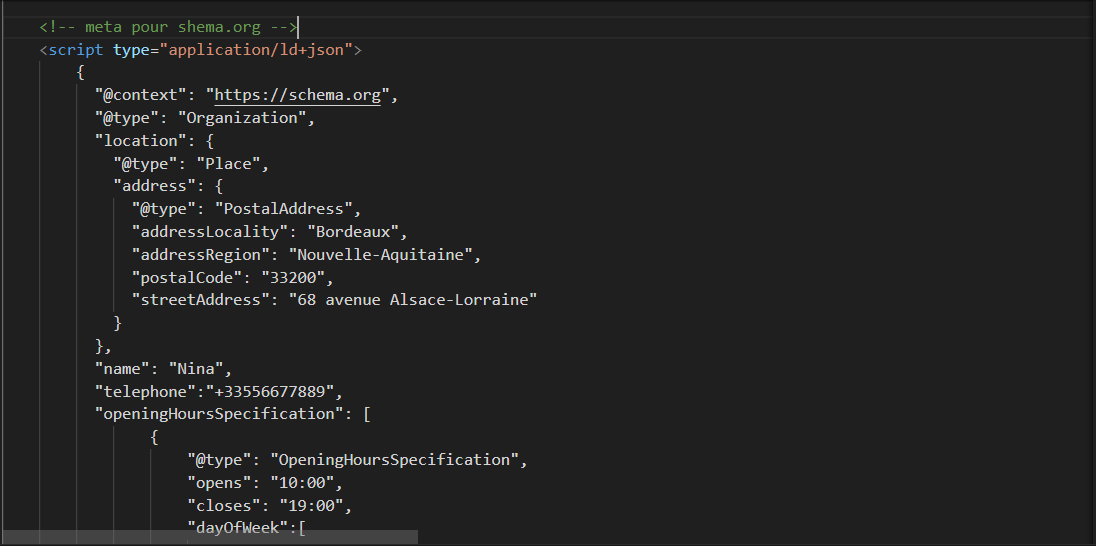
2 – Modifications apportées pour éliminer les erreurs

* Définir le français comme la langue du site en ajoutant à l’intérieur du balise **<html>** l’attribut **lang = ‘fr’.** celle-ci est utile pour permettre aux moteurs de recherche de comprendre la langue utilisée dans le site
* Ajout des textes alternatifs dans chaque balise **<img>** en ajoutant l’attribut alt. Celui-ci aidera également aux visiteurs du site de comprendre le contenu de chaque image en cas de bug ou latence lié à sa connexion internet
* L’ajout des labels dans notre formulaire de contact pour améliorer l’expérience utilisateur, il est vrai que le site avait des **labels** mais non reliés avec les **inputs** du formulaire, c’est ainsi que avec **l’id** de chaque input, j’ai pu compléter les labels en mettant un **for** pour lier chaque input ensemble avec son label.
* Changement des couleurs de l’arrière-plan du formulaire en bleu **« #1C6474 »** et du menu de filtres pour la catégorie en cramoisi **« crimson »** et des textes en blanc **« #fff »** pour améliorer la contraste utilisateurs

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

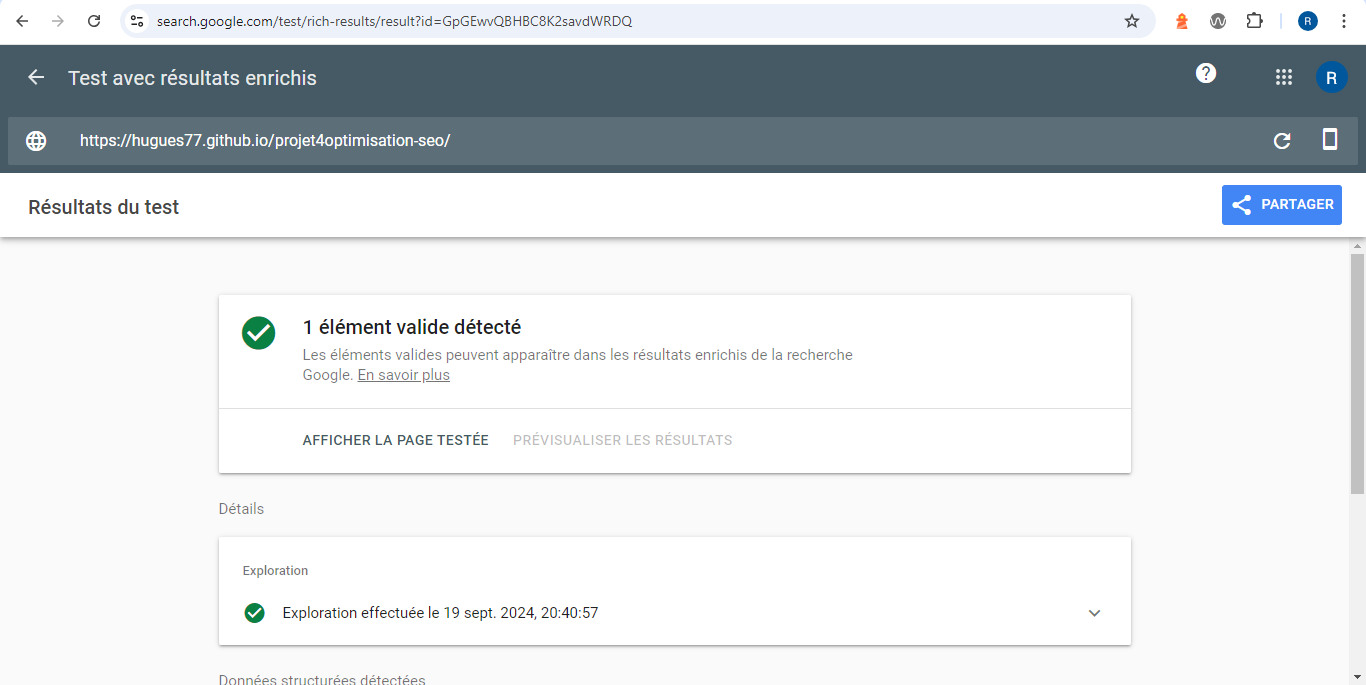
Il est certain que lors de mon entretien téléphonique avec mon client Nina Carducci, nous nous sommes convenu au-delà de l’optimisation globale du site, de la performance et du SEO comme cela a été signalé ci-haut ; il m’avait aussi été demandés de mettre en place **le** **référencement local**, d’ajouter des **balises** **métas pour les réseaux sociaux,** de corriger **la** **navigation dans la modale de la galerie** et de préciser à chaque changement de filtre, la catégorie sélectionnée.

* Le référencement local : Shema.org est un outil indispensable de balisage sémantique pour permettre aux moteurs de recherche de mieux comprendre le contenu d’un site web. Il facilite l’utilisation et l’intégration de tous les types de données structurés. J’ai ajouté le script de schema.org dans le **<head>** de la page principale pour répondre à la demande du client ; voici la capture du bout de code su script.

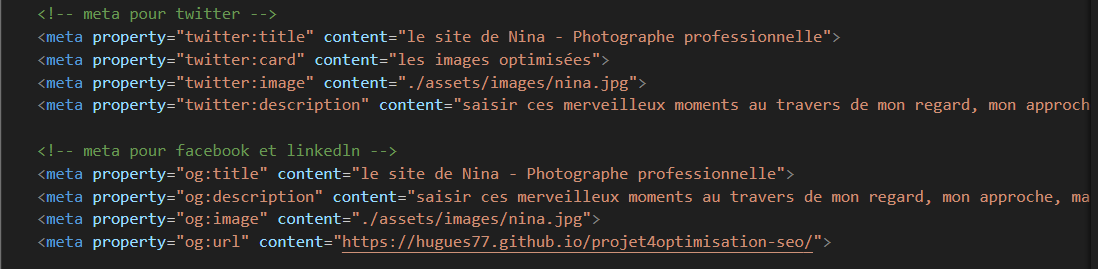


Audit Google Rich Snippet

Apres ajout du script, il est impératif de tester le balisage schéma.org au travers le test avec résultat enrichie de Google. Et comme vous pouvez constater le résultat est bel et bien positif.



* Les balises pour métas pour les réseaux sociaux : Appelles aussi balises méta property, elles sont utilisées pour ajouter des métas données supplémentaire dans le site pour les réseaux sociaux. Elle permet aussi de maitriser et de contrôler l’affichage de la page en optimisant les informations du type titre, image et description sur les réseaux sociaux lors du partage.

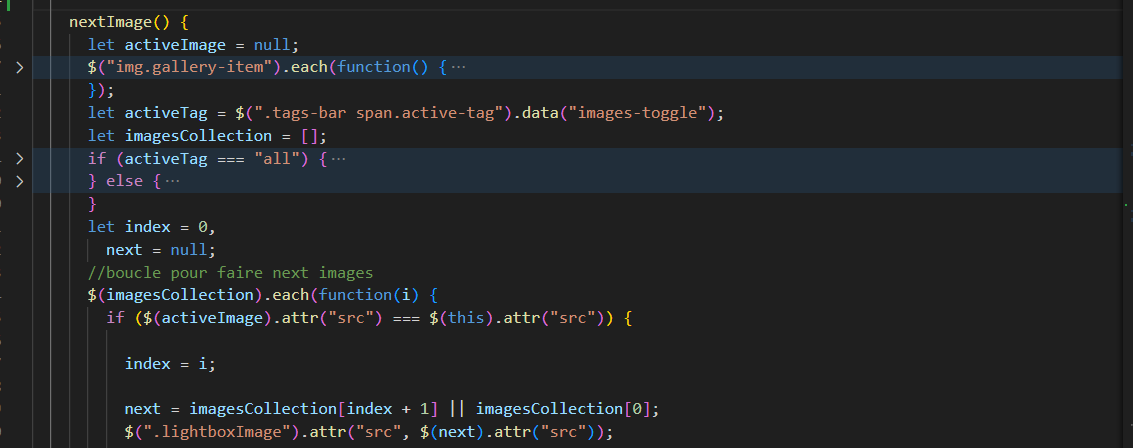


* Corriger la navigation de la modale : il était question de très bien étudier le code dans son entièreté et de corriger les bugs et les erreurs dans le code ; en parcourant l’ensemble du fichier du script **«** **maugallery.js** **»** en ligne 61 et 64, je retrouve les scripts pour lancer les functions **prevImage**  et **nextImage** du modale de la gallery

C’est ainsi que j’y vais à chaque function respective pour corriger le bug liés au boucle de la navigation des images.

En voici la capture d’écran:



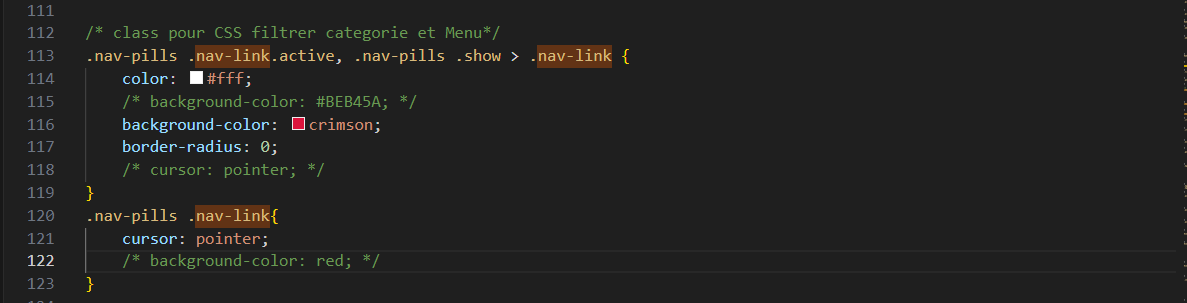


* Préciser chaque changement de filtre : Lorsque l’on change de filtre pour afficher les images, il était question d’ajouter un bout de code pour permettre aux visiteurs de savoir à quelle catégorie ils se retrouvent lors de la navigation du filtre (gallery)

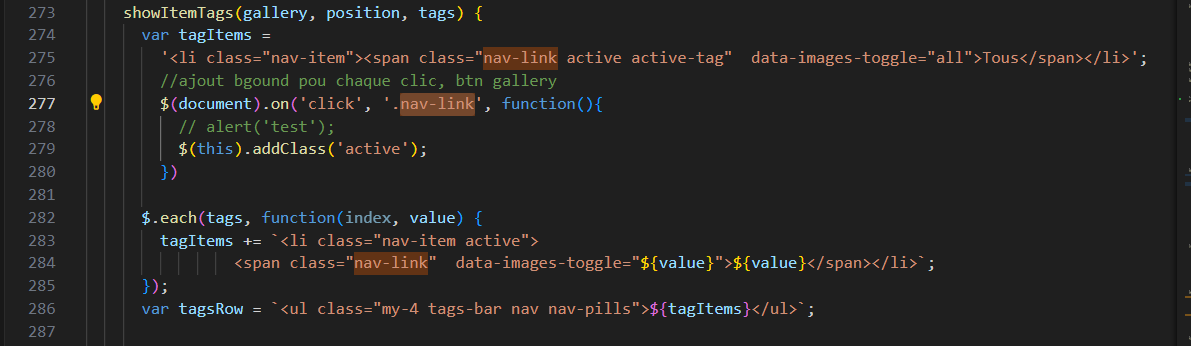
Premièrement, il était question de cibler la classe qui permettent de créer l’arrière-plan de chaque lien **« nav-link »** et faire qu’à chaque clic du visiteur, injecter une classe active pour garder le background du lien.

En voici la capture d’écran :

CSS



JS



# V - Cahier de recette

Détail des fonctionnalités débuggées et de leur statut :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Action | Résultat initial | Résultat attendu | Statut | Remarques et commentaires |
| 1 | Clique sur le filtre | Pas d’ajout de précision sur la catégorie sélectionnée | Afficher le couleur d’arrière-plan pour préciser la catégorie | Résolu |  |
| 2 | Clique sur les flèches de la navigation modale | Pas de navigation par flèche dans le modal | Naviguer pour voir différentes photos de la gallery depuis la modal | Résolu |  |

# Annexe

## Rapport complet de l’audit Lighthouse