



Portfolio

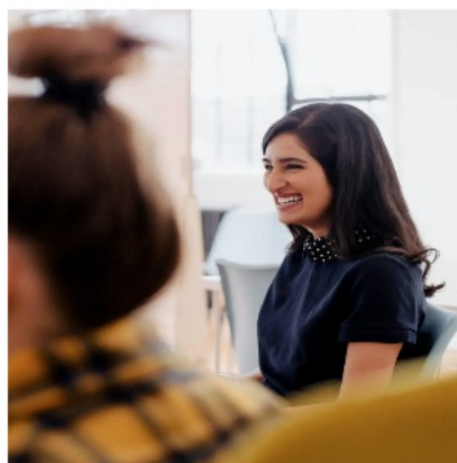
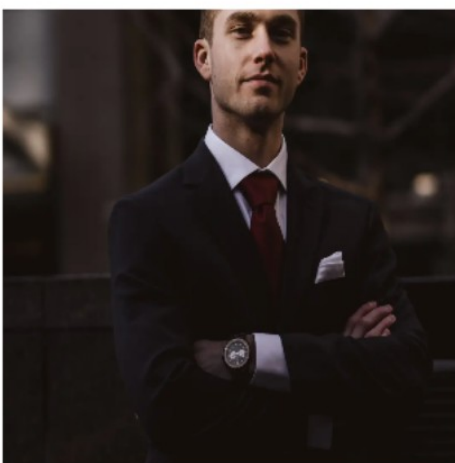
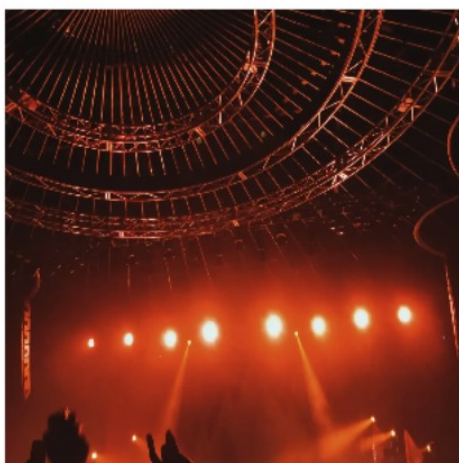
Tous

Concert

Entreprises

Mariages

Portrait



OPENCLASSROOMS

PAR VINCENT FUSEAU

RAPPORT

PROJET 4

FORMATION DÉVELOPPEUR WEB

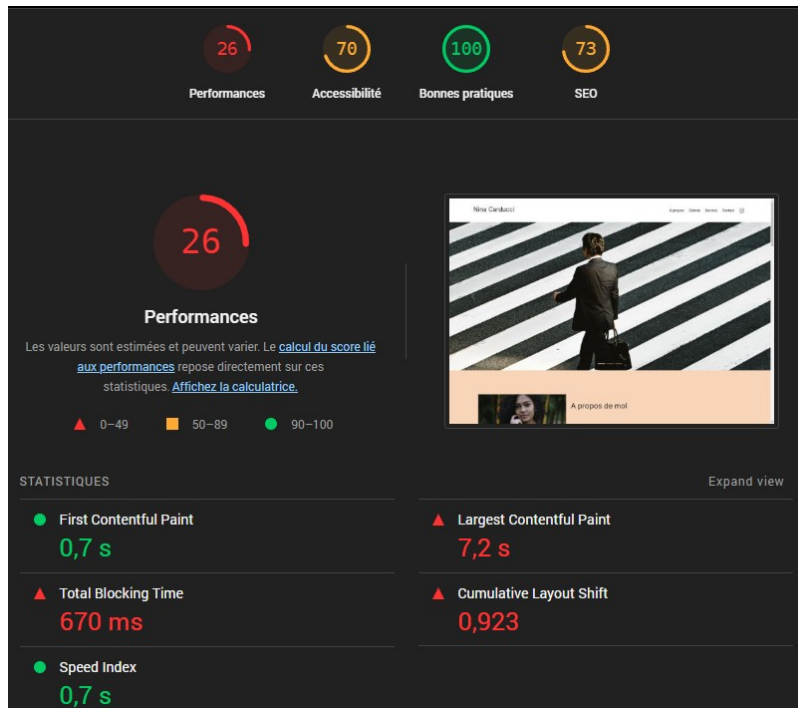
JANVIER 2025

Table des matières

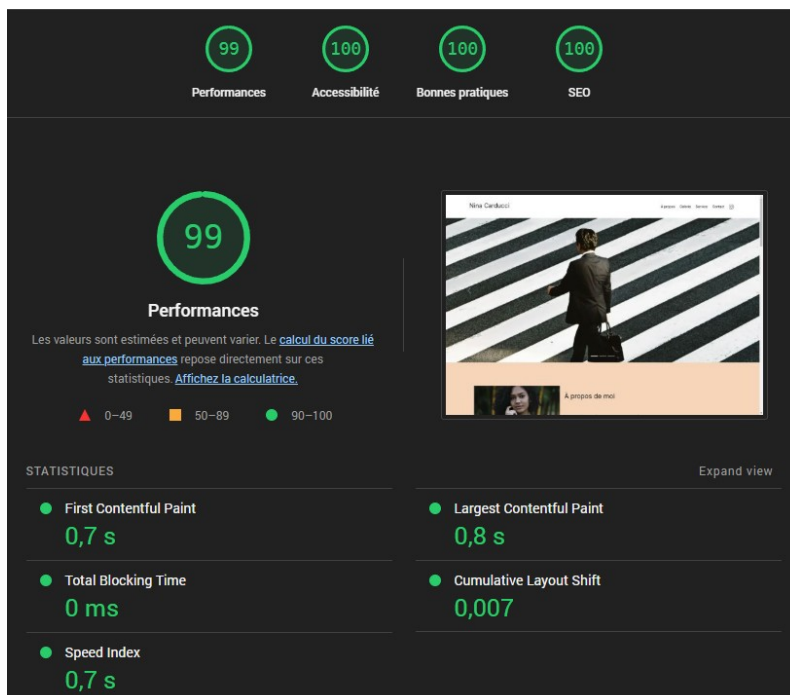
I. Comparatif avant et après optimisation.....	3
1. Score Lighthouse avant optimisation.....	3
2. Score Lighthouse après optimisation.....	3
II. Optimisations et performances.....	4
1. Optimisation des images.....	4
2. Bug de la modale (navigation entre les photos).....	4
3. Bug des filtres (affichage correcte des filtres lors de la sélection).....	7
4. Minimisation des fichiers CSS et JS :.....	9
5. Ajout des balises meta pour les réseaux sociaux :.....	10
6. Implémentation des Rich Snippets :.....	11
III. Accessibilité du site.....	13
1. Accessibilité avant optimisation.....	13
2. Accessibilité après optimisation.....	15
3. Correction des erreurs révélées par le validateur W3C.....	15
4. Ajustement des contrastes dans la navlink :.....	17
5. Correction des éléments d'en-tête non classés séquentiellement :.....	18
6. Texte visuellement masqué dans les boutons de contrôle du carrousel :.....	19
7. Ajout d'un attribut for dans les labels du formulaire :.....	20
8. Remplacement de <div class="header"> par <header> :.....	21
9. Ajout d'une balise méta description :.....	21
IV. Cahier de recette.....	23
V. Annexe.....	25
1. Rapport complet de l'audit Lighthouse.....	25

I. Comparatif avant et après optimisation

1. Score Lighthouse avant optimisation



2. Score Lighthouse après optimisation



II. Optimisations et performances

1. Optimisation des images

- **Problème initial :**

- Le projet comportait 18 images au format traditionnel (JPEG/PNG) pour un poids total de **22 730,4 KiB**, ce qui impactait la vitesse de chargement des pages.

- **Actions entreprises :**

- Conversion des images en formats modernes (WebP) pour améliorer les performances.
- Utilisation de la commande suivante pour automatiser la conversion des fichiers JPEG en WebP avec une qualité de compression de 70 % :

```
for img in *.jpg; do
    if [ -f "$img" ]; then
        cwebp -q 70 "$img" -o "${img%.*}.webp"
        echo "Converti $img en ${img%.*}.webp"
    fi
done
```

- La commande a permis :
 - Une conversion rapide et efficace des fichiers.
 - La génération de fichiers WebP avec un poids réduit, tout en maintenant une qualité visuelle acceptable.

- **Résultat :**

- Après optimisation, le poids total des images est réduit à **770,1 KiB**, soit un gain de **plus de 22 000 KiB**.
- Les temps de chargement des pages ont été améliorés, et les images sont désormais compatibles avec les navigateurs modernes supportant le format WebP.

2. Bug de la modale (navigation entre les photos)

- **Problème initial :**

- Les utilisateurs ne pouvaient pas passer d'une image à l'autre via les boutons "Précédent" et "Suivant". Les actions de navigation ne produisaient aucun effet ou renvoyaient des erreurs.
- Les événements associés aux boutons de navigation n'étaient pas correctement liés aux fonctions JavaScript. Cela a conduit à une mauvaise gestion des actions de l'utilisateur.
- L'index des images affichées dans la modale n'était pas correctement mis à jour lors des changements d'image, ce qui créait des incohérences.
- Lorsqu'une image était affichée, la modale ne se mettait pas à jour de manière fluide, et l'image suivante/précédente n'était pas correctement chargée.

- **Constats techniques :**

- Identification des événements de navigation :

La navigation reposait sur des événements (clic sur les boutons "Précédent" et "Suivant"), mais leur gestion était défectueuse. Les événements n'étaient pas correctement associés aux actions.

- Index des images :

La logique de navigation reposait sur un index dynamique pour déterminer quelle image devait être affichée en fonction des interactions de l'utilisateur. Si l'index n'était pas bien géré, la navigation ne fonctionnait pas comme prévu.

- Synchronisation avec la modale :

La modale ne se synchronisait pas correctement avec l'image actuelle. Chaque fois que l'utilisateur tentait de naviguer, l'image affichée dans la modale ne se mettait pas à jour comme prévu.

- **Actions correctives apportées dans le nouveau code :**

- Mise en place d'un index global :

Pour gérer l'index de l'image affichée dans la modale, un attribut global `activeImage` a été introduit dans la fonction `prevImage` et `nextImage`. Ça permet de suivre l'image actuellement affichée et d'effectuer la navigation en toute cohérence.

```
let activeImage = $(".lightboxImage").attr("src");
let imagesCollection = this.getImagesCollection();
let index = imagesCollection.indexOf(activeImage);
```

La variable `activeImage` récupère la source de l'image actuellement affichée dans la modale. Ensuite, `imagesCollection` récupère toutes les images de la galerie pour permettre de calculer l'image suivante ou précédente en fonction de l'index.

- Gestion des événements de navigation :

Les boutons "Précédent" et "Suivant" sont reliés à des fonctions qui récupèrent l'index de l'image actuelle, calculent l'image suivante/précédente, et mettent à jour la modale.

```
$(".gallery").on("click", ".mg-prev", () =>
    $.fn.mauGallery.methods.prevImage(options.lightboxId)
);
$(".gallery").on("click", ".mg-next", () =>
    $.fn.mauGallery.methods.nextImage(options.lightboxId)
);
```

Les événements de clic sur les boutons "Précédent" et "Suivant" sont écoutés et appellent respectivement les méthodes `prevImage` et `nextImage`, qui gèrent le calcul de l'index de l'image courante et la mise à jour de l'image affichée.

- Mise à jour dynamique de la modale :

Une fonction dédiée met à jour l'image affichée dans la modale en fonction de l'index actuel. Elle modifie la source de l'image chaque fois qu'un utilisateur clique sur un bouton de navigation.

```
$(".lightboxImage").attr("src", nextImage);
```

Lorsqu'une image est changée, la source de l'image dans la modale (.lightboxImage) est mise à jour en fonction de l'index actuel, en utilisant l'image suivante ou précédente de la collection d'images.

- Gestion des débordements :

Lorsqu'un utilisateur navigue à partir de la première ou de la dernière image, un système de "cycle infini" est mis en place pour que la navigation continue en boucle, revenant à la première image après la dernière, et vice versa.

```
let nextImage = imagesCollection[(index - 1 + imagesCollection.length) % imagesCollection.length];
```

```
let nextImage = imagesCollection[(index + 1) % imagesCollection.length];
```

La logique `(index - 1 + imagesCollection.length) % imagesCollection.length` permet de revenir à la dernière image si l'utilisateur est sur la première. De même, la logique `(index + 1) % imagesCollection.length` assure que lorsque l'utilisateur atteint la dernière image, il sera renvoyé à la première.

- Optimisation et accessibilité :

Les événements de navigation au clavier (flèches gauche et droite) sont ajoutés pour permettre à l'utilisateur de naviguer entre les images sans avoir à cliquer sur les boutons.

```
$(document).on("keydown", function(e) {  
  if (e.key === "ArrowLeft") {  
    $.fn.mauGallery.methods.prevImage(options.lightboxId);  
  } else if (e.key === "ArrowRight") {  
    $.fn.mauGallery.methods.nextImage(options.lightboxId);  
  }  
});
```

L'événement `keydown` est écouté au niveau du document pour détecter quand les touches fléchées gauche (`ArrowLeft`) ou droite (`ArrowRight`) sont pressées. Lorsque l'une de ces touches est enfoncée, les méthodes de navigation sont appelées pour changer l'image.

- **Résultat :**

- L'index global assure que la navigation est toujours cohérente, avec des images mises à jour dynamiquement à chaque action de l'utilisateur.
- Le cycle infini garantit que l'utilisateur peut naviguer sans rencontrer de restrictions à l'extrémité de la galerie.
- Les événements de navigation sont bien gérés, offrant une expérience de navigation fluide grâce aux boutons de navigation et aux touches clavier.
- La gestion des événements a été optimisée pour une expérience plus rapide et réactive, tout en évitant les erreurs liées à la mise à jour de l'index des images.

3. Bug des filtres (affichage correcte des filtres lors de la sélection)

- **Problème initial :**

- Les filtres sélectionnés ne s'affichaient pas correctement, pas d'effet visuel visible par l'utilisateur lors de la sélection d'une catégorie.

- **Constats techniques :**

- Gestion des classes actives :

Le code original utilise la classe `active-tag` pour marquer le filtre sélectionné.

Cependant, la logique de gestion des états dans la fonction `filterByTag()` présentait une incohérence :

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

Les classes `active` et `active-tag` étaient bien appliquées ou supprimées, mais il manquait une synchronisation avec le rendu visuel.

- Effet visuel manquant :

Bien que la classe soit ajoutée, aucun style visuel spécifique n'était défini dans le CSS pour la classe `active-tag`. Cela entraînait une absence de retour visuel.

- Propagation correcte de l'état actif :

La mise à jour de l'état actif n'était pas propagée à l'ensemble des éléments liés au filtre (images et leurs conteneurs). Cela causait des incohérences dans l'affichage des items filtrés.

- Performance :

L'utilisation de `.each()` sur des éléments filtrés pour appliquer l'affichage ou le masquage peut devenir coûteuse en termes de performance si le nombre d'éléments est important.

- **Actions entreprises :**

- Gestion améliorée de la classe active :

Dans le nouveau code, la méthode `filterByTag()` a été modifiée pour inclure une gestion plus robuste de l'état actif :

```
$(".nav-pills .nav-link").removeClass("active-tag");  
$(this).addClass("active-tag");
```

Cela garantit que seuls les filtres pertinents sont marqués comme actifs et que la classe est correctement appliquée.

- Propagation des filtres :

Une fois un filtre sélectionné, les images correspondantes sont affichées avec une animation fluide :

```
$(".gallery-item").each(function () {  
    $(this).parents(".item-column").hide();  
  
    if (tag === "all") {  
        $(this).parents(".item-column").show(300);  
    } else if ($(this).data("gallery-tag") === tag) {  
        $(this).parents(".item-column").show(300);  
    }  
});
```

Les images sont filtrées dynamiquement et le rendu visuel est instantanément mis à jour.

- Ajout d'effets visuels (CSS) :

Bien qu'il ne soit pas visible dans le code fourni, il est essentiel de s'assurer qu'un style distinctif est défini pour la classe `active-tag` dans le CSS :

```
.nav-pills .nav-link.active-tag {  
    color: white !important;  
    background-color: #BEB45A !important;  
}
```

- **Résultat :**

- Les filtres sélectionnés s'affichent désormais avec un retour visuel clair et précis.
- La transition entre filtres est fluide grâce aux animations et à une meilleure gestion de l'état actif.
- L'utilisateur peut facilement identifier le filtre sélectionné.
- Le passage d'un filtre à l'autre est cohérent et sans retard perceptible.

4. Minimisation des fichiers CSS et JS :

- **Problème initial :**

- Présence de fichiers CSS et JS volumineux contenant des ressources inutilisées, ce qui ralentit le chargement de la page.
- Liens et scripts non optimisés dans le fichier `index.html`, augmentant le temps de rendu initial.

- **Actions entreprises :**

- Utilisation de PurgeCSS :

Analyse du fichier CSS pour détecter et supprimer les styles inutilisés, réduisant ainsi sa taille. Par exemple, seuls les styles réellement utilisés dans le projet ont été conservés.

```
$ purgecss --css **/*.css --content **/*.html **/*.js -o cleanAssets
```

- Minification avec CleanCSS :

Le fichier `style.css` a été minimisé à l'aide de CleanCSS, éliminant les espaces, commentaires et autres éléments inutiles, réduisant significativement son poids.

```
$ cleancss -o ./assets/style.min.css ./assets/style.css
```

- Remplacement des liens dans `index.html` :

Les anciens liens vers des fichiers CSS volumineux ont été remplacés par la version optimisée et minifiée du fichier `style.css`.

```
<link rel="stylesheet" href="./assets/bootstrap/bootstrap.min.css">
<link rel="stylesheet" href="./assets/style.min.css">
```

- Utilisation de Terser :

Tous les fichiers JavaScript restants ont été minimisés grâce à Terser, supprimant les caractères superflus et compressant le code.

```
$ for file in ./assets/*.js; do terser "$file" -o "${file%.js}.min.js"; done
```

- Mise à jour des scripts dans `index.html` :

Les scripts originaux ont été remplacés par leurs équivalents minimisés, garantissant une charge rapide et efficace.

```
<script src="./assets/maugallery.min.js"></script>
<script src="./assets/scripts.min.js"></script>
```

- **Résultat :**

- Les fichiers CSS et JS ont été considérablement réduits, ce qui diminue le temps de chargement des ressources critiques.
- Le fichier `index.html` est désormais optimisé pour charger uniquement les ressources nécessaires, améliorant ainsi les performances globales.
- La réduction des ressources bloquantes améliore le score de performance des outils comme Google PageSpeed Insights, ce qui est bénéfique pour la SEO et l'expérience utilisateur.

5. Ajout des balises meta pour les réseaux sociaux :

- **Problème initial :**

- Les pages du site ne contenaient pas de balises meta spécifiques pour les réseaux sociaux, limitant la qualité de leur affichage lors du partage sur des plateformes comme Facebook et Twitter.
- L'absence des balises OpenGraph et Twitter Cards entraînait une présentation non optimisée des liens, ce qui pouvait réduire le taux de clics et l'engagement des utilisateurs.

- **Actions entreprises :**

- Ajout des balises OpenGraph :

Des balises OpenGraph ont été intégrées dans la section `<head>` de chaque page importante (page d'accueil, pages produits, articles de blog).

- Ajout des balises Twitter Cards :

Les Twitter Cards ont été ajoutées pour optimiser l'affichage sur Twitter.

```
<meta property="og:title" content="Nina Carducci - Photographie">
<meta property="og:description" content="Nina Carducci est une photographe passionnée spécialisée dans les
<meta property="og:image" content="https://vinny031.github.io/OPENCLASSROOMS_P4/assets/images/nina.webp">
<meta property="og:url" content="https://vinny031.github.io/OPENCLASSROOMS_P4/">
<meta property="og:type" content="website">
<meta name="twitter:card" content="summary_large_image">
<meta name="twitter:title" content="Nina Carducci - Photographie">
<meta name="twitter:description" content="Nina Carducci est une photographe passionnée spécialisée dans le
<meta name="twitter:image" content="https://vinny031.github.io/OPENCLASSROOMS_P4/assets/images/nina.webp">
```

- **Résultat :**

- Les pages du site apparaissent désormais avec un aperçu optimisé lors du partage sur les réseaux sociaux, ce qui améliore la présentation et attire davantage d'utilisateurs.
- L'ajout des balises meta améliore le taux de clics en rendant les liens plus attrayants visuellement.
- L'expérience utilisateur et l'engagement sur les réseaux sociaux sont renforcés grâce à des descriptions claires et des images adaptées.
- La mise en conformité avec les bonnes pratiques des réseaux sociaux a un impact positif sur la visibilité et la perception du site.

6. Implémentation des Rich Snippets :

- **Problème initial :**

- Le site ne comportait pas de balises spécifiques pour les Rich Snippets, limitant la visibilité des informations importantes dans les résultats de recherche.
- L'absence de données structurées empêchait l'affichage d'informations enrichies (comme l'adresse, le numéro de téléphone ou les horaires) dans les résultats Google, réduisant ainsi l'efficacité du référencement local.

- **Actions entreprises :**

- Des balises JSON-LD ont été intégrées dans la section <head> pour décrire l'entreprise sous le format LocalBusiness.
- Les données ajoutées incluent le nom de l'entreprise, l'adresse, le numéro de téléphone, les horaires d'ouverture et les coordonnées géographiques.
- Optimisation pour le référencement local : Les informations spécifiques comme le code postal, la ville et les heures de disponibilité ont été détaillées pour améliorer la pertinence des résultats de recherche.
- Test et validation des données : Les microdonnées ont été vérifiées avec l'outil Rich Results Test pour garantir leur conformité avec les exigences de Google.

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "LocalBusiness",
  "name": "Nina Carducci - Photographie",
  "address": {
    "@type": "PostalAddress",
    "streetAddress": "68 avenue Alsace-Lorraine",
    "addressLocality": "Bordeaux",
    "postalCode": "33200",
    "addressCountry": "FR"
  },
  "telephone": "+33 5 56 67 78 89",
  "openingHours": "Mo-Fr 10:00-19:00",
  "url": "https://vinny031.github.io/OPENCLASSROOMS_P4/",
  "image": "https://vinny031.github.io/OPENCLASSROOMS_P4/assets/images/nina.webp",
  "priceRange": "€€",
  "geo": {
    "@type": "GeoCoordinates",
    "latitude": 44.8378,
    "longitude": -0.5792
  }
}
</script>
```

- **Résultat :**

- Les pages du site affichent désormais des informations enrichies dans les résultats de recherche, telles que l'adresse et le numéro de téléphone de l'entreprise.
- L'ajout des données structurées améliore la visibilité dans les recherches locales, augmentant les chances d'attirer des clients proches géographiquement.
- Les Rich Snippets renforcent la crédibilité et l'attractivité du site dans les résultats Google, tout en améliorant l'expérience utilisateur.
- La mise en œuvre des bonnes pratiques SEO pour les données structurées contribue à un meilleur classement dans les moteurs de recherche.

Résultats du test PARTAGER

 **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

Détails

Exploration

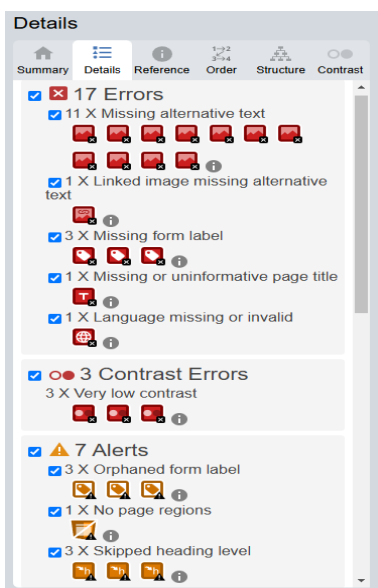
 Exploration effectuée le 28 janv. 2025, 15:51:39

Données structurées détectées

 Commerces et services à proximité	1 élément valide détecté	>
 Organisation	1 élément valide détecté	>

III. Accessibilité du site

1. Accessibilité avant optimisation



- **Erreurs :**

- Texte alternatif manquant (11 erreurs) :

Plusieurs images sur la page n'ont pas de texte alternatif (alt), ce qui est essentiel pour l'accessibilité, notamment pour les utilisateurs de lecteurs d'écran. L'absence de ce texte empêche une partie des utilisateurs de comprendre le contenu visuel de ces images.

- Image liée sans texte alternatif (1 erreur) :

Une image cliquable (avec un lien) manque de texte alternatif. Cela nuit à l'accessibilité de la page, car les utilisateurs de lecteurs d'écran ne peuvent pas savoir ce que l'image représente ni où elle mène.

- Labels de formulaire manquants (3 erreurs) :

Des champs de formulaire ne sont pas associés à des labels. Les labels permettent aux utilisateurs de savoir à quoi chaque champ correspond, ce qui est crucial pour l'accessibilité.

- Titre de page manquant ou informatif (1 erreur) :

Le titre de la page est soit manquant, soit insuffisamment descriptif. Un titre pertinent et unique est essentiel pour l'indexation par les moteurs de recherche et pour aider les utilisateurs à comprendre rapidement de quoi parle la page.

- Langue manquante ou invalide (1 erreur) :

La langue de la page n'a pas été définie, ce qui peut poser des problèmes d'accessibilité pour les utilisateurs de lecteurs d'écran et affecter la SEO (optimisation pour les moteurs de recherche).

- Contraste insuffisant (3 erreurs) :

Il y a des problèmes de contraste de couleur entre le texte et l'arrière-plan, ce qui rend le contenu difficile à lire pour les utilisateurs ayant des déficiences visuelles.

- **Alertes :**

- Labels de formulaire orphelins (3 alertes) :

Des labels de formulaire sont présents, mais ils ne sont pas correctement associés à leurs champs respectifs. Cela empêche les utilisateurs de comprendre clairement la fonction de chaque champ dans le formulaire.

- Absence de régions de la page (1 alerte) :

Aucun rôle ou attribut ARIA n'a été utilisé pour définir des régions principales de la page (par exemple, "navigation", "contenu principal"), ce qui peut rendre la navigation difficile, surtout pour les utilisateurs de technologies d'assistance.

- Niveaux de titres sautés (3 alertes) :

La hiérarchie des titres (balises <h1>, <h2>, etc.) n'est pas respectée, ce qui peut nuire à la structure logique de la page et à la navigation. Par exemple, un <h3> est utilisé avant un <h2>, ce qui perturbe la navigation sémantique.

- **Fonctionnalités :**

- Texte alternatif (4 erreurs) :

La présence de texte alternatif est de nouveau un point clé manquant. Les éléments visuels ne sont pas suffisamment décrits pour être accessibles aux utilisateurs de technologies d'assistance.

- Images sans description (2 erreurs) :

- Certaines images n'ont pas de descriptions associées, rendant leur contenu inaccessible pour les utilisateurs malvoyants.

- **Éléments structuraux :**

- Problèmes avec les niveaux de titres :

Le rapport montre une utilisation incohérente des niveaux de titres (par exemple, plusieurs <h3> sans passer par un <h2>, ou un <h6> après un <h3>). La structure des titres est importante pour assurer une navigation logique et fluide, particulièrement pour les utilisateurs de technologies d'assistance.

- Listes non ordonnées (2 erreurs) :

Des listes non ordonnées sont mal structurées ou absentes dans des sections où elles seraient appropriées. Cela peut affecter la compréhension du contenu.

- **ARIA (Accessibilité Rich Internet Applications) :**

- Problèmes ARIA :

L'utilisation d'attributs ARIA (comme aria-hidden, aria-label, aria-tabindex, etc.) semble incorrecte ou manquante. L'attribut aria-hidden est utilisé dans certains cas où des éléments sont peut-être cachés à la vue mais doivent être accessibles pour certaines interactions.

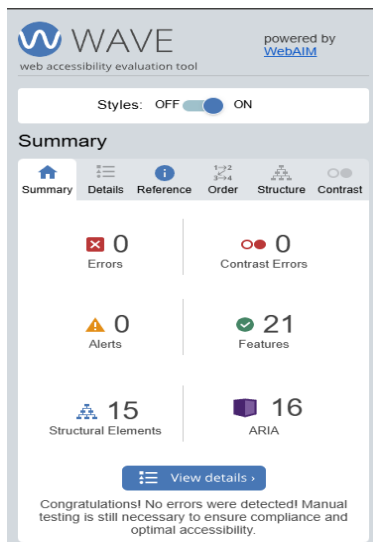
- Problèmes avec aria-label (3 erreurs) :

Certaines balises ou éléments interactifs nécessitent des labels ARIA explicites, qui sont soit manquants soit mal appliqués. Ces labels sont cruciaux pour fournir des descriptions aux utilisateurs de lecteurs d'écran.

- Problèmes avec aria-tabindex (1 erreur) :

Le tabindex est peut-être mal utilisé, ce qui pourrait perturber la navigation au clavier sur la page.

2. Accessibilité après optimisation



3. Correction des erreurs révélées par le validateur W3C

- **Problème initial :**

- Absence de l'attribut `lang="fr"` dans la balise `<html>`. Ca empêche les moteurs de recherche et les lecteurs d'écran de détecter la langue du contenu, ce qui peut nuire à l'accessibilité et à la SEO.
- Absence de la balise `<title>` dans la section `<head>`. La balise `<title>` est essentielle pour l'identification et la description de la page, en particulier dans les résultats des moteurs de recherche et dans l'onglet du navigateur.
- Images sans attribut `alt`. L'attribut `alt` est crucial pour l'accessibilité, notamment pour les utilisateurs de lecteurs d'écran, car il fournit une description textuelle des images.
- Utilisation inappropriée du slash terminal sur les balises auto-fermantes (ex. ``). Bien que techniquement valide en XHTML, l'utilisation du slash terminal est inutile en HTML5, ce qui peut créer des incohérences.
- Champ action vide dans un formulaire. L'attribut `action` d'un formulaire doit être correctement renseigné pour que le formulaire soumette les données vers une URL cible.

- **Actions entreprises :**

- L'attribut `lang="fr"` a été ajouté dans la balise `<html>`, ce qui permet aux navigateurs et autres outils d'identifier que la langue du contenu est le français.

```
<html lang="fr">
```

- Un titre pertinent a été ajouté dans la balise `<head>`, ce qui améliore la SEO et offre une description de la page dans l'onglet du navigateur.

```
<title>Nina Carducci - Photographe</title>
```

- Chaque balise `` a été enrichie d'un attribut `alt` approprié, afin de garantir l'accessibilité de l'image pour les utilisateurs qui ne peuvent pas la voir (ex : utilisateurs de lecteurs d'écran).

```

```

- Les slashes de fermeture (`/`) ont été retirés des balises auto-fermantes comme ``, car ils ne sont pas nécessaires en HTML5, améliorant ainsi la lisibilité et la conformité.

```
<input type="submit" value="Envoyer"/>
```

```
<input type="submit" value="Envoyer">
```

- L'attribut `action` du formulaire a été correctement renseigné avec l'URL de destination, permettant ainsi la soumission correcte des données.

```
<form action="/contact" method="post">
```

- **Résultat :**

- Après les corrections, le code HTML respecte désormais les normes du W3C, garantissant ainsi une meilleure compatibilité avec les navigateurs modernes.
- L'ajout des attributs `lang` et `alt` améliore l'accessibilité du site, permettant une meilleure expérience utilisateur, notamment pour les personnes en situation de handicap.
- L'inclusion du titre et la conformité générale du code contribuent à une meilleure indexation par les moteurs de recherche, augmentant ainsi la visibilité du site.
- La suppression des erreurs liées aux balises auto-fermantes et au champ `action` vide assure une meilleure prévisibilité et stabilité du rendu de la page sur tous les navigateurs.

4. Ajustement des contrastes dans la navlink :

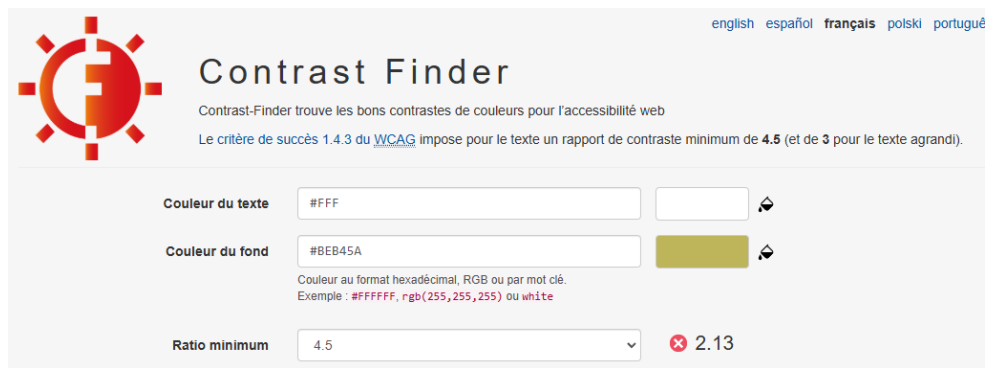
- **Problème initial :**

- Les couleurs d'arrière-plan et de premier plan pour le lien de navigation sélectionné (navlink) ne présentaient pas un contraste suffisant.
- Avec un ratio initial de **2,13**, la lisibilité était compromise, en particulier pour les utilisateurs souffrant de troubles visuels ou dans des conditions d'éclairage défavorables.

- **Actions entreprises :**

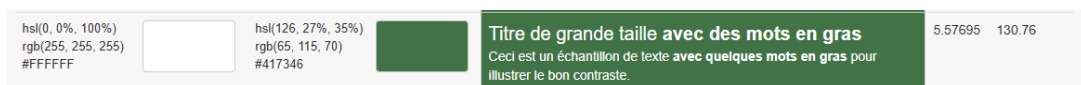
- Analyse des couleurs :

Les couleurs existantes ont été évaluées à l'aide d'outils de vérification de contraste, ici Contrast Finder.



- Modification des couleurs :

Une nouvelle palette a été sélectionnée pour améliorer le contraste : couleur d'arrière-plan modifiée pour atteindre un ratio de **5,57** (conforme aux standards WCAG AA pour les textes normaux).



- Mise à jour des styles CSS :

Les modifications ont été appliquées au fichier `style.css`, ciblant spécifiquement la classe liée à la navlink sélectionnée.

- **Résultat :**

- La navlink sélectionnée est désormais parfaitement lisible, même pour les utilisateurs ayant des besoins spécifiques en matière d'accessibilité.
- Le ratio de contraste amélioré à **5,57** respecte les recommandations WCAG, renforçant ainsi l'accessibilité et la qualité globale de l'expérience utilisateur.

5. Correction des éléments d'en-tête non classés séquentiellement :

- **Problème initial :**

- La structure des en-têtes (H1, H2, H3, etc.) ne respectait pas une hiérarchie logique, compromettant la sémantique et l'accessibilité de la page.
- Exemples relevés :
 - Un en-tête **H1** utilisé pour une citation, ce qui n'est pas conforme à son rôle principal de titre principal de la page.
 - Des en-têtes **H3** appliqués à des sections entières, au lieu de titres de niveau supérieur appropriés.
 - Des sauts incohérents dans la hiérarchie, comme un **H6** placé directement après un **H3**.

- **Actions entreprises :**

- Révision de la hiérarchie des en-têtes :
 - L'en-tête **H1** a été déplacé pour correspondre au titre principal de la page (nom du site).
 - Le titre de la section *About Me* a été changé de **H3** en **H2**, pour mieux refléter sa position dans la hiérarchie.
 - Les en-têtes **H6** de la section *About Me* ont été modifiés en **H3**, assurant une progression logique.
 - Les titres des sections *Portfolio*, *Mes Services*, et *Contact* ont été ajustés de **H3** à **H2** pour correspondre à leur importance dans la structure globale.
 - La citation en fin de page a été changée de **H1** à une balise **<p>**, ce qui reflète mieux son rôle non structurel.

- **Résultat :**

- La structure des en-têtes est désormais cohérente, garantissant une meilleure sémantique HTML et une navigation facilitée pour les technologies d'assistance.
- L'organisation hiérarchique correcte améliore également le SEO, en fournissant aux moteurs de recherche une structure claire et logique.
- La page respecte les standards de l'accessibilité, offrant une expérience utilisateur optimisée pour tous les visiteurs.

6. Texte visuellement masqué dans les boutons de contrôle du carrousel

:

- **Problème initial :**

- La balise `Image précédente dans le carrousel` générait une alerte WAVE concernant l'accessibilité. Cette balise était utilisée pour fournir une description textuelle pour les utilisateurs de technologies d'assistance, mais elle pouvait être perçue comme redondante ou incorrecte par l'outil de validation.

```
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
  <span class="visually-hidden">Image suivante dans le carrousel</span>
</button>
```

- **Actions entreprises :**

- Le texte a été supprimé de la span visuellement masquée et remplacé par un attribut `aria-label` directement dans le bouton de navigation. Cela permet d'améliorer l'accessibilité sans perturber l'interface utilisateur visuelle, tout en garantissant que les informations sont toujours accessibles via des technologies d'assistance.

```
<button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next" aria-label="Image suivante dans le carrousel">
  <span class="carousel-control-next-icon" aria-hidden="true"></span>
</button>
```

- **Résultat :**

- L'ajout de l'attribut `aria-label` dans le bouton de contrôle du carrousel a permis de résoudre le problème signalé par WAVE concernant la balise span visuellement masquée. Cette modification garantit une description claire et précise des actions du carrousel pour les utilisateurs de technologies d'assistance, sans altérer l'interface visuelle du site.
- Le remplacement du texte visuellement masqué par un `aria-label` assure une meilleure conformité avec les exigences d'accessibilité, notamment pour les utilisateurs de lecteurs d'écran. Les alertes WAVE liées à la balise span ont été supprimées, ce qui renforce la qualité d'accessibilité du site.
- La mise en place de l'attribut `aria-label` améliore l'interaction des utilisateurs avec les éléments de contrôle du carrousel, rendant l'interface plus inclusive. Les utilisateurs de technologies d'assistance bénéficient désormais d'une meilleure description des actions disponibles, contribuant ainsi à une navigation plus fluide et agréable.

7. Ajout d'un attribut for dans les labels du formulaire :

- **Problème initial :**

- Certains éléments de formulaire manquaient d'un attribut for sur leurs labels, ce qui compliquait la navigation pour les utilisateurs de technologies d'assistance. L'attribut for permet d'associer explicitement un label à un champ de formulaire, facilitant ainsi la navigation pour les utilisateurs de lecteurs d'écran et améliorant l'interaction avec les formulaires.

```
<form action method=  
  <label>Nom</label>  
  <input type="text"
```

- **Actions entreprises :**

- L'attribut for a été ajouté à chaque label manquant, afin de lier correctement chaque champ de formulaire avec son texte explicatif. Cela garantit que les lecteurs d'écran peuvent annoncer le champ de manière claire et précise, et que les utilisateurs peuvent facilement interagir avec les éléments du formulaire.

```
<form action="/contact" method="po  
  <label for="nom">Nom</label>  
  <input type="text" name="nom"
```

- **Résultat :**

- L'ajout de l'attribut for aux labels a permis d'améliorer l'accessibilité des formulaires. Chaque champ de formulaire est maintenant correctement associé à son label, ce qui facilite la navigation et l'interaction avec les éléments du formulaire pour les utilisateurs de technologies d'assistance.
- La mise à jour assure une meilleure conformité avec les directives WCAG concernant l'association des labels avec leurs champs respectifs. Cette correction renforce l'accessibilité du site, rendant l'expérience plus fluide et compréhensible pour les utilisateurs de lecteurs d'écran.
- Les utilisateurs ayant des besoins spécifiques bénéficient désormais d'une navigation plus intuitive. L'ajout de l'attribut for rend les formulaires plus accessibles, contribuant ainsi à une meilleure interaction avec le site.

8. Remplacement de `<div class="header">` par `<header>` :

- **Problème initial :**

Le site utilisait une `<div id="header">` pour structurer l'en-tête de la page. Or, cet élément générique ne véhicule aucune signification sémantique, ce qui compliquait la navigation et la compréhension de la structure du document pour les utilisateurs de technologies d'assistance.

```
<div id="header">
```

- **Action entreprise :**

La `<div class="header">` a été remplacée par une balise `<header>`, un élément HTML sémantique spécifiquement conçu pour représenter un en-tête de page ou de section.

Cette modification permet aux lecteurs d'écran et autres outils d'accessibilité d'identifier plus clairement la structure de la page.

```
<header>
```

- **Résultat :**

- L'utilisation de `<header>` améliore l'accessibilité en fournissant une structure sémantique explicite, facilitant la compréhension et la navigation pour les utilisateurs de technologies d'assistance.
- Cette mise à jour améliore également le SEO, car les moteurs de recherche privilégient les documents HTML bien structurés.
- La conformité avec les bonnes pratiques du HTML et les recommandations WCAG est renforcée, rendant le site plus accessible et ergonomique pour tous les utilisateurs.

9. Ajout d'une balise méta description :

- **Problème initial :**

- La balise `<meta name="description">` était absente dans le fichier HTML, ce qui empêchait les moteurs de recherche d'afficher une description claire et optimisée dans leurs résultats.
- Cela pouvait également nuire au taux de clic (CTR) en ne fournissant pas d'aperçu pertinent du contenu de la page aux utilisateurs.

- **Actions entreprises :**

- Ajout de la balise méta description :

Une balise <meta> a été ajoutée dans la section <head> avec une description concise et pertinente de la page, en respectant les bonnes pratiques :

- Texte limité à environ **150-160 caractères** pour éviter une troncature dans les résultats des moteurs de recherche.
 - Utilisation de mots-clés ciblés pour améliorer la pertinence SEO sans surcharger la description.

```
<meta name="description" content="Nina Carducci est une photographe passionnée spécialisée dans les portraits,"
```

- **Résultat :**

- La balise méta description optimise la visibilité du site dans les résultats de recherche, en offrant un résumé clair et engageant.
 - Une description bien rédigée améliore le taux de clics en attirant les utilisateurs intéressés par le contenu de la page.
 - La conformité avec les pratiques recommandées renforce la qualité globale du code et la stratégie SEO du site.

IV. Cahier de recette

ID	Action	Résultat initial	Résultat attendu	Statut	Commentaires
1	Optimisation des images	Images non optimisées	Gain \geq 43%	Résolu	Utilisation de cwebp pour redimensionner et compresser les images tout en les convertissant en WebP.
2	Switch des photos dans modale	Non fonctionnel	Fonctionnel	Résolu	Navigation fluide entre les images avec mise à jour de l'index et gestion du cycle infini.
3	Affichage des filtres	Non fonctionnel	Fonctionnel	Résolu	Système de filtrage optimisé avec retour visuel et transitions fluides.
4	Minimisation des fichiers CSS/JS	Non optimisé	Réduction taille fichiers	Résolu	Utilisation de PurgeCSS et CleanCSS pour CSS, Terser pour JS.
5	Rich Snippets	Absent	Présent et validé	Résolu	Implémentation des données structurées pour améliorer la visibilité dans les résultats de recherche.
6	Méta-balises réseaux sociaux	Absentes	Présentes	Résolu	Ajout des balises OpenGraph et Twitter Cards.
7	Accessibilité labels formulaire	Non conforme	Conforme WCAG	Résolu	Ajout des attributs for sur les labels et correction des contrastes.
8	Structure HTML sémantique	Non conforme	Conforme	Résolu	Remplacement des div par des balises sémantiques appropriées.
9	Hierarchie des titres	Non structurée	Structurée	Résolu	Correction de la hiérarchie des titres H1-H6.

10	Meta description	Absente	Présente	Résolu	Ajout d'une meta description optimisée pour le SEO.
11	Contrôles du carrousel	Non accessible	Accessible	Résolu	Remplacement du texte masqué par des attributs aria-label.
12	Langue du site	Non définie	Définie	Résolu	Ajout de l'attribut lang="fr" dans la balise HTML.
13	Balise title	Absente	Présente	Résolu	Ajout de la balise title dans le head.
14	Formulaire action	Vide	Renseigné	Résolu	Correction de l'attribut action du formulaire.

V. **Annexe**

1. **Rapports complets des audits Lighthouse**

Cf : Rapport_lighthouse_pré_opti.pdf

Cf : Rapport_lighthouse_post_opti.pdf