



Rapport Sprint 4 & 3 :

Front

Study & Read

Création d'une page web responsive

Réalisé par :

Sara Akharraz

Entreprise :

Void

Chapitre 1 :

Objectifs :

- Apprendre HTML.
- Apprendre CSS & Tailwind CSS
- Apprendre JavaScript

En me basant sur la documentation fournie sur le site Void Training, j'ai appris pas mal de choses concernant HTML : les bonnes pratiques, l'accessibilité, le SEO, la sémantique... Il en est de même pour Tailwind CSS et JavaScript. **Chapitre 2 :**

Création d'une page web responsive complète utilisant **HTML sémantique**, **Tailwind CSS** et **JavaScript natif**.

Objectifs :

- Maîtriser la sémantique HTML .
- Utiliser Tailwind CSS pour le styling et la responsivité
- Implémenter le dark mode avec Tailwind CSS
- Créer des images responsive avec **picture** et **srcset**
- Intégrer et styler des **icons SVG**
- Créer des animations CSS et JavaScript
- Utiliser l'**aspect ratio**
- Développer un header avec **mega menu** , **burger menu**
- Implémenter la version **RTL (arabe)** de la page
- Gérer le **preloading** et le **lazy loading** des médias
- Intégrer des custom fonts via Google Fonts
- Intégrer une vidéo embed responsive
- Créer un formulaire fonctionnel avec **validation JavaScript**

Bonus :

- Lighthouse
- Accessibilité
- Tester sur différent navigateur avec browser stack

Livrable :

Le lien du site web

le lien du dépôt github

La sémantique HTML :

En me basant sur les bonnes pratiques apprises lors du sprint 3, une structure HTML sémantique a été mise en considération. La structure du projet repose sur une architecture HTML sémantique testée en premier lieu sur le site : **validator.w3.org**"

Réponsivité :

Pour la responsivité, j'ai utilisé les points d'arrêt (breakpoints) des utilitaires Tailwind (sm, md, lg, xl, 2xl) :

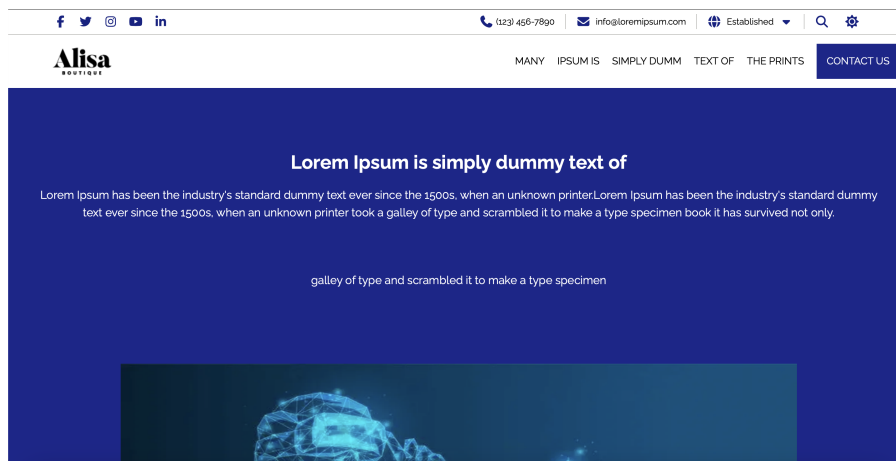


FIGURE 1 – desktop

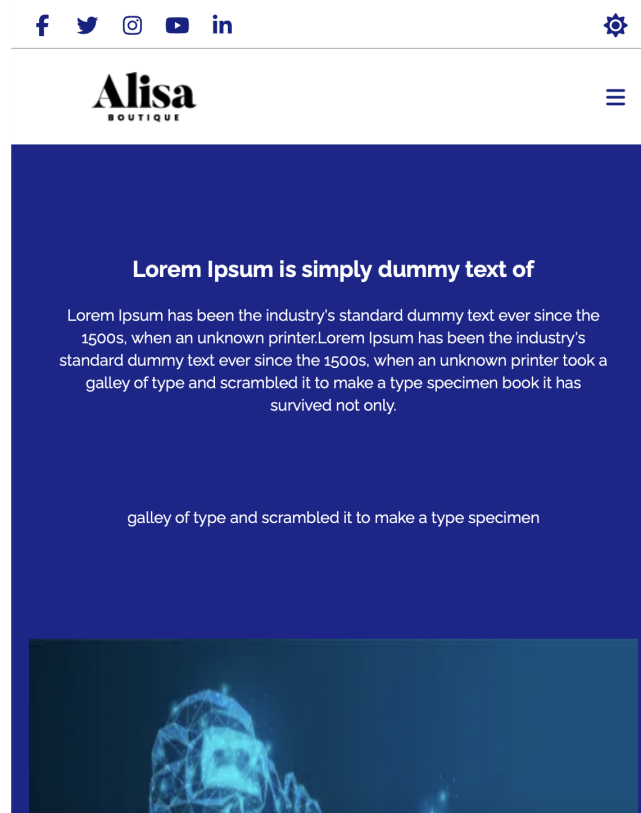


FIGURE 2 – mobile

Dark mode :

J'ai implémenté en utilisant java script et les utilitaires tailwind css :

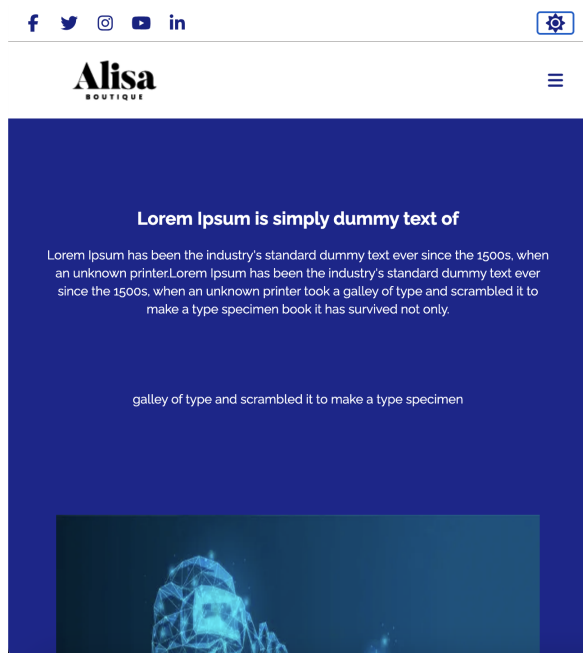


FIGURE 3 – light mode

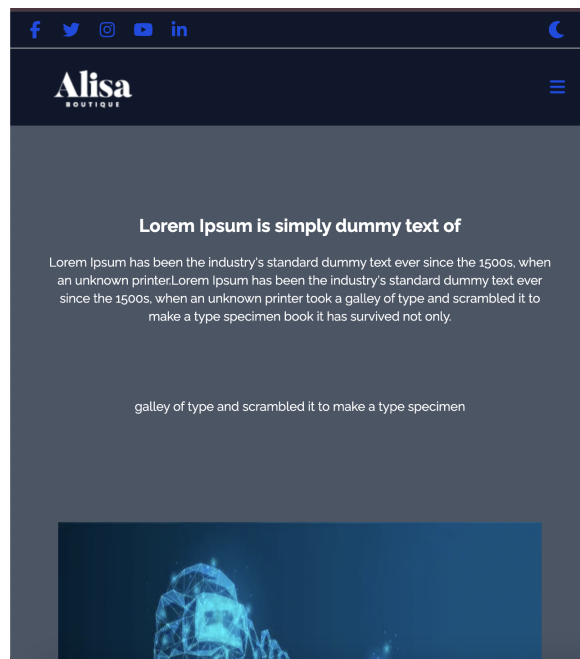


FIGURE 4 – dark mode

L'optimisation des images :

picture et srcset

Pour optimiser les images en fonction de la taille de l'écran et des performances, j'ai utilisé les éléments `<picture>` et l'attribut `srcset`.

`<picture>` : Élément HTML qui permet de proposer plusieurs versions d'une image selon différents critères (taille d'écran, format, résolution).

`srcset` : Attribut qui définit une liste d'images avec leurs dimensions respectives, permettant au navigateur de choisir la plus appropriée.

```
<div class="relative w-full flex justify-center ">
  <picture class="absolute -top-40 z-0 w-full sm:w-10/12 sm:h-11/12 lg:w-3/4 flex justify-center px-4 sm:px-0">
    <!-- Mobile -->
    <source
      srcset="./images/hello-mobile.webp"
      media="(max-width: 640px)"
      type="image/webp"
    />
    <!-- Desktop -->
    <source
      srcset="./images/hello-desktop.webp"
      type="image/webp"
    />
    <!-- Fallback img -->
    
  </picture>
</div>
```

Formats WebP et AVIF

J'ai utilisé les formats WebP et AVIF :

- WebP et AVIF sont des formats d'image modernes qui offrent une meilleure compression que JPEG et PNG, réduisant le poids des fichiers tout en maintenant une excellente qualité.
- AVIF est encore plus performant que WebP.

J'ai utilisé Squoosh pour redimensionner et compresser les images : **squoosh.app**

Preloading et Lazy loading

Preloading : Technique qui charge les ressources importantes à l'avance pour améliorer les performances et réduire les délais d'affichage.

Lazy loading : Technique qui retarde le chargement des images jusqu'à ce qu'elles soient visibles à l'écran, réduisant ainsi le poids initial de la page.

Les icons SVG :

Pourquoi utiliser des icons SVG plutôt que Font Awesome ? :

Font Awesome est une librairie externe qu'il faut télécharger et charger dans le navigateur. Cela signifie :

- Un fichier supplémentaire à charger (augmente le poids de la page)
- Une requête HTTP supplémentaire
- Un délai d'affichage des icons
- Une dépendance envers un service externe

Avec les SVG, on crée directement les icons en HTML/CSS sans dépendre d'une librairie externe.

Les icons utilisés dans le header sont au format SVG :



FIGURE 5 – svg icons

Formulaire :validation JavaScript

Le formulaire a été validé avec JavaScript :

Name

Email Email is required

Phone Phone must be exactly 10 digits

Message Message must be at least 10 characters

L'accessibilité :

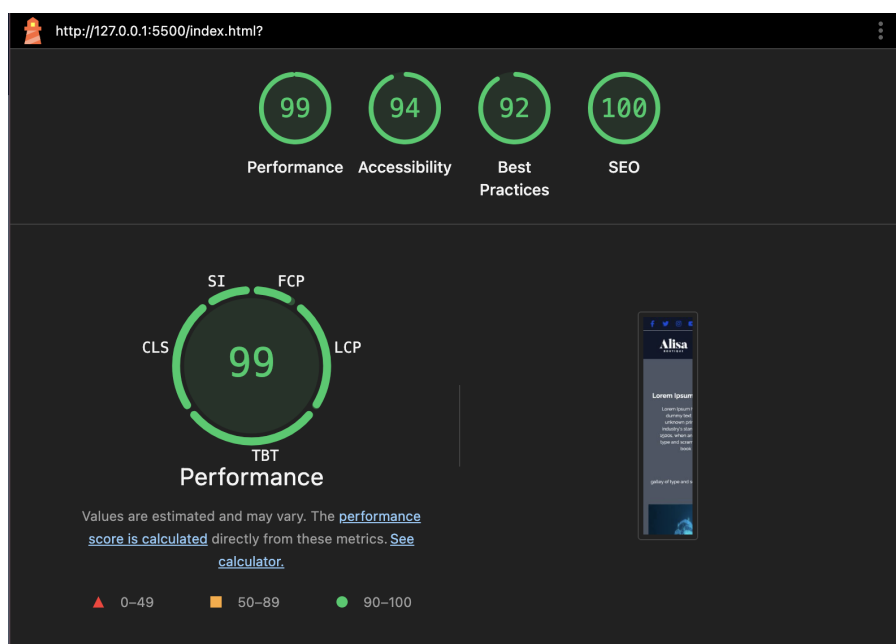
L'attribut `aria-label` est utilisé pour décrire les icons aux lecteurs d'écran, et la sémantique HTML est respectée pour assurer une meilleure accessibilité.

SEO :

Les métadonnées et la sémantique HTML sont utilisées pour améliorer le référencement naturel et la visibilité dans les résultats de recherche.

Lighthouse :

La page a été auditée avec Lighthouse.



METRICS

[Expand view](#)

● First Contentful Paint
1.7 s

● Largest Contentful Paint
1.7 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0.013

● Speed Index
1.7 s