

Étapes clés - Projet 6 du parcours Développeur Front-End

Créez un site accessible pour une plateforme de photographes

Besoin d'indications pour avancer dans votre projet ? Suivez les étapes clés proposées ci-dessous !

Recommandations générales

Dans ce projet, vous allez devoir coder une application accessible qui contient des données et plusieurs fonctionnalités assez conséquentes. Quand vous vous retrouvez confronté à ce genre de situation, le mieux est de tenter de découper l'application en une succession de tâches plus abordables à réaliser successivement. Ce document vise à vous simplifier le découpage pour ce projet.

Il sera plus simple de réaliser les tâches 1 à 4 dans l'ordre, en revanche, les tâches 5 à 9 peuvent totalement être réalisées dans le désordre.

Etape 1 : Prendre en main les éléments, la maquette et la base de code

5% d'avancement

Avant de commencer, il est essentiel de prendre connaissance des éléments mis à disposition, notamment la maquette et la base de code mise en place.

⌚ Une fois cette étape réalisée, vous aurez :

- une compréhension des maquettes
- une compréhension des livrables attendus
- une connaissance du fonctionnement / du découpage des pages HTML / JS et CSS.

📌 Recommandations :

- Affichez les 2 pages HTML fournies dans votre navigateur.

- Faites des console.log dans les différentes parties du JavaScript pour vous assurer que vous comprenez ce que le code fait.
- Modifiez une propriété CSS dans style.css pour vérifier que tout est bien connecté.

 **Points de vigilance :**

- Attention à vous assurer que vous comprenez bien tout le code présent. N'hésitez pas à le modifier pour vous l'approprier.

Etape 2 : Importer les datas

10% d'avancement

Maintenant que vous maîtrisez la base de code présente, vous allez devoir importer les vraies données de votre application.

 **Une fois cette étape réalisée, vous aurez :**

- Votre application qui récupère les données provenant du fichier JSON.

 **Recommandations :**

- Vous aurez votre fichier JSON dans datas. Vous devrez également mettre les images associées dans assets.

Dans scripts/pages/index.js vous allez :

- 1- Ajouter fetch dans la fonction getPhotographers pour récupérer vos datas, et faire un console.log de ces datas
- 2- Retourner les datas
- 3- Modifier `scripts/templates/photographer.js` pour récupérer les données nécessaires (id, tagline, city, etc.)

- Ici, vous pouvez utiliser l'API native de JavaScript Fetch afin de récupérer des données.

 **Ressources :**

- Le chapitre "[Envoyez une requête depuis le navigateur](#)" du cours "Créez des pages web dynamiques avec JavaScript" vous indiquera comment utiliser "fetch" pour faire vos appels API.

Etape 3 : Intégrer la page d'accueil

20% d'avancement

Maintenant que vous avez réussi à récupérer les données dans votre page d'accueil, vous allez pouvoir aller plus loin en intégrant votre page d'accueil.

⌚ Une fois cette étape réalisée, vous aurez :

- La page avec la liste des cards des photographes
- L'accessibilité de la page index.html

📌 Recommandations :

C'est le moment de créer le HTML, le CSS et le JS associés à la page d'accueil.

- Pour ce qui est du JS, vous trouverez un template qui a été fait pour les cartes des photographes dans scripts/factories/photographer.js. Vous devrez étendre ce template afin de générer tous les éléments nécessaires, et retourner les éléments du DOM associés.
- Les éléments du DOM ainsi que le HTML et CSS devront être écrits afin d'assurer l'accessibilité : vous devrez surtout être vigilant sur les déclarations alt et aria-label.
- Le début d'intégration CSS qui a été fait utilise flexbox et CSS grid, n'hésitez pas à jeter un œil à leur documentation.
- Pour le moment, pas la peine de vous embêter à créer les liens, vous le ferez à l'étape suivante.

📘 Ressources :

- [L'accessibilité avec aria-label et alt](#)
- [Layouts Flexbox et CSS Grids](#)

Etape 4 : Gérer la navigation entre la page accueil et la page photographe

35% d'avancement

Vous avez la base de votre page d'accueil. L'idée de cette étape est de permettre la navigation vers la page détaillée du photographe en créant les liens, et en gérant le chargement des données correspondant à la page sur laquelle vous vous trouvez.

⌚ Une fois cette étape réalisée, vous aurez :

- Votre page d'accueil finalisée
- Le rapport AChecker de votre page
- Le chargement des données correspondant à la page détaillée

📌 Recommandations :

Ici, vous devrez construire un système permettant de passer du lien cliqué au chargement de la page. Pour cela, vous appuierez sur l'id du photographe sur lequel l'utilisateur a cliqué et vous le passerez en paramètre de l'url affichée.

- Vous pouvez commencer par faire un console.log des données correspondant au photographe sélectionné depuis l'URL.
- Vous afficherez ensuite le contenu de la page à la prochaine étape.
- Attention à l'accessibilité de vos liens (aria-label, gérer le focus, etc.)
- Maintenant que vous avez finalisé votre page d'accueil, vous pouvez réaliser un rapport d'accessibilité avec un validateur d'accessibilité ou une checklist, et corriger votre code en fonction

📘 Ressources :

- [Documentation de searchParams](#)
- [Les outils pour valider l'accessibilité](#)

Etape 5 : Afficher le contenu statique de la page photographe

45% d'avancement

Le moment est venu de créer le contenu HTML et CSS de votre page photographe !

⌚ Une fois cette étape réalisée, vous aurez :

- Le contenu HTML et CSS de la page photographe

📍 Recommandations :

- Vous devrez réutiliser la fonction photographer Template que vous aviez étendu à l'étape 3 pour afficher le contenu de votre page, ainsi que votre fonction permettant d'utiliser fetch.
- Vous afficherez également les réalisations des photographes, en créant une factory pour Media.
- N'oubliez pas le petit encart qui affiche le tarif journalier du ou de la photographe affiché.
- Vous vous occuperez du nombre de likes, de la LightBox et du ContactForm dans une étape ultérieure.

⚠ Points de vigilance :

- Dans la factory Media, vous devrez gérer les différents cas où le média est une image ou une vidéo.

📘 Ressources :

- [Gérer l'accessibilité de vos médias](#)
- [Vidéo sur les factory functions](#)
- [Cet article sur les factory functions](#)

Etape 6 : Créer la modale de contact

55% d'avancement

Développez une modale de contact qui s'affiche correctement en cliquant sur "Contactez-moi".

⌚ Une fois cette étape réalisée, vous aurez :

- Une modale qui s'affiche lorsque l'on clique sur "Contactez-moi".

📌 Recommandations :

- La base de la modale est déjà présente dans la codebase. Vous devrez y ajouter la gestion du formulaire ainsi que le style manquant.
- Pas d'API complexe avec laquelle interagir ici pour envoyer le contenu de votre formulaire. Un simple console.log des données entrées par l'utilisateur suffira à montrer que vous maîtrisez votre sujet.

⚠ Points de vigilance :

- Attention : dans la codebase actuelle, rien n'a été fait pour gérer l'accessibilité de votre modale. Vous devrez gérer le focus sur toute la page, le role, l'aria-label, etc.

📘 Ressources :

- [Comment créer une modale accessible](#)

Etape 7 : Gérer les médias de la Lightbox

65% d'avancement

Vous allez pouvoir réutiliser la base de votre modale afin de créer la Lightbox.

⌚ Une fois cette étape réalisée, vous aurez :

- La LightBox qui s'affiche lorsque vous cliquez sur une photographie, et dans laquelle vous pouvez faire défiler les autres photographies. La LightBox se ferme au clic sur le bouton de fermeture.

📌 Recommandations :

- Vous devrez gérer les images et les vidéos affichées.

- Attention à bien gérer l'ouverture de la Lightbox, le défilement et la fermeture avec le clic souris, mais aussi avec le clavier. Pour cela, vous devrez utiliser des EventListeners et bien déclencher les actions correspondantes.

Points de vigilance :

- Comme pour les autres étapes, n'oubliez pas de bien déclarer les aria-label et le rôle de votre LightBox.

Ressources :

- Vous pouvez réutiliser [la ressource précédente](#) ici pour construire votre Lightbox.

Etape 8 : Afficher et gérer les likes

80% d'avancement

Au tour des likes. Dans cette étape vous allez incrémenter les likes sur les photos, et lister les likes de toutes les photographies pour l'encart en bas de page qui représente tous les likes d'un photographe.

Une fois cette étape réalisée, vous aurez :

- Les photos et vidéos de la galerie pourront être likées.
- L'encart en bas de page affiche maintenant le nombre total de likes

Recommandations :

- Vous pouvez gérer le nombre de likes total depuis votre photographer Template.
- Ici, il n'y a pas besoin de sauvegarder vos likes : si vous rafraîchissez la page, les likes ne doivent pas être sauvegardés.

Points de vigilance :

L'utilisateur ne doit pouvoir liker chaque photo qu'une seule fois, attention à votre implémentation.

Étape 9 : Créer le système de tri

90% d'avancement

Vous allez maintenant implémenter le système de tri pour les photographies sur la page détaillée du photographe.

 **Une fois cette étape réalisée, vous aurez :**

- La page photographe finale intégrant la fonctionnalité de tri
- Rapport d'accessibilité de la page photographe

 **Recommandations :**

- Vous pouvez utiliser la méthode `sort` pour faire votre tri.
- Une fois votre page photographe finalisée, vous pouvez également réaliser un rapport d'accessibilité avec un validateur d'accessibilité ou une checklist, et corriger votre code en fonction.

 **Points de vigilance :**

- Attention à bien gérer l'accessibilité en permettant de sélectionner les éléments au clavier dans le menu déroulant.

 **Ressources :**

- Dans le chapitre "[Manipulez les listes en JavaScript](#)" du cours "Créez des pages web dynamiques avec JavaScript" vous verrez comment utiliser la méthode "`sort`" pour trier des tableaux

Étape 10 : Vérifier le code avec un linter

100% d'avancement

Pour terminer ce projet, vous devrez faire tourner un linter sur votre code.

 **Une fois cette étape réalisée, vous aurez :**

- La base de code terminée et vérifiée avec un linter de code

 **Recommandations :**

- Vous devrez mettre en place votre linter de code (ESLint, JSLint, etc.)
- Vous êtes autorisé à faire taire certains avertissements

 **Ressources :**

- Le chapitre “[Vérifiez que votre code respecte les conventions de codage](#)” du cours “Créez des pages web dynamiques avec JavaScript” vous aidera à prendre en main ESLint.



Projet terminé !