



Créez un site accessible pour une plateforme de photographies

FishEye



Sommaire

- Rappel de ce qu'est FishEye
- Rappel sur la mission et consignes technique qui m'ont été donné
- Présentation de l'environnement de travail
- L'architecture
- Résultat



Présentation FishEye

- FishEye est un site web qui permet aux photographes indépendants de présenter leurs meilleurs travaux



Consignes

- Le code est séparé en différents fichiers (HTML avec balises d'accessibilité, CSS, JavaScript).
- Une version moderne (ES6 ou supérieure) de JavaScript est utilisée et les fonctionnalités obsolètes ne sont pas utilisées
- La programmation orientée objet est utilisée
- Commenter le code
- Le code est lisible



Consignes supplémentaires

- L'accessibilité est clé !
 - "Il est très important que notre site soit accessible aux utilisateurs malvoyants. Toutes nos photos doivent comporter des descriptions textuelles, et vous devez les inclure dans la page. De plus, l'utilisateur doit pouvoir utiliser les commandes du clavier pour naviguer sur le site, comme les touches fléchées de la lightbox".
 - Utilisez des éléments HTML "sémantiques" qui décrivent leur intention autant que possible, au lieu de mettre des éléments `<div>` et `` partout.
 - Lorsque vous devez créer un élément personnalisé, ajoutez des attributs ARIA pour décrire ce qu'il fait.
 - Le code devrait passer les tests AChecker sans "known issue" (afin qu'il soit conforme aux WCAG).
 - Toute la gestion des événements (par exemple, les clics et les pressions au clavier) doit être configurée (utilisez `KeyboardEvent.key` ou `KeyboardEvent.code`).
 - Utilisez un lecteur d'écran gratuit pour vous faire une idée de ce que représente l'utilisation du site pour une personne malvoyante.



Environnement de travail

- Éditeur VSCODE
- Linter JS Eslint
- Linter css stylelint
- Versionnage via Git et GitHub
- Firefox et chrome



Page d'accueil

- Liste de tous les photographes avec leur nom, leur slogan, leur localisation, leur prix/heure, leurs tags et une image miniature de leur choix.
- En cliquant sur une étiquette (tag) dans la barre de navigation, la liste des photographes est filtrée pour n'afficher que ceux qui correspondent à cette étiquette.
- Lorsque l'utilisateur clique sur la vignette d'un photographe, il est amené à sa page



Pages des photographes

- Affiche une galerie des travaux du photographe.
- Les photographes peuvent montrer à la fois des photos et des vidéos.
 - Dans le cas des vidéos, montrer une image miniature dans la galerie.
- Chaque média comprend un titre et un nombre de likes.
 - Lorsque l'utilisateur clique sur l'icône "Like", le nombre de likes affiché est incrémenté.
 - Le nombre de likes total d'un photographe doit correspondre à la somme des likes de chacun de ses médias.
- Les médias peuvent être triés par popularité ou par titre.
- Lorsque l'utilisateur clique sur un média, celui-ci doit s'ouvrir dans une lightbox



Lightbox

- Lightbox
 - Lorsque la lightbox est affichée, il y a une croix dans le coin pour fermer la fenêtre.
 - Des boutons de navigation permettent de passer d'un élément média à l'autre (les utilisateurs peuvent cliquer sur ces boutons pour naviguer).
 - Les touches fléchées permettent également de naviguer entre les médias.
- Afficher un bouton pour contacter le photographe.
 - Le formulaire de contact est une modale qui s'affiche par-dessus le reste.
 - Il comprend des champs pour les noms, l'adresse électronique et le message.
 - Plus tard, le bouton de contact enverra un message au photographe, Pour l'instant, seulement afficher le contenu des trois champs dans les logs de la console.



Architecture

- Architecture MVC
 - url optimisées pour le SEO
- Modules
 - Lightbox
 - Contact form
 - likes
- Composants html
- Utilisation du factory design pattern pour la gestion des medias