Soutenance projet 11 Kasa

* Présentation des différentes pages, avec la navigation entre elles, et interactions avec tous les éléments interactifs.
* Parcours des fichiers de code avec le mentor. Mise en avant des différents composants React.

Bonjour Laura, nous allons donc voir, contrôler et vérifier ensemble le projet qui m’a été confié récemment, la refonte du site Kasa pour notre client.

Je me suis donc servi de l’éditeur de code Visual studio Code, du framework (ou librairie) React, de React Router pour la gestion des routes, du pré-processeur Sass pour le Css, et n’ai utilisé aucune librairie externe non plus.

Nous allons aussi voir que le site est entièrement responsive, dans le plus grand respect de la maquette que ce soit pour les versions desktop ou mobile. J’ai d’ailleurs codé celui-ci dans l’approche mobile-first.

Il n’y a évidemment aucune alerte ou avertissement à la console, et le code est bien lisible, indenté et commenté pour que toute personne puisse s’y retrouver facilement.

Nous allons constater ensemble que les contraintes techniques sont bien respectées :

* Le code est découpé en composants modulaires et réutilisables (Montrer)
* Un composant par fichier
* -Structure logique des différents fichiers
* Utilisation des props et du state (voir composant Announces, SampleLoc, Logement)
* Gestion des évènements (Composant Slider)
* Les composants sont tous à base de classes
* Utilisation des listes (Composant Announces)

Pour le Router

* Les paramètres des routes sont bien gérés par React Router et il existe une route par page (démo)
* Quand une route est inexistante, le composant Erreur est retourné. Il y a d’ailleurs une erreur générale, et une autre gérée grâce aux props pour afficher en cas de problème de récupération des données du logement (démo)
* Et le tout est réuni dans un seul fichier

Nous allons ensuite faire un tour du site et de son fonctionnement, et de l’intercation entre les différentes pages et éléments.

Nous avons le Header, avec le logo du site, et 2 liens ‘Accueil’ et ‘A propos’. La page sur laquelle nous nous trouvons doit être soulignée en rouge, ce qui est le cas. Il suffit de rajouter underline dans le NavLInk correspondant (composant Header)

Les logements issus du fichier json fourni en attendant que le backend soit connecté à l’appli, apparaissent sur la page principale.

De petits effets amènent un peu de vie au site et le rendent plus convivial pour l’utilisateur, nous pourrons regarder plus en détail cela ensemble et vir avec le client si il apprécie ou souhaite y amener des modifications bien sûr.

Au clic sur un logement, la page dédiée à celui-ci apparaît. Dans le composant announces nous pouvons voir que l’adresse vers laquelle nous sommes redirigés affiche l’id de l’appartement, nous le récupérons ensuite pour afficher les données propres à chaque logement.

Nous avons donc la galerie, et comme demandé dans les consignes, si il y a plusieurs images et que nous arrivons à la dernière, le retour à la première est automatique, et la même chose quand nous allons dans l’autre sens.

Si le logement ne fournit qu’une seule image, à ce moment là les flèches de changement d’image disparaissent, comme sur cette annonce.

Nous avons ensuite les informations statiques sous la galerie (ville, description, notation, propriétaire…)

Et viennent ensuite les collapses, qui sont donc fermées à l’ouverture, et qui au clic vont s’ouvrir et afficher leur contenu.

Ensuite la page à propos, qui ne contient qu’une bannière (là aussi légèrement animée) et des collapses là aussi. La différence avec celles de la page logement, est que dans la version mobile, la taille de la police reste la même que la version desktop, tandis que celles de la page logement ont une taille de police plus petite sur la version mobile. J’ai seulement appliqué une classe différente pour chacune.

Enfin, il est à noter que j’ai rendu le site accessible à la navigation au clavier et pour les lecteurs d’écran