

Vous êtes développeur front-end en freelance pour Kasa, une entreprise de location d'appartements entre particuliers.

Avec plus de 500 annonces postées chaque jour, Kasa fait partie des leaders de la location d'appartements entre particuliers en France.

L'occasion parfaite pour vous d'ajouter une belle référence à votre portfolio de freelance !

Le site de Kasa a été codé il y a maintenant plus de 10 ans en ASP.NET avec un code legacy important. Laura, la CTO, a donc lancé une refonte totale pour passer à une stack complète en JavaScript avec NodeJS côté back-end, et React côté front-end.

Kasa en a également profité pour commander de nouvelles maquettes auprès de Paul, son designer habituel, qui est freelance. Un gros chantier pour cette année !

Un point est prévu avec la CTO plus tard dans la semaine, mais pour que vous puissiez vous familiariser avec les enjeux de votre mission, vous trouvez le récapitulatif suivant dans votre boîte mail :

Objet : Brief refonte site Kasa

De : Laura

À : Moi

Hello,

Bienvenue ! Tu vas pouvoir nous aider à donner vie à un chantier sur lequel on travaille depuis plusieurs mois.

Ton objectif : démarrer le projet React et développer l'ensemble de l'application, les composants React, les routes React Router, en suivant les maquettes Figma (responsives !) et toutes les infos que je te donne ci-dessous. Et ce, avec un code de qualité !

Back-end/data : une première version du backend a été réalisée et se trouve disponible sur GitHub.

Tu vas pouvoir t'en servir pour récupérer les données du site. Pour l'instant, le back-end n'a pas encore été déployé, tu pourras donc le télécharger et, en suivant les instructions du document README.md, le lancer localement pour pouvoir récupérer les informations des propriétés. Tu trouveras dans les coding guidelines de Kasa les informations sur les différentes routes qui te seront utiles pour récupérer les données des propriétés. Pour réaliser les tâches qui te sont assignées, tu n'auras pas besoin de modifier le code du back-end.

Contraintes techniques : pour la réalisation du projet, n'utilise pas d'autres packages que ton bundler (Vite), React Router et éventuellement SASS si tu veux l'utiliser. On souhaite que tu réalises l'application front-end avec le minimum de dépendances. Les coding guidelines te seront utiles ici.

Très bonne journée à toi !

Laura

Deux heures plus tard

Vous recevez un e-mail de Paul.

Objet : Fonctionnalité et design

De : Paul

À : Moi

Salut,

Bienvenue parmi nous ! Laura m'a dit de te briefer sur le design de la nouvelle version du site, alors voici les infos clés.

Design : Voici les maquettes sur Figma pour le design d'interface.

J'ai l'habitude de travailler avec la logique de composants sur Figma ; Laura m'a dit que ça te faciliterait le travail sur React.

Tu trouveras toutes les ressources dont tu as besoin directement dans la maquette (logo, icônes pour les composants, etc.). Pour cela, il suffit de cliquer sur la ressource souhaitée et de faire "Exporter" au format voulu. Sur les vignettes des logements, pour les images de couverture, j'ai mis un rectangle orange que tu peux remplacer par l'image correspondante.

Pour avoir un rendu le plus réaliste possible de l'application, utilise les prototypes du site disponibles ici.

Ils te permettront notamment de voir les animations attendues sur les menus déroulants (Collapse).

Contraintes fonctionnelles - Quelques précisions sur les fonctionnalités du site :

Pour le défilement des photos dans la galerie (composant Gallery) :

si l'utilisateur se trouve à la première image et qu'il clique sur "Image précédente", la galerie affiche la dernière image ;

inversement, quand l'image affichée est la dernière de la galerie, si l'utilisateur clique sur "Image suivante", la galerie affiche la première image ;

s'il n'y a qu'une seule image, les boutons "Suivant" et "Précédent" n'apparaissent pas.

La galerie doit toujours rester de la même hauteur, celle indiquée sur la maquette Figma. Les images seront donc coupées et centrées dans le cadre de l'image ;

Collapse : par défaut, les Collapses sont fermés à l'initialisation de la page ;

Si le Collapse est ouvert, le clic de l'utilisateur permet de le fermer ;

Inversement, si le Collapse est fermé, un clic permet de l'ouvrir.

Bon courage,

Paul

Quelques minutes plus tard

Vous recevez un mail de Laura.

Objet : Testing

De : Laura

À : Moi

Salut,

J'aimerais que tu prépares des tests unitaires sur l'appli pour rendre l'application plus robuste.

Cependant, afin de pouvoir évaluer ce que cela implique de changer dans nos développements, j'aimerais que tu puisses faire les tests unitaires d'au moins un composant avec state et d'un composant sans state.

Nous souhaitons pour ces premiers tests un taux de coverage pour chacun des composants testés d'au moins 80 %.

Laura

Ça y est, vous avez toutes les informations nécessaires, vous pouvez démarrer votre développement !