



Kosa

Application web de location immobilière avec React

Introduction

- **Rôle et contexte :** Développeur front-end freelance, refonte du site Kasa.
- **Objectifs :**
 - Apprentissage de React avec Vite
 - Navigation fluide avec React Router
 - Respect des maquettes Figma et gestion des erreurs.
- **Contraintes techniques :**
 - React Router,
 - Récupération de données,
 - Animations CSS,
 - Responsivité.

Architecture et technologies

- Pourquoi Vite plutôt que CRA ?
- Organisation du projet : arborescence des fichiers.

Logique de routage avec React Router

- Définition des routes (/home, /about, /properties/:id) via `PublicRouter.jsx`.
- Gestion des erreurs : redirection vers `Error.jsx` si la route n'existe pas.
- `Layout.jsx` : Structure globale avec Header et Footer.

Présentation des composants et de leur logique

- React est basé sur une approche modulaire : chaque élément est un composant réutilisable.
- Exemples :
 - Banner.jsx (statique)
 - Gallery.jsx (dynamique avec useEffect)
 - Card.jsx (données dynamiques)
 - Carrousel.jsx (useState)
 - Collapse.jsx (useState)

Gestion des données et interactions utilisateur

- useEffect et useState sont essentiels en React :
- useEffect : récupération des logements depuis l'API (Gallery.jsx).
- useState : gestion de l'affichage dynamique (Carrousel, Collapse).

Expérience utilisateur et design

- Animations CSS : transitions fluides (Collapse, Carrousel).
- Responsive design : adaptation sur mobile (media queries dans banner.css, gallery.css).

Conclusion et perspectives

- Ce que j'ai appris : Développement React avec Vite, navigation avec React Router, hooks (useState, useEffect).
- Respect du cahier des charges : composants réutilisables, récupération dynamique des données, animations fluides.
- Axes d'amélioration : Ajout de tests unitaires, nouvelles fonctionnalités (ex : filtres de recherche).