

# Développer un algorithme de recherche en JavaScript

Projet **Kosa**

Sonia KECHIT

# Sommaire

1 – Présentation du contexte

2 – Présentation du projet

3 – Fonctionnalités

4 – Choix et contraintes techniques

5 – Démonstration

6 – Bilan

7 – Questions

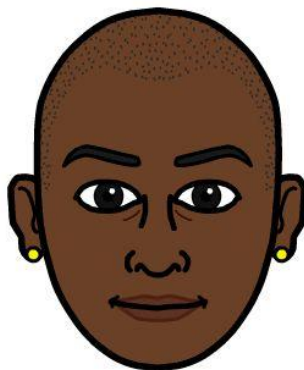
Lien Git : [https://github.com/KsoniaK/KECHIT\\_Sonia\\_Kasa\\_12-2022.git](https://github.com/KsoniaK/KECHIT_Sonia_Kasa_12-2022.git)

# Présentation du contexte

## L'ÉQUIPE



Laura  
CTO



Paul  
Designer



Sonia  
Développeuse front-end

# Présentation du projet

Kasa est dans le métier de la location d'appartements entre particuliers depuis près de 10 ans maintenant. Avec plus de 500 annonces postées chaque jour, Kasa fait partie des leaders de la location d'appartements entre particuliers en France.

## Les objectifs de l'entreprise:

- ➡ Une refonte totale pour passer à une stack complète en JavaScript avec NodeJS côté Back-end, et React côté Front-end
- ➡ Avoir un code de qualité

# Fonctionnalités

- Galerie :

1. Si l'utilisateur se trouve à la première image et qu'il clique sur "image précédente", la galerie affiche la dernière image.
2. Inversement, quand l'image affichée est la dernière de la galerie, si l'utilisateur clique sur "image suivante", la galerie affiche la première image.
3. S'il n'y a qu'une seule image, les boutons "suivant" et "précédent" ainsi que la numérotation n'apparaissent pas.
4. La galerie doit toujours rester de la même hauteur que celle indiquée sur la maquette Figma. Les images seront donc coupées et centrées dans le cadre de l'image

- Collapse :

1. Par défaut, les Collapse sont fermés à l'initialisation de la page
2. 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.

# Choix et contraintes techniques

## Choix techniques :



React



React Router



HTML5



CSS3

## Contraintes techniques :

### React :

- Découpage en composants modulaires et réutilisables
- Un composant par fichier
- Structure logique des différents fichiers
- Utilisation des props entre les composants
- Utilisation du state dans les composants quand c'est nécessaire
- Gestion des événements
- Listes : itérer (avec l'opérateur map par exemple) autant que possible

### React Router :

- Les paramètres des routes sont gérés par React Router dans l'URL pour récupérer les informations de chaque logement
- Il existe une page par route
- La page 404 est renvoyée pour chaque route inexistante, ou si une valeur présente dans l'URL ne fait pas partie des données renseignées
- La logique du routeur est réunie dans un seul fichier



# Démonstration

# Bilan

- ➡ Création d'une application React / React router
- ➡ Utilisation des composants fonctionnels
- ➡ Utilisation des props entre les composants
- ➡ Utilisation des Hooks (useState, useEffect)
- ➡ Création d'une page Error et redirection
- ➡ Gestion des évènements
- ➡ Création de composants réutilisables



The background features a cluster of overlapping circles in various colors (red, yellow, green, blue, and dark green) of different sizes. Each circle contains a white question mark. Some circles are semi-transparent, creating a layered effect. The entire graphic is set against a dark gray background.

Questions ?