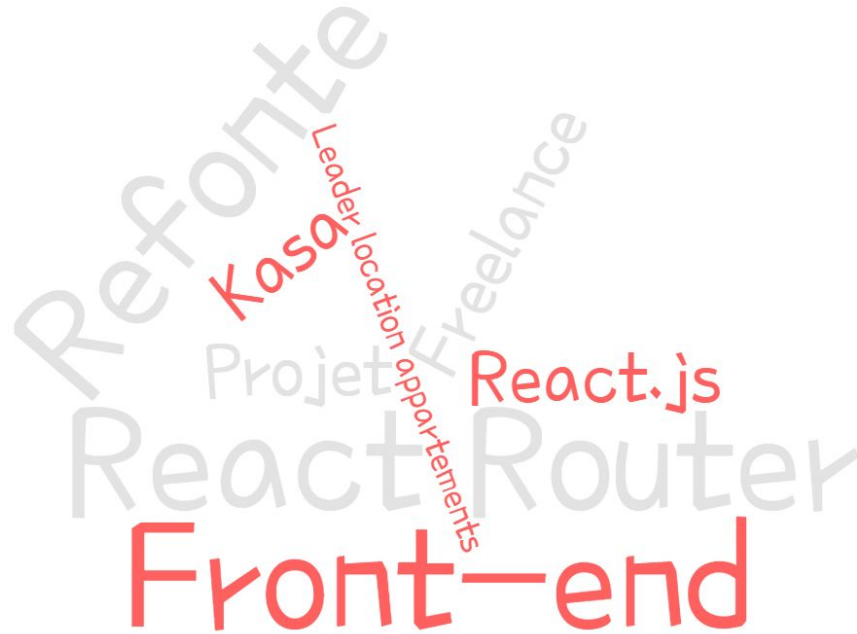


Projet



Développez une application Web  
avec React et React Router

# Contexte



A word cloud graphic with the following text elements:

- Refonte
- Kasa
- Leader location appartements
- Freelance
- Projet
- React.js
- React Router
- Front-end

# Adaptation projet

Pour s'imprégner du projet, documents et outils à disposition :

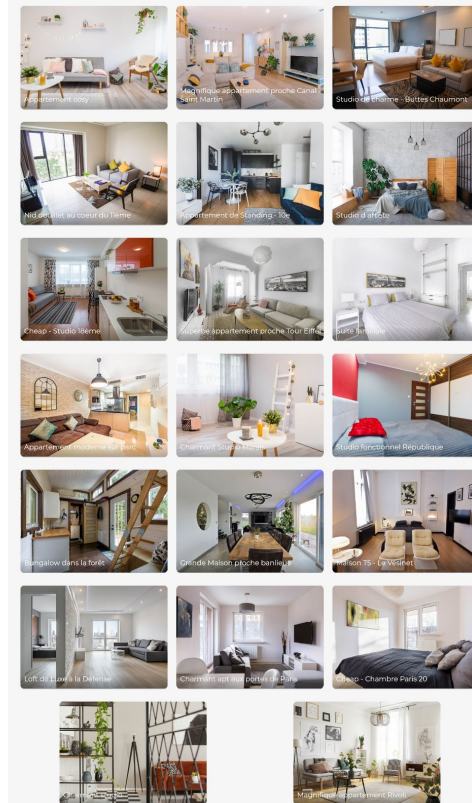
- Mail d'informations sur la mission Laura CTO et Paul Designer
- Outils et contraintes techniques (PDF)
- Maquette Figma :  
<https://www.figma.com/file/bAnXDNqRKCRRP8mY2gcb5p/UI-Design-Kasa-FR?node-id=4%3A2>
- Fichiers JSON

Projet codé

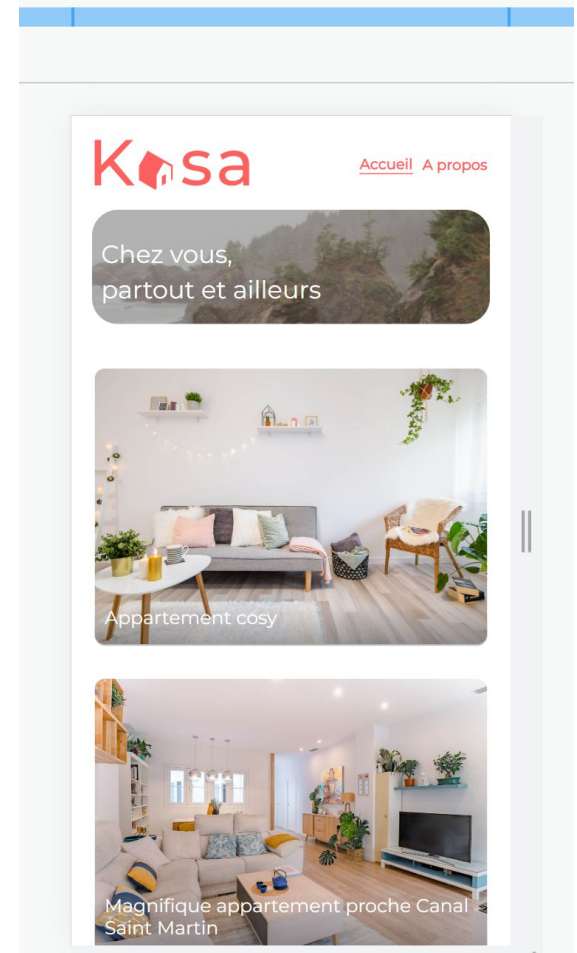
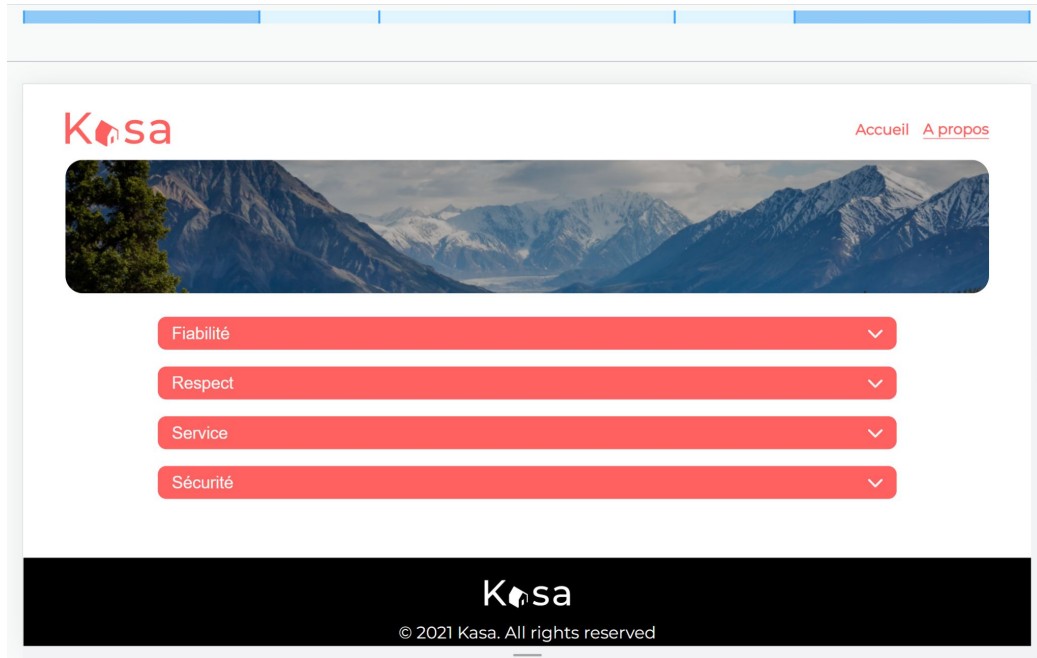
# Projet codé

- Projet codé et conforme à la maquette Figma
- Comprend :
  - L'accueil
  - A propos
  - Les locations
  - Erreur 404

Chez vous, partout et ailleurs



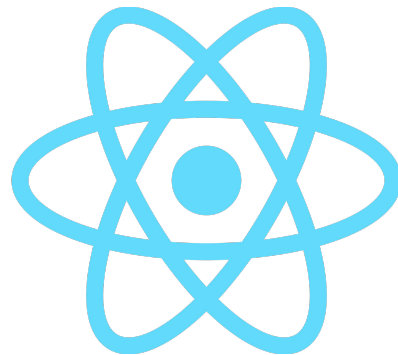
# Responsive



React

# React.js

- Bibliothèque JS
- Créer des UI à partir de composants :
  - modulaires
  - flexibles
  - réutilisables
  - rapides



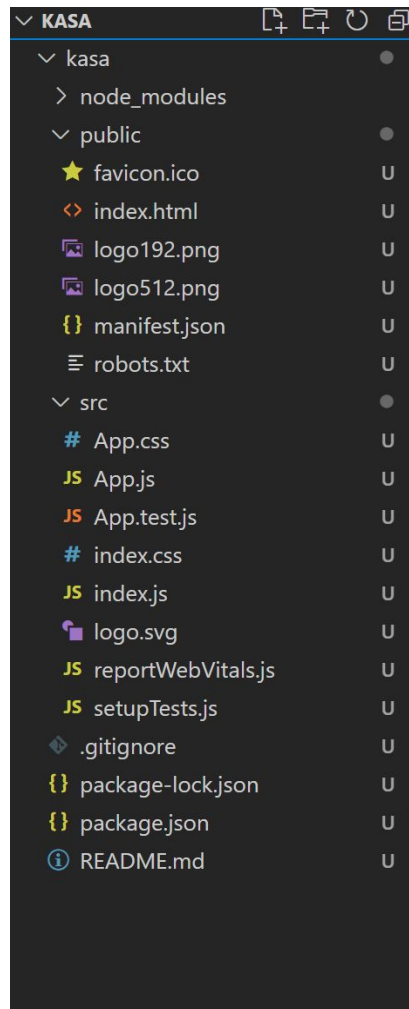


# Fonctionnement React

- React génère un DOM virtuel avec les instructions du JSX, distinct du DOM du navigateur
- améliore la performance
- Un composant par fichier ( composants fonctionnels dans ce projet)
- Utilisation des props et propTypes
- Utilisation des hooks dans les composants quand c'est nécessaire
- Gestion des événements
- Listes (en itérant dessus, par exemple avec map) et conditions
- Syntaxe : déstructuration et décomposition Es6
- Routeur

# Architecture du code

- install via le terminal : `npx create-react app`
- structure de fichiers
- rendu sur <http://localhost:3000/> : `npm start`



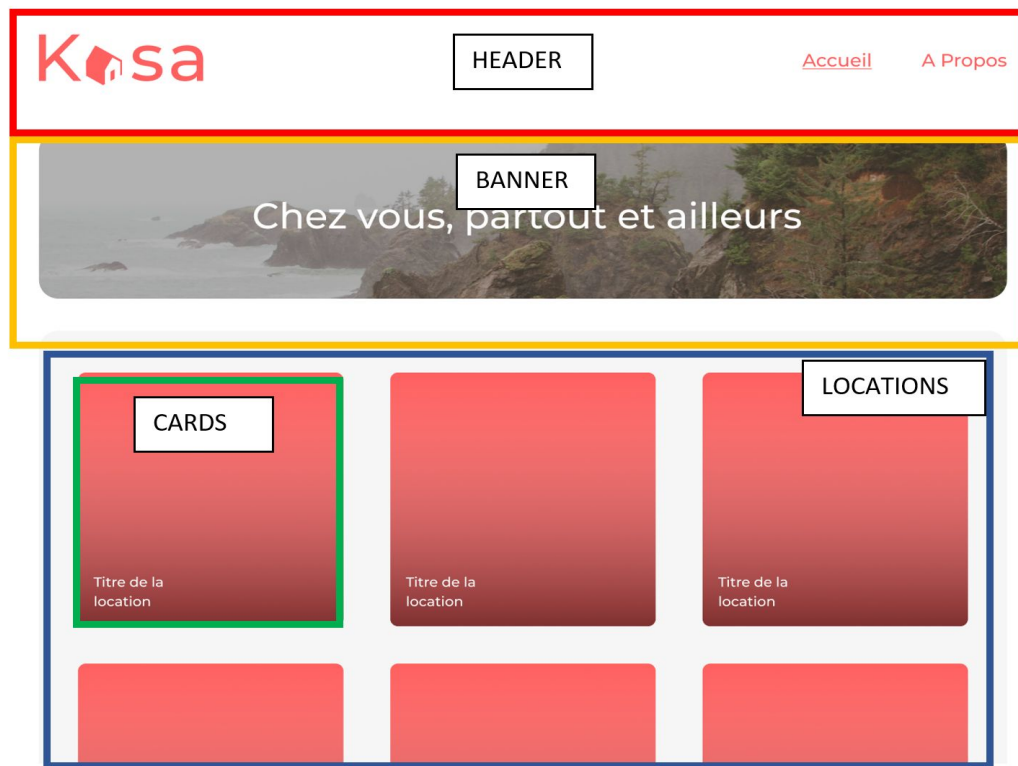
# Décomposition du projet

# Concept : composant

- Penser “Composant” :
  - UI est composée de multiples composants réutilisables
  - regroupe : structure (HTML) + style (CSS) + comportement (JS)
  - React traduit le composant en “gros” objet greffé au DOM
  - Cycle de vie :
    - le montage
    - le changement d'état ou mise à jour
    - la destruction

# Décomposition maquette

DOC WORD



# Composant générique

- Banner et Dropdown

```
import PropTypes from 'prop-types';

/*
 * @param {object} props change la bannière
 * @param {string} props.image > background picture
 * @param {string} props.alt > picture alt text
 * @param {string} props.title > banner title (can be different or empty)
 * @returns banner
 */

const Banner = ({ image, title, alt }) => {
  return (
    <section className="banner">
      <h1 className="banner-title">{title}</h1>
      <img className="banner__image" src={image} alt={alt}></img>
    </section>
  );
};

Banner.propTypes = {
  image: PropTypes.string,
  alt: PropTypes.string,
  title: PropTypes.string,
};

export default Banner;
```

# Composant générique 2

- construction dans le code :
  - réutilisable dans les pages

```
const Home = () => {  
  return (  
    <div className="Home">  
      <Header />  
      <Banner image={banner1} alt="Paysage" title="Chez vous, partout et ailleurs" />  
  
      <div className="Cards">
```

```
const Apropos = () => {  
  return (  
    <div className="Apropos">  
      <Header />  
      <Banner image={banner2} type="about" />  
      <section className="accordion">
```

# Props

- Propriété que l'on passe à un composant :
  - passé en extérieur à la fonction
  - rend le composant unique
  - dynamise le rendu

```
const Banner = ({ image, title, alt }) => {
```

- Déstructuration ES6
  - syntaxe qui vous permet d'affecter des propriétés d'objets ou des éléments de tableau comme variables.
  - réduit les lignes de code nécessaires



# PropTypes

- vérifie que les données reçues sont valides
- précise le type de props utilisées
- permet d'éviter les erreurs dans la console

```
Banner.propTypes = {  
  image: PropTypes.string,  
  alt: PropTypes.string,  
  title: PropTypes.string,  
};
```

# Gestion des évènements

- syntaxe : déclarer les évènements + { fonction à appeler }
- Ajout de “onClick” dans le HTML

```
return (  
  <section className="carrousel">  
    <FontAwesomeIcon  
      icon={faChevronLeft}  
      className={`carrousel__chevronLeft chevron ${className}`}  
      //gestion des évènements  
      onClick={previous}  
    />  
    <img  
      src={pictures[activeIndex]}  
      className="carrousel__image"  
      alt="cover"  
    />  
    <FontAwesomeIcon  
      icon={faChevronRight}  
      className={`carrousel__chevronRight chevron ${className}`}  
      onClick={next}  
    />  
  </section>  
)
```

# Hook ( V16)

- fonction en JavaScript avec la possibilité de lui passer un ou plusieurs paramètres et cette fonction va renvoyer un résultat.
- 2 règles :
  - on appelle un hook uniquement à la racine de la fonction
  - on appelle un hook depuis des composants fonctionnels
- refactorisation
- logique /esprit pratique

# Hook : useState

- Etat local interne au composant
- fonction qui interagit avec l'état du composant et modifie son cycle de vie

```
const [house, setHouse] = useState(null);
```

- syntaxe en décomposition:
  - `const [ état d'origine , setState de l'état ] = useState(etat initial choisi)`
  - renvoie une paire de valeurs dans un tableau de 2 valeurs qui sont récupérées entre les crochets.

# Hook : useEffect

- permet d'intervenir sur les événements du cycle de vie du composant
- gère les effets de bords

```
useEffect(() => {  
  // Recherche de l'annonce avec l'identifiant qui se trouve dans l'URL.  
  const foundHouse = housing.find((house) => house.id === param.id);  
  if (!foundHouse) {  
    navigate('/error');  
  } else {  
    setHouse(foundHouse);  
  }  
}, []);
```

- syntaxe
  - fonction avec une callback
  - tableau de dépendances [ ] que l'on veut observer

# Hook : useParams et UseNavigate ( react router)

- useParams()
  - permet d'accéder au paramètre de l'URL pour déterminer ce qui est affiché dans le composant rendu par une route dynamique
- useNavigate()
  - permet de naviguer entre les pages web

```
const Location = () => {  
  const param = useParams();  
  const [house, setHouse] = useState(null);  
  
  // useNavigate() : hook qui permet de naviguer entre les pages  
  const navigate = useNavigate();
```

# React Router

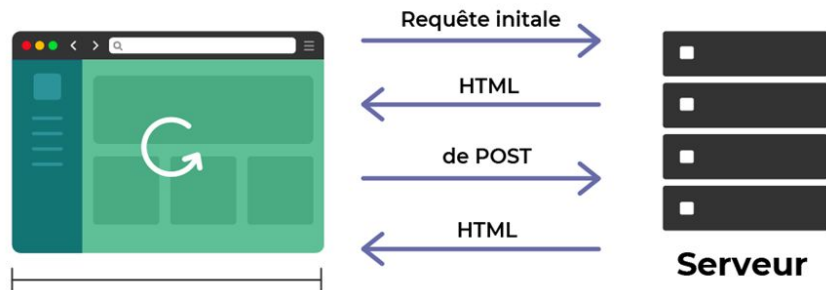
# Single Page

- Web classique : rechargement page à chaque interaction du user

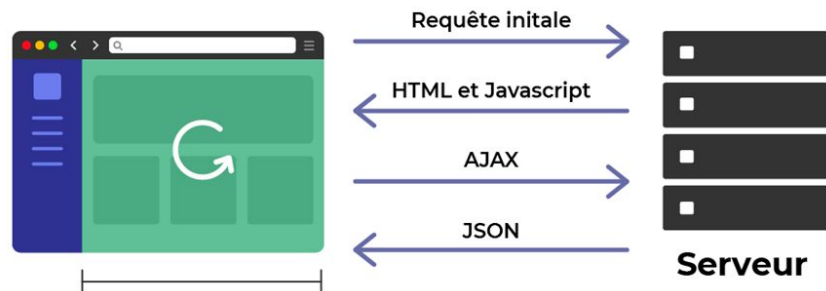
- SPA : charge une page 1 seule fois
- interactions dynamiques
- navigue entre les pages sans besoin

de recharger à chaque action

## Site web classique



## SPA





# React Router

- Bibliothèque de routage standard dans React
- maintient l'interface synchrone avec l'URL dynamique
- commande Terminal : `npm install react-router-dom`
- routes : affiche les composants si le chemin de l'URL correspond au chemin de Route

```
ReactDOM.createRoot(document.getElementById('root')).render(  
  <BrowserRouter>  
    <App />  
  </BrowserRouter>  
);
```

```
const App = () => {  
  return (  
    <>  
      <Routes>  
        <Route exact path="/" element={<Home />} />  
        <Route path="/housing/:id" element={<Location />} />  
        <Route path="/a-propos" element={<About />} />  
        <Route path="*" element={<Error404 />} />  
      </Routes>  
    </>  
  );  
};  
  
export default App;
```

Eslint

# Eslint

- Install Eslint
- Fichier JSX

```
.eslintrc.js X
kasa > .eslintrc.js > [?] <unknown> > parserOptions > ecmaFeatures

1  module.exports = {
2    env: {
3      browser: true,
4      amd: true,
5      node: true,
6    },
7    extends: ['eslint:recommended', 'plugin:react/recommended', 'plugin:react/jsx-runtime'],
8
9    parserOptions: {
10     ecmaFeatures: {
11       jsx: true,
12     },
13     ecmaVersion: 'latest',
14     sourceType: 'module',
15   },
16   plugins: ['react'],
17
18   rules: {
19     semi: ['error', 'always'],
20     quotes: ['error', 'single'],
21   },
22 };
23
24
```

# Repository

# Github

Lien vers le repo Github :

[https://github.com/Laurene45/LaureneCourde\\_11\\_24032022/tree/main/kasa](https://github.com/Laurene45/LaureneCourde_11_24032022/tree/main/kasa)

Laurene45 / LaureneCourde\_11\_24032022 Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main LaureneCourde\_11\_24032022 / kasa / Go to file Add file ...

Laurene45 suppr commentaires et vérifications des routes react -router ✓ b00dd23 22 minutes ago History

..		
public	suppression fichiers inutiles	21 days ago
src	suppr commentaires et vérifications des routes react -router	22 minutes ago
.eslintrc.js	mise à jour des fichiers indentation et mise en forme du code + créat...	13 days ago
.gitignore	Ajout de l'architecture des dossiers react kasa	last month
.prettierrc	mise à jour des fichiers indentation et mise en forme du code + créat...	13 days ago
README.md	Ajout de l'architecture des dossiers react kasa	last month
package-lock.json	mise à jour des fichiers indentation et mise en forme du code + créat...	13 days ago
package.json	mise à jour des fichiers indentation et mise en forme du code + créat...	13 days ago

# Avez-vous des questions ?

