L'objectif est de redévelopper la page d'accueil de l'application Kaz'A'Burger en utilisant des composants React

Le fichier #02-starter.zip contient les éléments dont vous aurez besoin pour démarrer

I. Création de l'application

- Placez-vous dans votre dossier de travail, depuis un terminal
- Créez un dossier **kazaburger**
- Placez-vous dans le dossier créé
- Créez une nouvelle application Vite/React nommée « **front-end** »
- Placez-vous dans le dossier de l'application
- Installez les modules nécessaires à Vite
- Installez et configurez le nécessaire pour Tailwind CSS
- Lancez le serveur pour vérifier son bon fonctionnement

II. Préparation

- Supprimez le dossier « src » de l'application
- Décompressez le fichier « #02-starter.zip »
- Déplacez son contenu dans le dossier de l'application
- Ouvrez le Composant App, observez les modifications sur les attributs « class », le chemin des images , la structure des sections
- Ouvrez le fichier index.css et observez l'utilisation des directives TailWind

A ce stade la page devrait ressembler à ceci

III. Décomposition de la page d'accueil

La page d'accueil sera décomposée en 3 composants : <Header> , <Content> et <Footer>

- Dans le dossier « src », créez un dossier « components »
- Créez-y 3 fichiers Header.jsx, Content.jsx et Footer.jsx
- Transformez ces fichiers en composants dignes de ce nom ☺
- Répartissez le code interne du composant <App> comme suit
  - composant <Header> : La section « header »
  - composant <Content> : La section « content », « suggestions », «newsletter » et « témoignages » . Attention il y a plusieurs éléments HTML
  - composant <Footer> : la section « footer »
- Modifiez le composant App pour utiliser les composants précédents

IV. Le composant <Logo>

Il sera chargé d'afficher le logo utilisé dans les composants <Header> et <Footer>.

- Dans le dossier « components », créer un composant <Logo> et placez-y la balise <img ****> en provenance du composant <Header>
- Modifiez les composants <Header> et <Footer> pour utiliser le composant <Logo>

V. Le composant <Content>

Il sera chargé d'afficher le contenu du site , pour le moment juste la page d'accueil présentant les produits en vedette, les suggestions, l'inscription à la newsletter et les témoignages

- Créez un dossier « Content » qui contiendra les différentes sections du site
- Créez à l'intérieur un dossier « Home » qui contiendra les composants de la page d'accueil
- Dans le dossier « Content », Créez un composant <Home>
- Un premier niveau de décomposition serait déjà de séparer les différentes sections de la page d'accueil, en 4 composants <Featured>, <Suggests>, <KazaNews> et <Testimonies >
- Dans le dossier « Home », créez 4 composants <Featured>, <Suggests>, <KazaNews> et <Testimonies >
- Importez-y le code correspondant, à partir du composant <Content>
- Modifiez le composant <Home> pour utiliser ses sous-composants
- Modifiez le composant <Content> à son tour, pour utiliser le composant <Home>

VI. Le composant <Suggests>

Il affiche les suggestions de produits, en fonction des gouts de l'utilisateur.
Ouvrez le composant, et observez la structure des balises
Elle pourrait se résumer à :

<section>
Titre
Contenu
Element1
Element2
Element3

```
<section className="suggests">
    <h2>Nos suggestions</h2>
    <div className="content">
      <div>…
      </div>
      <div>…
      </div>
      <div>…
      </div>
    </div>
  </section>
```

Est-ce qu'il ne serait pas intéressant de déléguer à un composant « SuggestItem » l'affichage des éléments de chaque suggestion et éviter la duplication de code ?

Pour cela nous allons faire pas à pas quelques actions :

- Mettre dans un tableau les données des suggestions
- Factoriser dans un composant le code qui gère les éléments
- Parcourir le tableau et Appeler notre composant en lui fournissant les données à afficher

Joli programme non ? Allez on y va ? GOOOOOOOOOOO

- Dans le corps de la fonction « Suggests », la partie entre l'accolade ouvrante et l'instruction return

```
const Suggests = () => {
  return (
    <section className="suggests">…
    </section>
  )
}
```

- Insérez le code suivant

```
const data = [
    {
      image: "src/assets/images/classic-burger.jpg",
      imageAlt: "Classic burger",
      title: "Classic burger",
      description:
        "Le classNameic Burger est un délicieux sandwich à base d'un pain brioché moelleux et doré à point, qui
renferme un savoureux steak haché de bœuf grillé à la perfection. Il est parfait pour les petites faims.",
      price: "6.90 €",
    },
    {
      image: "src/assets/images/kingdom-burger.jpg",
      imageAlt: "kingdom burger",
      title: "Kingdom burger",
      description:
        "Le Kingdom Burger est un délicieux sandwich à base d'un pain brioché moelleux et doré à point, qui
renferme deux savoureux steaks hachés de bœuf grillé à la perfection. Il est parfait pour les gourmands !",
      price: "7.90 €",
    },
    {
      image: "src/assets/images/supreme-burger.jpg",
      imageAlt: "supreme burger",
      title: "Suprême burger",
      description:
        "Le Suprême Burger est un délicieux sandwich à base d'un pain brioché moelleux et doré à point, qui
renferme un savoureux steak haché avec des rondelles de tomates et du cheddar.",
      price: "9.90 €",
    },
  ];
```

Attention, il se peut qu'il y ait des soucis avec les sauts de ligne lors du copier/coller depuis le document. Vérifiez dans le code js que les chaines sont bien reprises sans saut de ligne dans les textes longs.

- Dans le dossier « Content », créez un dossier « Suggests »
- Créez à l'intérieur un composant « SuggestItem »
- Déplacez-y le contenu HTML d'une des div de suggestion
- Dans le composant <Suggests>, supprimez le contenu du div.content
- Appelez à la place le sous composant <SuggestItem>
- Vérifiez que le site affiche bien une suggestion

Maintenant nous allons pouvoir parcourir le tableau et appeler le composant sur chaque itération

```
<div className="content"> {data.map((item,i)=><SuggestItem/>)}</div>
```

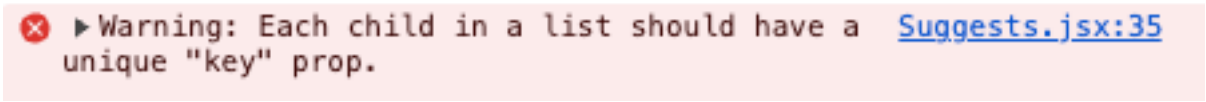
La fonction map nous sauve la vie. Elle prend en paramètre une fonction de callback (ici une fonction anonyme en notation fléchée) cette dernière a, à son tour, 2 paramètres :

- Une variable contenant l'élément lu
- Une variable facultative, correspondant à l'indice de l'élément lu

Pour chaque élément du tableau data, on appellera la fonction SuggestItem, qui affiche un élément de suggestion. Si vous souhaitez faire une boucle for, il faudra le faire dans le corps de la fonction, en ajoutant les composants dans un tableau et afficher ce tableau dans le JSX

Ici vous noterez que c'est le même élément qui est affiché

- Ouvrez l'inspecteur de code et allez dans l'onglet console



A ce stade c'est normal, car on n'a pas mis d'identifiant unique sur chaque SuggestItem comme l'exige React.
Ajoutez l'attribut key sur l'appel du sous composant, en utilisant l'indice. Il a été mis là pour ça ☺

- En utilisant le passage de paramètres entre un composant parent et un composant enfants, via les props, modifiez les 2 composants pour afficher les suggestions.

VII. Le composant <Testimonies>

Il est chargé d'afficher les témoignages clients

- Créez un dossier Testimonies qui contiendra les sous composant nécessaires à l'affichage des témoignages
- A l'instar de ce qui a été fait pour les suggestions, dans le dossier, créez :
  - un composant <Testimony> chargé d'afficher les données d'un témoignage à partir du tableau de données
  - un composant <Rating> chargé d'afficher l'évaluation pour chaque témoignage

```
const data=[
        {
          "image": "src/assets/images/avatar1.jpg",
          "name": "Ella DEUTRANCHE",
          "rating": 4,
          "review": "J'ai essayé plusieurs restaurants de hamburgers dans la région, mais celui-ci est de loin le meilleur. Les
hamburgers sont énormes et savoureux et le service est excellent. Je reviendrai à coup sûr !"
        },
        {
          "image": "src/assets/images/avatar2.jpg",
          "name": "Lambert GOER",
          "rating": 3,
          "review": "Je suis un végétarien difficile à satisfaire, mais j'ai été très content de voir que ce restaurant avait des
options végétariennes pour moi. J'ai commandé un hamburger végétarien et il était délicieux !"
        },
        {
          "image": "src/assets/images/avatar3.jpg",
          "name": "Eva DEVORET",
          "rating": 4,
          "review": "Je suis une habituée de ce restaurant depuis des années et je n'ai jamais été déçue. Les hamburgers sont toujours
frais et délicieux et le personnel est sympathique et accueillant."
        },
        {
          "image": "src/assets/images/avatar4.jpg",
          "name": "Jean REVEU",
          "rating": 5,
          "review": "J'ai été agréablement surpris par la qualité des hamburgers dans ce restaurant. Ils utilisent des ingrédients frais
et les hamburgers sont cuits à la perfection. Je recommande fortement ce restaurant à tous les amateurs de hamburgers !"
        }
      ];
```

- Le composant <Testimonies> parcourt le tableau et affiche, à chaque itération, <Testimony> en lui passant en paramètres les attributs (images, name, rating, review)
- Le composant <Testimony> utilisera le composant <Rating> en lui passant en paramètre la note (rating)
- Le composant <Rating> gère l'affichage de l'évaluation sous forme d'étoiles

Un petit récap des composants créés