

KANAP

Construisez un site e-commerce en JavaScript.

Part Rhani

Spécifications fonctionnelles.

- Les articles doivent s'afficher de manière dynamique.
- Possibilité de sélectionner un coloris et une quantité pour chaque produit.
- Le panier ne doit contenir qu'une ligne par produit de même ID et même couleur.
- Possibilité de changer les quantités ou de supprimer un produit depuis la page panier.
- Le formulaire de commande doit contenir des saisies dans un bon format.
- La validation doit retourner un numéro de commande unique.
- L'utilisation de framework est interdite.

Page d'accueil (index)

Nos produits

Une gamme d'articles exclusifs



KANAP SINOPÉ

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt...



KANAP CYLLÈNE

Morbi nec erat aliquam, sagittis urna non, laoreet justo. Etiam sit amet interdum...



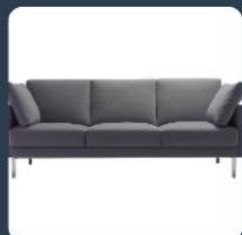
KANAP CALYCÉ

Pellentesque fermentum arcu venenatis ex sagittis accumsan. Vivamus lacinia...



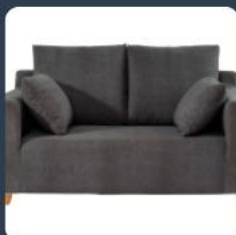
KANAP AUTONOÉ

Donec mattis nisl tortor, nec blandit sapien fermentum at. Proin hendrerit efficitur...



KANAP EURYDOMÉ

Ut laoreet vulputate neque in commodo. Suspendisse maximus quis erat in sagittis....



KANAP HÉLICÉ

Curabitur vel augue sit amet arcu aliquet interdum. Integer vel quam mi. Morbi nec...



KANAP THYONÉ

EMauris imperdiet tellus ante, sit amet pretium turpis molestie eu. Vestibulum et...



KANAP ORTHOSIE

Mauris molestie laoreet finibus. Aenean scelerisque convallis lacus at dapibus....

Récupération des éléments directement depuis l'API


Manipulation des éléments du DOM pour favoriser un affichage dynamique.

Création d'une boucle pour aller récupérer mes valeurs et les afficher.

Appel des fonctions dans une fonction asynchrone de façon à ce qu'à la lecture du script la fonction d'affichage des produits soit priorisée.

index.js

```
1 // Permet de communiqué avec le serveur
2 //On envoie une requête au serveur à l'URL /api/products pour récupérer la liste des produits
3 fetch("http://localhost:3000/api/products")
4 .then(function (res) {
5     //Si la requête est correcte on transforme les données en JSON
6     if (res.ok) {
7         return res.json();
8     }
9 })
10 //Puis on va exploiter la liste des canapés
11 .then(function (listeCanapes) {
12     //console.log(listeCanapes[0].name);
13     //On récupère l'élément HTML d'id items dans lequel on devra insérer tous les canapés de la liste précédemment récupérée
14     let section = document.getElementById("items");
15     //Pour chaque canapé
16     listeCanapes.forEach((canape, index) => {
17         //On crée un élément HTML a dans lequel on va insérer du HTML et des données sur le canapé en cours d'exploitation issues du serveur
18         let a = document.createElement("a");
19         a.innerHTML = " \
20         <article>\
21         <img src=\" \" + canape.imageUrl + \" \" alt=\" \" + canape.altTxt + \" \">\
22         <h3 class=\"productName\">\" + canape.name + \"</h3>\
23         <p class=\"productDescription\">\" + canape.description + \"</p>\
24         </article>\
25         ";
26         //On ajouter à l'élément HTML a une cible pour atteindre la page d'un canapé lorsqu'on clique dessus.
27         a.href = "./product.html?id="+canape._id;
28         //On ajouter cet élément à Items pour faire afficher le canapé
29         section.appendChild(a);
30     });
31 })
32 //Ceci est utilisé en cas d'erreur lors de la requête (si le serveur ne répond pas par exemple)
33 .catch(function (error) {
34     console.log("Erreur : " + error);
35 })
```

Kanap Sinopé

Prix : 1849€

Description :

Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Choisir une couleur

Nombre d'article(s)

✓ --SVP, choisissez une couleur --

Blue

White

Black

Affichage du produit avec possibilité de sélectionner une couleur et une quantité,

Création d’une quantité minimum de 1 pour l’input quantité.

```
product.js
1 //On va récupérer l'ID du canapé que l'on veut afficher à partir de l'URL
2 const queryString = window.location.search;
3 const urlParams = new URLSearchParams(queryString);
4 const id = urlParams.get("id");
5 //console.log("ID : " + id);
6
7 // Permet de communiqué avec le serveur
8 //On utilise une autre api avec l'id du canapé que l'on veut faire afficher pour récupérer toutes ses caractéristiques (couleur, prix, image ...)
9 fetch("http://localhost:3000/api/products/" + id)
10 .then(function (res) {
11     //Si la requête est correcte on transforme les données en JSON
12     if (res.ok) {
13         return res.json();
14     }
15 })
16 //On exploite les données du canapé pour les faire afficher
17 .then(function (canape) {
18     //console.log("Canape : " + canape.name);
19     //console.log("id canape " + canape._id);
20     //console.log("Prix : " + canape.price);
21     //console.log("URL image : " + canape.imageUrl);
22     //console.log("Alt text : " + canape.altTxt);
23     //console.log("Description : " + canape.description);
24     //On récupère chaque élément HTML que l'on veut modifier pour y ajouter l'une des données du canapé que l'on affiche, par exemple ci-dessous on récupère l'élément d'id titel pour y insérer le titre du canapé
25     let titre = document.getElementById("title");
26     titre.innerText = canape.name;
27     let prix = document.getElementById("price");
28     prix.innerText = canape.price;
29     let desc = document.getElementById("description");
30     desc.innerText = canape.description;
31     let image = document.getElementById("image");
32     image.src = canape.imageUrl;
33     image.alt = canape.altTxt;
34     let select = document.getElementById("colors");
35     //Pour chaque couleur du canapé on ajoute un élément HTML option dans le select, la liste déroulante.
36     canape.colors.forEach((couleur, index) => {
37         let option = document.createElement("option");
38         option.value=couleur;
39         option.innerText=couleur;
40         select.appendChild(option);
41     });
42 }
```

Page panier (confirmation)

Prénom:

Veuillez saisir un prénom valide, sans chiffres.

Nom:

Adresse:

Ville:

Email:

Veuillez saisir un email valide exemple: johndoe6@gmail.com.

```
const regexemail = /^(^[\w-\.\.]+\.[\w]{2,4})/;  
//Gestion de l'email  
let email = document.getElementById('email').value;  
if(!regexemail.test(email)){  
    erreur = true;  
    //alert("Vous devez entrer au minimum 3 lettres pour votre email.");  
}
```

```
const regexemail = /^(^[\w-\.\.]+\.[\w]{2,4})/;  
//Gestion de l'email  
let email = document.getElementById('email').value;  
if(!regexemail.test(email)){  
    erreur = true;  
    //alert("Vous devez entrer au minimum 3 lettres pour votre email.");  
}
```

Mise en place du contrôle des input saisis par l'utilisateur.

Utilisation du regex afin de filtrer les caractères.

Mise en place de message d'erreurs personnalisés en fonction de l'input et de son contenu.

Envoi de la confirmation au serveur

```
let listeIdCanape = [];  
panier.forEach((choix, index) => {  
  listeIdCanape.push(choix._id);  
});  
  
let message = {  
  contact: {  
    firstName: document.getElementById('firstName').value,  
    lastName: document.getElementById('lastName').value,  
    address: document.getElementById('address').value,  
    city: document.getElementById('city').value,  
    email: document.getElementById('email').value  
  },  
  products: listeIdCanape  
};  
console.log("Contenu de la variable message : "+ JSON.stringify(message));  
localStorage.removeItem("confirmationCommande");  
fetch("http://localhost:3000/api/products/order", {  
  method: "POST",  
  headers: {  
    'Accept': 'application/json, text/plain, */*',  
    'Content-Type': 'application/json'  
  },  
  body: JSON.stringify(message)  
})  
then(function (res) {  
  return res.json().then(function (data) {  
    console.log(data);  
  });  
});
```

Création de la liste des id de canapé en prévision de l'envoi au serveur

Création du message pour le serveur avec :

- L'objet contact rempli à partir du formulaire
- La liste des id de canapé

Envoi du message POST au serveur avec un entête personnalisé

Réception de la confirmation du serveur

```
fetch("http://localhost:3000/api/products/order", {
  method: "POST",
  headers: {
    'Accept': 'application/json, text/plain, */*',
    'Content-Type': 'application/json'
  },
  body: JSON.stringify(message)
})
.then(function (res) {
  if (res.ok) {
    return res.json();
  }
})
.then(function (confirmationCommande) {
  console.log("Confirmation de commande : " + JSON.stringify(confirmationCommande));
  localStorage.setItem("confirmationCommande", JSON.stringify(confirmationCommande));
  window.location.href="confirmation.html";
})
.catch(function (error) {
  console.log("Erreur : " + error);
});

});
```

Une fois le message envoyer au serveur on attend la réponse

Si la réponse est correcte on exploite celle-ci

On récupère la confirmation commande et on la stocke dans le Localstorage

Pour finir on redirige l'utilisateur vers la page de confirmation

Confirmation de commande

```
1 // Confirmation de la commande a partir du stockage local
2 let confirmationCommande = localStorage.getItem("confirmationCommande");
3 confirmationCommande = JSON.parse(confirmationCommande);
4 console.log("OrderId : " +confirmationCommande.orderId);
5 let OrderId = document.getElementById("orderId");
6 OrderId.innerText = confirmationCommande.orderId;
7 localStorage.removeItem("panier");
```

On récupère la confirmation du Localstorage

On affiche cette confirmation dans l'élément HTML portant l'id « orderId »

On supprime le panier en vu d'une prochaine utilisation du site