MVP – Spécifications fonctionnelles et techniques du site de Kanap

SOMMAIRE

Architecture générale

L’application web sera composée de 4 pages :

- Une page d’accueil montrant (de manière dynamique) tous les articles disponibles à la vente. **ok**
- Une page “produit” qui affiche (de manière dynamique) les détails du produit sur lequel l'utilisateur a cliqué depuis la page d’accueil. Depuis cette page, l’utilisateur peut sélectionner une quantité, une couleur, et ajouter le produit à son panier. **ok**
- Une page “panier”. Celle-ci contient plusieurs parties :
    - Un résumé des produits dans le panier, le prix total et la possibilité de modifier la quantité d’un produit sélectionné ou bien de supprimer celui-ci.
    - Un formulaire permettant de passer une commande. Les données du formulaire doivent être correctes et bien formatées avant d'être renvoyées au back-end. Par exemple, pas de chiffre dans un champ prénom.
- Une page “confirmation” :
    - Un message de confirmation de commande, remerciant l'utilisateur pour sa commande, et indiquant l'identifiant de commande envoyé par l’API.

Planification de tests

Planifiez une suite de tests d’acceptation pour couvrir l’ensemble des fonctionnalités listées dans ce document (spécifications fonctionnelles et techniques Kanap).

Voici le modèle à partir duquel écrire ce plan : Modèle de plan de test.

Informations complémentaires

La page d’accueil

Cette page présente l’ensemble des produits retournés par l’API.
Pour chaque produit, il faudra afficher l’image de celui-ci, ainsi que son nom et le début de sa description.
En cliquant sur le produit, l’utilisateur sera redirigé sur la page du produit pour consulter celui-ci plus en détail.

La page Produit

Cette page présente un seul produit ; elle aura un menu déroulant permettant à l'utilisateur de choisir une option de personnalisation, ainsi qu’un input pour saisir la quantité. Ces éléments doivent être pris en compte dans le panier.

La page Panier

Sur cette page, l'utilisateur va pouvoir modifier la quantité d'un produit de son panier ; à ce moment, le total du panier devra bien se mettre à jour. ok
L'utilisateur aura aussi la possibilité de supprimer un produit de son panier, le produit devra donc disparaître de la page. ok
Les inputs des utilisateurs doivent être analysés et validés pour vérifier le format et le type de données avant l'envoi à l'API. Il ne serait par exemple pas recevable d'accepter un prénom contenant des chiffres, ou une adresse e-mail ne contenant pas de symbole "@". En cas de problème de saisie, un message d'erreur devra être affiché en dessous du champ correspondant. ok

La page Confirmation

Sur cette page, l'utilisateur doit voir s'afficher son numéro de commande. Il faudra veiller à ce que ce numéro ne soit stocké nulle part.

Le code source ok

Celui-ci devra être indenté et utiliser des commentaires en début de chaque fonction pour décrire son rôle. Il devra également être découpé en plusieurs fonctions réutilisables (nommées). Une fonction doit être courte et répondre à un besoin précis. Il ne faudrait pas avoir de longues fonctions qui viendraient répondre à plusieurs besoins à la fois. Exemple : il ne serait pas accepté de mettre une seule et unique fonction en place pour collecter, traiter et envoyer des données. ok

API

Concernant l'API, des promesses devront être utilisées pour éviter les callbacks. Il est possible d'utiliser des solutions alternatives, comme fetch, celle-ci englobant la promesse. L'API n'est actuellement que dans sa première version. La requête post qu'il faudra formuler pour passer une commande ne prend pas encore en considération la quantité ni la couleur des produits achetés. ok

Fonctionnement du panier

Dans le panier, les produits doivent toujours apparaître de manière regroupée par modèle et par couleur. ok
Si un produit est ajouté dans le panier à plusieurs reprises, avec la même couleur, celui-ci ne doit apparaître qu'une seule fois, mais avec le nombre d'exemplaires ajusté. ok
Si un produit est ajouté dans le panier à plusieurs reprises, mais avec des couleurs différentes, il doit apparaître en deux lignes distinctes avec la couleur et la quantité correspondantes indiquées à chaque fois. ok

Types de données

Tous les produits possèdent les attributs suivants :

Champ	**Type**
colors	array of string
id	string
name	string
price	number
imageUrl	string
description	string
altTxt	string

Technologies utilisées

HTML, CSS, JavaScript.

URL des API

- Catalogue de canapés : http://localhost:3000/api/products

Paramètres des API

Chaque API contient 3 paramètres :

Verbe	Paramètre	Corps de la demande prévue	Réponse
GET	/	-	Retourne un tableau de tous les éléments
GET	/{product-ID} {product-ID} doit être remplacé par l'id d'un produit	-	Renvoie l'élément correspondant à {product-ID}, identifiant d'un produit
POST	/order	Requête JSON contenant un objet de contact et un tableau de produits	Retourne l'objet contact, le tableau produits et orderId (string)

Validation des données

ok Pour les routes POST, l'objet contact envoyé au serveur doit contenir les champs firstName,
ok lastName, address, city et email. Le tableau des produits envoyé au back-end doit être un
ok array de strings product-ID. Les types de ces champs et leur présence doivent être validés avant l'envoi des données au serveur.