MVP - Spécifications fonctionnelles et techniques ORINOCO

SOMMAIRE

Architecture générale

L’application web sera composée de 4 pages :

- une page de vue sous forme de liste, montrant tous les articles disponibles à la vente ;
- une page “produit”, qui affiche de manière dynamique l'élément sélectionné par l'utilisateur et lui permet de personnaliser le produit et de l'ajouter à son panier ;
- une page “panier” contenant un résumé des produits dans le panier, le prix total et 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 texte dans les champs date ;
- une page de confirmation de commande, remerciant l'utilisateur pour sa commande, et indiquant le prix total et l'identifiant de commande envoyé par le serveur.

Produits présentés

Dans un premier temps, une seule catégorie de produits sera présentée.

Choix à faire entre les 3 propositions suivantes :

- ours en peluche faits à la main ;
- caméras vintage ;
- meubles en chêne.

Planification de tests unitaires

Planifiez une suite de tests unitaires pour couvrir au minimum 80 % de la base de code pour le front-end. Vous devrez formaliser un plan pour atteindre ce résultat, sans obligation d’écrire ces tests Expliquez quelles lignes seront testées, et quels “test cases” seront envisagés.

Informations complémentaires

Pour le MVP, la personnalisation du produit ne sera pas fonctionnelle : la page contenant un seul article aura un menu déroulant permettant à l'utilisateur de

choisir une option de personnalisation, mais celle-ci ne sera ni envoyée au serveur ni reflétée dans la réponse du serveur.

Le code source devra être indenté et utiliser des commentaires. Il devra également utiliser des fonctions globales.

Concernant l'API, des promesses devront être utilisées pour éviter les rappels.

Les inputs des utilisateurs doivent être validés avant l'envoi à l'API.

Types de données

Tous les produits possèdent les attributs suivants :

Champ	**Type**
id	ObjectID
name	string
price	number
description	string
imageUrl	string

Chaque type de produit comporte un tableau contenant les strings correspondant aux options de personnalisation :

Type de produit	**Tableau de personnalisation**
Caméras	lentilles
Ours en peluche	couleurs
Meubles en chêne	vernis

Technologies utilisées

HTML, CSS, JavaScript.

URL des API

- Ours en peluche faits à la main : http://localhost:3000/api/teddies
- Caméras vintage : http://localhost:3000/api/cameras
- Meubles en chêne : http://localhost:3000/api/furniture

Paramètres des API

Chaque API contient 3 paramètres :

Verb	**Paramètre**	**Corps de la demande prévue**	**Réponse**
GET	/	-	Retourne un tableau de tous les éléments
GET	/:_id	-	Renvoie l'élément correspondant à identifiant given_id
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

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

Repository GitHub à cloner

https://github.com/OpenClassrooms-Student-Center/JWDP5.git