

PLAN DE TEST - KANAP

15/7/2022 - LAURA FROMENTIN

	Fonctionnalité	Action	Résultat attendu	Résultat observé
1	Une page d'accueil montrant (de manière dynamique) tous les articles disponibles à la vente.	Ouvrir sur la page d'accueil du site web dans un navigateur	Affichage de l'ensemble des produits	OK / Description erreur
PAGE PRODUIT				
2	Etre dirigé sur la bonne page correspondante au produit sélectionné.	Insérer des liens dynamiques sur l'image et le nom du produit avec le paramètre d'url de l'id	Lien actif de chaque article vers sa page produit correspondante	OK
3	Faire apparaître le détail d'un article sur sa page produit.	Récupérer les paramètres de requête du produit, ses caractéristiques depuis l'API, les données HTML pour la mise en forme et tout faire concorder	Affichage du détail du produit	OK
4	Récupérer les options "couleurs".	Créer une boucle pour récupérer les données de l'API qu'on va intégrer dans le HTML avec .innerHTML	Affichage des coloris disponibles pour chaque produit	OK
5	Vérifier que l'utilisateur a bien sélectionné une couleur.	Créer condition avec if si le champ color n'est pas nul	Affichage d'une alerte si valeur non sélectionnée	OK
6	Vérifier que l'utilisateur a bien sélectionné une quantité.	Créer condition avec if si le champ quantité n'est pas nul	Affichage d'une alerte si valeur nulle et procédure de commande bloquée.	OK
7	Vérifier que la quantité demandée est entre 1 et 100.	Créer une condition	Affichage d'une alerte si la quantité demandée ne respecte pas la condition et procédure de commande bloquée.	
7	Activer le bouton "ajouter au panier".	Créer un "click event", une alerte de confirmation et envoyer le produit dans le local storage.	Affichage d'une bulle popup dirigeant l'utilisateur à continuer son shopping ou à se diriger directement vers la page panier.	OK
PAGE PANIER				
8	Faire apparaître toutes les infos utiles de l'article dans le panier.	Récupérer les infos du LS et de l'API car pour des raisons de sécurité, le prix ne doit pas être stocké dans le LS.	Affichage de tous les produits sélectionnés ainsi que le détail (photo, prix, couleur, quantité).	OK
9	Ne pas dupliquer les lignes si même produit et même coloris.	Récupérer l'ID et la couleur sélectionnée et comparer avec .find	Quantité incrémentée sur produit si déjà existant dans le panier et/ou nouvelle ligne de produit générée si produit nouveau dans le panier.	OK
10	Modifier la quantité de l'article directement depuis le panier.	Récupérer la valeur initialement demandée depuis le LS et s'assurer que la valeur modifiée souhaitée ne dépasse pas la quantité maximale autorisée.	Affichage d'une alerte informant que la quantité a été mise à jour sinon quantité réinitialisée à la quantité initiale si la demande n'est pas conforme.	OK
11	Afficher le montant total du panier automatiquement.	Récupérer la liste des produits sélectionnés, leur prix et leur quantité et réunir le tout dans une fonction asynchrone.	Calcul automatique et instantané lorsqu'on rajoute ou qu'on supprime un produit directement depuis le panier.	OK
12	Supprimer le bon article.	Récupérer les infos du LS et de l'API soit l'ID et le coloris afin de ne supprimer qu'un seul article et dynamiser le bouton supprimer.	Affichage d'une alerte informant la bonne suppression de l'article + mise à jour automatique du panier.	OK
13	Elaborer un formulaire intelligent.	Utiliser la méthode regex et des alertes.	Les caractères inadéquats sont refusés et des messages d'alerte avec exemple de réponse attendue apparaissent.	OK
14	Bloquer le processus de commande si panier vide.	Créer une condition pour bloquer le processus puis message d'alerte.	Le formulaire n'est même pas pris en compte tant que le panier est vide, un message d'alerte apparaît et invite l'utilisateur à remplir son panier.	
15	Diriger l'utilisateur vers la page commande.	Utiliser la méthode POST et ne pas stocker le n° de commande dans le local storage.	Affichage de la page commande avec le numéro de commande.	OK