

	Fonctionnalité	Action	Résultat attendu	Résultat observé
	La page d'accueil			
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
2	Cette page présente l'ensemble des produits retournés par l'API.	Interroger l'api pour obtenir la liste des produits	récupération des produits	OK
3	Pour chaque produit, il faudra afficher l'image de celui-ci, ainsi que son nom et le début de sa description	afficher par produit les informations retournées par l'api et les affiche dans la page selon le frontend fourni	affichage des produits et de leurs informations relatives	OK
4	En cliquant sur le produit, l'utilisateur sera redirigé sur la page du produit pour consulter celui-ci plus en détail.	renvoyer vers le produit ciblé par le click	renvoi vers la page produit pour afficher celui choisi	OK
5	La page Produit			
6	Cette page présente un seul produit	interroger l'api pour obtenir le produit sélectionné par son id	affichage du produit sélectionné par son id et de ses informations relatives selon le schéma du frontend	OK
7	elle aura un menu déroulant permettant à l'utilisateur de choisir une option de personnalisation	afficher une liste déroulante avec les options de couleurs du produit en court	la liste déroulante affiche bien les options de couleurs du produit; la couleur choisie devra être envoyée au panier	OK
8	elle aura un input pour saisir la quantité.	avoir la possibilité de choisir une quantité (entre 1 et 100) du produit	une input de type number est fourni dans le frontend. Elle est utilisée pour obtenir la quantité à envoyer au panier. quantité limitée de 1 à 100	OK
9	Ces éléments (option de couleurs et quantité) doivent être pris en compte dans le panier.	enregistrer l'id, les options de couleurs et la quantité choisies dans le localStorage via le bouton " ajouter au panier".	enregistrement de l'id, de la couleur et de la quantité dans le panier, dans le localStorage	OK
10	La page Panier			
11	L'utilisateur va pouvoir modifier la quantité d'un produit de son panier	modifier la quantité de produit(s) que l'utilisateur souhaite acheter	Une input de type number est fournie par le frontend. La quantité est modifiable et doit bien être enregistrée dans le localStorage	OK
12	le total du panier devra bien se mettre à jour.	ajuster le nombre d'articles et le prix total dynamiquement lorsque l'utilisateur modifie la quantité des produits	ajustement dynamique du nombre et du prix total	OK
13	L'utilisateur aura aussi la possibilité de supprimer un produit de son panier	donner la possibilité à l'utilisateur de supprimer un article	écoute le clic sur le bouton supprimer pour exécuter la suppression	OK
14	le produit devra donc disparaître de la page.	enlever le produit de la page et du LocalStorage	lorsque l'utilisateur supprime un article, il disparaît de la page et est supprimé du LocalStorage	OK
15	La page Confirmation			
16	utilisateur doit voir s'afficher son numéro de commande.	ouvrir la page confirmation et affiche le numéro de commande	récupération et affichage du numéro de commande retourné par l'api.	OK
17	Il faudra veiller à ce que ce numéro ne soit stocké nulle part.	ne pas être stocker le numéro de commande	ne pas stocker le numéro de commande et "nettoyer" le panier	OK
18	Le code source			
19	Celui-ci devra être indenté et utiliser des commentaires en début de chaque fonction pour décrire son rôle.	indenter et commenter le code source	le code source est indenté et commenté	OK
20	Il devra également être découpé en plusieurs fonctions réutilisables (nommées).	découper les fonctions, les rendre réutilisables et les nommer	les fonctions sont découpées réutilisables et nommées	OK
21	Une fonction doit être courte et répondre à un besoin précis.	faire des fonctions courtes qui ne répondent qu'à un besoin précis	les fonctions sont courtes et ne répondent qu'à un besoin précis	OK
22	API			
23	des promesses devront être utilisées pour éviter les callbacks. Il est possible d'utiliser des solutions alternatives, comme fetch, celle-ci englobant la promesse.	utiliser des promesses et éventuellement fetch	fetch est utilisé et le callback Hell est évité	OK
24	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.	prendre connaissance du fonctionnement de l'api pour exécuter les requêtes et interpréter les résultats retournés correctement	les requêtes sont structurée selon le fonctionnement de l'api	OK
25	Fonctionnement du panier			
26	Dans le panier, les produits doivent toujours apparaître de manière regroupée par modèle et par couleur.	regrouper les produits par id et par couleur	le articles sont regroupés par id et couleur	OK
27	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.	ne faire apparaître un produit strictement identique (id et couleur) qu'une fois dans le localStorage	modification dynamique du panier pour ne pas avoir de doublon (même produit/couleur, quantité différente)	OK
28	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.	trier le panier pour afficher les produit de même id mais de couleur différentes	la méthode ".sort" tri le panier par id	OK