

	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 (chrome et firefox testés)	- Affichage de l'ensemble des produits => données récupérées via méthode fetch et boucle for pour ajouter contenu HTML en fonction du nombre de produits dans le catalogue	Affichage dynamique ok - test console.log => pour vérifier si les données retournées par l'API s'affiche correctement - si erreur lors de la requête fetch => un message contenant l'erreur s'affiche à l'écran (utilisation fonction catch)
2	La page d'accueil devra afficher les informations suivantes pour chaque produit: le nom, l'image, le début de la description	Possible de cliquer sur les fiches produits	les fiches produits doivent contenir les informations suivantes : nom, image, description	- affichage des informations ok - description alternatives, et ID du produits aussi récupérées et utilisées
3	En cliquant sur le produit, l'utilisateur est redirigé vers la page produit de celui-ci	suite au clique sur la fiche produit, redirection vers la page de celui-ci	page produit s'affiche avec l'image du produit, nom, prix, description	affichage correspondant à l'ID récupéré dans l'URL. test console effectué pour la récupération de de l'ID et pour vérifier si objet récupéré correspond
4	la page produit doit contenir: un menu déroulant pour choisir la couleur, un input pour saisir la quantité	utilisateur peut sélectionner couleur et quantité du produit avant de l'ajouter au panier	- après sélection couleur et quantité, celles si sont enregistrés dans des variables après vérification - Les variables vont servir pour créer la ligne produit dans le panier	- sélection couleur et quantité opérationnel - gestion des exceptions mise en place pour choix de la quantité => doit être un nombre entre 1 et 100 sinon message d'erreur - test console pour voir la valeur des variables
5	Les éléments choisis dans la page produits doivent se retrouver dans le panier via un bouton "ajout au panier"	utilisateur clique sur le bouton pour ajouter le produit à panier	- si données valident, le produit est ajouté au panier - si le produit est déjà dans le panier, et si couleur identique on modifie uniquement la quantité	- produit enregistré dans le localStorage - gestion des exceptions au moment ajout au panier - si produit déjà enregistré, demande de confirmation à l'utilisateur pour modifier la quantité
6	La page panier doit contenir un résumé des produits ajoutés au panier (avec quantité, image, couleur et prix unitaire)	utilisateur peut visualiser le résumé de son panier et avoir accès à la suppression et modification de chaque articles	une ligne par modèle et couleur + total articles dans le panier + prix total + affichage du formulaire et du bouton commander	affichage ok => une ligne par modèle et couleur (les produits sont gérés via une clé Id//couleur dans le localStorage) 2 fonctions pour calculer la quantité totale et le prix total
7	Dans la page panier, l'utilisateur doit pouvoir modifier la quantité par produit et le panier doit se mettre à jour	dans le champ input, l'utilisateur peut modifier manuellement la quantité	après modification et vérification de la quantité, la page est actualisée et les totaux sont mis à jour	- modification et actualisation opérationnelle - gestion des exceptions mises en place pour tester la validité de la saisie utilisateur - confirmation demandée à l'utilisateur pour enregistrer la modification - test console pour vérifier le script

	Fonctionnalité	Action	Résultat attendu	Résultat observé
8	Dans la page panier, l'utilisateur doit pouvoir supprimer un produit et le panier doit se mettre à jour	l'utilisateur peut supprimer un article en cliquant sur le bouton supprimer	après suppression d'un article, la page est actualisée et les totaux sont mis à jour	<ul style="list-style-type: none"> - suppression opérationnelle - article supprimé de la page et du localStorage - confirmation demandée à l'utilisateur pour enregistrer la suppression - test console pour vérifier le script
9	Dans la page panier, un formulaire doit être rempli pour passer la commande	l'utilisateur doit remplir chaque champ du formulaire pour valider commande	si les champs du formulaire ne sont pas remplis => des messages d'erreur apparaissent sous les champs non valides et la commande n'est pas validée	<ul style="list-style-type: none"> - validation mise en place au moment de la soumission de la commande - chaque champ est testé individuellement et renvoie soit un message d'erreur, soit valide - si un champ n'est pas valide, la commande n'est pas validée.
10	les données du formulaire doivent être analysées et validées avant d'être envoyées au back-end	l'utilisateur doit remplir chaque champ du formulaire pour valider commande	si les champs du formulaire ne sont pas remplis => des messages d'erreur apparaissent sous les champs non valides et la commande n'est pas validée	<ul style="list-style-type: none"> - validation mise en place au moment de la saisie d'un champ - chaque champ est testé individuellement et renvoie soit un message d'erreur, soit valide - si un champ n'est pas valide, l'utilisateur peut le voir tout de suite et corriger
11	le traitement des données du formulaire doit passer par plusieurs fonctions en Javascript. Chacune devant répondre à un besoin précis	une fonction de test et de validation par champ	chaque champ a ses propres règles de validation en fonction de sa nature	<ul style="list-style-type: none"> - par défaut, la validation du navigateur est utilisée dans le message d'erreur - le test des champs se fait à l'aide de regex personnalisée
12	Une fois la commande validée, le numéro de commande devra s'afficher sur la page confirmation et ne devra pas être stockée	une fois formulaire valide, l'utilisateur clique sur le bouton commander pour finaliser la commande	la page confirmation s'ouvre et affiche le numéro de commande. Le numéro de commande n'est pas stocké en local	<ul style="list-style-type: none"> - page et numéro de commande s'affiche correctement - gestion réponse de l'API avec promesse et un try & catch - transmission et récupération du numéro de commande via l'URL de la page confirmation