

Nr	Localisation	Action	Attendu	Résultat
1	Index.js Index.html	Au chargement de la page index.html, vérification que la requête fetch() sur <a href="http://localhost:3000/api/cameras">http://localhost:3000/api/cameras</a> renvoi bien l'ensemble des données articles	L'ensemble des images et informations de de tous les produits doivent apparaitre sur la page index.html	OK
2	Index.js Index.html	Vérification de la réponse erreur catch(). Réalisation d'une requête avec le serveur fermé	Affichage de la réponse catch() : "Il y a eu un problème avec l'opération fetch: Failed to fetch" dans la console.	OK
3	detail.js produit.html	On clique sur le bouton "voir le détail du produit" des différents produits présents sur index.html	Affichage du détail de l'article sélectionné avec l'id article présent dans l'url de la page detail.html	OK
4	detail.js produit.html	Au chargement de la page detail.html vérification que la requête fetch() sur <a href="http://localhost:3000/api/cameras">http://localhost:3000/api/cameras</a> + "/" + idByArticle renvoi bien toutes les données de l'article	La page produit.html doit afficher le détail du produit sélectionné avec : image, nom, id, descriptif, prix et options lentilles	OK
5	detail.js produit.html	Vérification de la présence de l'ensemble des lentilles en option (1, 2 ou 3 lentilles)	Le menu déroulant doit afficher toutes les lentilles sur tous les articles	OK
6	detail.js produit.html	Vérification de la création d'un localStorage à l'ajout d'un produit si le panier est vide.	Dans application/Local storage/file affichage type : 0: {id: "5be1ef211c9d44000030b062", name: "Hirsch 400DTS", price: " 3099", no: 1}	OK
7	detail.js produit.html	Vérification de l'ajout d'un nouvel article dans le localStorage si un autre article est déjà présent.	Dans application/Local storage/file affichage type : 0: {id: "5be1ef211c9d44000030b062", name: "Hirsch 400DTS", price: " 3099", no: 1} 1: {id: "5be9bc241c9d440000a730e7", name: "Franck JS 105", price: " 2099", no: 1}	OK
8	detail.js produit.html	Vérification de l'augmentation des quantités articles à l'ajout d'un article déjà présent dans le local storage.	Dans application/Local storage/file affichage type : 0: {id: "5be1ef211c9d44000030b062", name: "Hirsch 400DTS", price: " 3099", no: 1} 1: {id: "5be9bc241c9d440000a730e7", name: "Franck JS 105", price: " 2099", no: 3}	OK

9	panier.js panier.html	Vérification du remplissage du panier conformément au local storage	Le panier affiche les mêmes produits que le local storage	OK
10	panier.js panier.html	Vérification que le format (JSON) des données présentes dans le panier permet bien d'additionner les prix et quantités entre elles et que le total est exact.	Total prix et quantités correspondent aux articles présents et sont recalculé à chaque fois	OK
11	panier.js panier.html	La fonction Delete permet de supprimer une ligne du panier et permet un bon re-calcul du total panier	La ligne disparaît du panier, le nouveau total doit être exact. La quantité d'articles à côté de l'icône panier (entête) doit correspondre également.	OK
12	panier.js panier.html	Saisie de différents caractères dans les zones de saisie : uniquement chiffres, uniquement lettres, majuscules, email sans @,...	Le formulaire de contact n'accepte que le type de données validés par le Regex. Affichage d'un message d'erreur sous le champ à la fin de la saisie et blocage de la requête POST si saisie incorrecte.	OK
13	panier.js panier.html	Vérification que localStorage.clear() à la fin de la requête POST vide bien le panier et le local storage.	LocalStorage et panier doivent être vides	OK
14	confirmation.js confirmation.html	La requête POST renvoie un objet comprenant {contact, products : [], orderId } Après envoi de la requête, la page confirmation.html doit s'ouvrir et affiche un message avec le prénom du contact ainsi que la référence de commande.	Le prénom du contact saisi dans le formulaire doit s'afficher ainsi qu'un numéro de commande généré par l'api serveur	OK