

Nr	Localisation	Action	Attendu	Résultat
1	Index.js	Faire une requête fetch() sur l'adresse " <u>http://localhost:3000/api/cameras</u> "	Objet JSON avec les différents articles classé par index	OK
2	Index.js	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"	OK
3	detail.js	Affichage de l'id article sélectionné présent dans l'URL de la page détail.html avec : <ul style="list-style-type: none"> const urlID = Window.location.search; const urlParams =new URLSearchParams(urlID); Const idByArticle = urlParam.get('id'); 	Affichage de l'id article sans espace et sans autre caractère	OK
4	detail.js	Faire une requête fetch() sur l'adresse <u>http://localhost:3000/api/cameras</u> + "/" + idByArticle	Objet JSON de l'article sélectionné	OK
5	detail.js	Affichage des options lentilles dans la requête avec : data.lenses[i]	Affichage des options lentille suivant index	OK
6	detail.js	Vérification de la création d'un local storage 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	Vérification de la l'ajout d'un nouvel article dans le local storage 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	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	Le panier affiche aucun article et aucune quantité lorsque l'on n'a pas encore sélectionné de produit et que l'on supprime toutes les lignes du panier	Le tableau est vide	OK
10	panier.js	Le panier affiche bien les produits, quantités et prix correspondant au localStorage, le format JSON du prix et quantités permet d'additionner et multiplier les valeurs entre elles	Total prix et quantité correspondant aux articles présents	OK
11	panier.js	La fonction delete permet bien de supprimer une ligne dans le panier et retire les quantités et prix du total et de l'icône "panier" de l'entête	La ligne disparaît et les quantités+prix correspondant sont retirés	OK
12	panier.js	Le formulaire de contact n'accepte que le type de donnée "required" par son "type"	@ doit être présent sans email	OK
13	panier.js	La requête POST renvoie bien la liste des articles sélectionnés et un numéro de commande dans un objet JSON	Objet comprenant {contact, products : [], orderId }	OK
14	panier.js	Après la réception du résultat de la requête, le localStorage est vidé avec localStorage.clear()	LocalStorage et panier sont vides	OK
15	Confirmation.js	Après réception du résultat de la requête, la page confirmation.html doit s'ouvrir et affiche le prénom du contact ainsi que la référence de commande.	Message avec prénom du contact et le numéro de commande	OK