

## Plan de Tests

Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
orinoco.js	Ligne 4 à 21	Récupérer la liste des produit du serveur par l'API : <code>async function loadCameras() { const response = await fetch</code> appelée au début du fichier doivent s'afficher convenablement dans la page de listes des produits	La fonction asynchrone <code>fetch()</code> lance une requête et renvoie une promesse, mais puisqu'il y a <code>await</code> , la fonction asynchrone sera suspendue jusqu'à la finition de la demande pour faire l'affichage	<code>catch(e) { console.log(e); }</code>	En cas de problème de réseau par exemple, il se peut que la liste d'éléments récupérés et a afficher soit nulle.	Voir ✓
orinoco.js	Ligne 31 à 33	Fonction permettant d'afficher la liste de produit récupérée par la fonction précédente <code>fetch</code> du serveur avec une boucle <code>forEach</code> pour chaque élément: <code>cameras.forEach((camera) =&gt; { mainContainer.innerHTML += buildCamera(new Camera(camera)); })</code>	Tous les 5 produits récupérés par la fonction <code>async function loadCameras() { const response = await fetch</code> appelée au début du fichier doivent s'afficher convenablement dans la page de listes des produits	Se positionner dans le répertoire back-end et appeler le serveur avec la commande <code>node server</code> , puis démarrer la page <code>products-list.html</code>	Le nombre d'élément du DOM peut être Différent du nombre attendu, voir inexistant., surtout si le serveur est déconnecté ou rencontre un problème de communication	Voir ✓
Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
product.js	Ligne 6 à 9	PRODUIT SÉLECTIONNÉ PAR _id DE L'API : <code>function getCamera() { const cameraId = new URL(location.href).searchParams.get("id"); const url = `http://localhost:3000/api/cameras/\${cameraId}`; fetch(url)</code>	Récupère les données de la caméra sélectionnée avec l'API Fetch, / _id de la caméra : Création du lien pour afficher la caméra sélectionnée ; La caméra sélectionnée apparaît sur la Page	REF. au dossier backend/routes/camera.js -> <code>router.get('/:id', cameraCtrl.getOneCamera)</code> ; Le lien fonctionne et nous envoie vers la caméra sélectionnée	« id » absent du « backend » ou objet Vide : Le lien ne fonctionne pas ou la caméra ne correspond pas avec celle qui a été sélectionnée	Voir ✓
product.js	Ligne 22 à 26 Et Ligne 46 à 53	Récupération des données de la caméra et injection du html depuis le js grâce au <code>innerHTML</code> : <code>let mainContainerProduct = document.getElementById('cardProduct'); mainContainerProduct.innerHTML = buildCameraProduct(selectedCamera); buildCameraLensesSelectList(selectedCamera.lenses);</code>	Chaque caméra a bien sa page avec ses propres données (nom, prix, image, options de lentilles et sa description)	<code>console.log(camera);</code>	Absence « key : value »  La valeur retournée pourrait être indéfini, ou La valeur ajouté pourrait ne pas être prise en compte	Voir ✓
product.js	Ligne 32 à 38 Et Ligne 54	ÉCOUTE DU BOUTON ET AJOUT DE LA SÉLECTION AU PANIER :  <code>const btn_addCart = document.querySelector("#add_product_to_basket");</code>	Récupération des données API correspondant a l'ID dans une variable produit. L'objet CAMERA stocke les valeurs.	Tester le bouton Ajouter au panier ou appeler la fonction en omettant de Le produit pourrait ne pas être ajouté au sélectionner une option	Le produit pourrait ne pas être ajouté au sélectionner une option localStorage.	Voir ✓
product.js	Ligne 39 à 40	ÉCOUTE DE L'EVENEMENT « CLICK » AVEC ENVOI DU CHOIX A LA PAGE PANIER / LA PRISE EN COMPTE DU TYPE DE LENTILLE CHOISIE : <code>let selectLenses = document.getElementById("cameraLenses"); console.log('Lense sélectionné == &gt; ' + selectLenses.value);</code>	Affichage DE L'ALERTE : Vous Avez ajouté le produit dans Votre panier ! AVEC Affichage dans la page HTML du menu déroulant des lentilles	<code>console.log('Lense sélectionné == &gt; ' + selectLenses.value);</code>	Aucune lentille n'a été sélectionnée (alert), la lentille ne correspond pas, ou la page ne s'ouvre pas ou les données ne correspondent pas ou la caméra n'apparaît pas	Voir ✓

Fichiers	Lignes	Fonctions Testées	Résultat attendu	Comment vérifier le résultat attendu	Problème possible	Validé
basket.js	Ligne 8 à 27	Récupération des données du panier du Local Storage et les dispatcher dans le HTML <code>Const basketRecovery = JSON.parse(localStorage.getItem("myBasket"));</code>	La fonction récupère le dernier élément du local storage et regroupe les mêmes cameras par leur ID, puis les affiche dans le Panier.html (basket)	Il faut carrément choisir deux ou trois produit et voir si nous les retrouvons sur la page panier (basket.html)	Les informations sont erronées ou le tableau est vide	Voir ✓
basket.js	Ligne 29 à 31	Affichage du prix total de la commande dans le panier : <code>const totalPriceDisplay = `&lt;div id="total-price" class="total-cart-price"&gt; Le prix total de vos achats est de : \${totalPrice}.00 €&lt;/div&gt;`</code>	Le prix total de la commande doit s'afficher sur la page HTML du Panier (Basket)	il faut jouer sur la quantité en choisissant un produit, et voir si le total se calcule et s'affiche bien sur la page du panier	La valeur retournée pourrait ne pas correspondre au total attendu	Voir ✓
basket.js	Ligne 34 à 39	insertion des produits récupérés du local storage avec ajout du prix total : <code>basketDisplay.innerHTML = structureProductBasket; divTotalPrice.innerHTML = totalPriceDisplay; localStorage.setItem('totalCommand', totalPrice);</code>	Les produits choisis ainsi que le prix total de ces produits doivent tous s'afficher sur la page du panier	il faut choisir différents produits (minimum 2) avec des quantités supérieures à 2 et avec différentes lentilles pour voir si le tout s'affiche bien sur la page du panier	La fonction pourrait ne rien récupérer du local Storage et la valeur retournée serait nulle	Voir ✓
basket.js	Ligne 44 à 47	Fonction pour supprimer l'article de la commande, avec affichage d'un message de confirmation de suppression de l'article : <code>function deleteItem(event, itemId) { event.preventDefault(); let choice = confirm("Voulez-vous retirer l'article du panier ?");</code>	La fonction permet de vider le panier, afficher un message 'votre panier est vide' et redirection vers la page d'accueil, et si le panier contient tjrs des items sélectionnés, il faut un reload du panier pour afficher ces derniers.	Il faut Tester l'écouteur d'événement	Le panier pourrait ne pas se vider	Voir ✓
basket.js	Ligne 60 à 66	Fonction pour obtenir un Id de la commande : <code>function getOrderid(respondId) { let orderId = respondId.orderId; console.log(orderId);</code>	Fonction pour obtenir un Id de commande / au clic du bouton, on arrive sur la page de confirmation affichant un numéro de commande aléatoire	Modifier les valeurs de totalCommand() puis la valeur de la variable numeroCde et appeler la fonction	La valeur de retour pourrait ne pas être prise en compte	Voir ✓
basket.js	Ligne 69 à 116	Fonction pour gérer la soumission du formulaire : Récupération des données saisie dans le formulaire, récupère les id des produits et créer le contact = créer la commande avec le contact de cette dernière : <code>async function sendCommandToServer(order) { const config = { method: 'POST', headers: { 'Accept': 'application/json', 'Content-Type': 'application/json' }, body: JSON.stringify(order) } Puis return await response.json();</code>	Acceptation des caractères saisies dans le formulaire, et redirection vers la page de confirmation avec un message de validation, et un numéro de commande.	Jouer sur plusieurs type de quantité, de caméra, de lentille et de contact, pour voir la différence dans l'affichage du résultat.	"data" inexistante.	Voir ✓
confirmation.js	Ligne 02 à 29	La fonction permet d'afficher l'identifiant et le total de la commande tout en supprimant le contenu du local storage	Affichage du message de confirmation de la commande, avec un numéro de commande, et le total de cette dernière, tout en supprimant le panier, c'est à dire le local storage	c'est simple, il faut juste passer commande	La valeur retournée pourrait ne pas correspondre au résultat attendu	Voir ✓