

PLAN DE TEST

Objectif du plan de test

Tester les fonctionnalités de l'application pour essayer de déceler des problèmes et ainsi effectuer des corrections ou des améliorations si besoin.

(TESTS UNITAIRES ET CHRONOLOGIQUE PRINCIPAUX)

Test N ° 1 : Vérifier les requêtes HTTP (API)

Vérifier que la fonction « XHRFetch » est fonctionnelle.

Test à effectuer : `console.log(allCameras)`, `console.log(camera)` et `console.log(reponse)`

Réussi : Si conforme aux informations de l'API.

Test N ° 2 : Vérifier que les produits sont ajoutés sur la page d'accueil

Vérifier que la boucle for (pour chaque caméra) est fonctionnelle ainsi que l'affichage des produits.

Test à effectuer : exemple pour le titre(nom) de la caméra : `let title = document.createElement("h5")`
`title.textContent = allCameras[i].name`

Réussi : Si les titres des produits s'affichent sur la page d'accueil.

Test N ° 3 : Vérifier l'ajout d'un élément au panier (page camera)

Vérifier que l'Event Listener est fonctionnel ainsi que l'ajout dans le LocalStorage et la gestion de l'affichage dynamique.

Test à effectuer : `store.addItemToCart(parameters._id)`

Réussi : Si le LocalStorage contient après clic l'élément ajouté, et si le bouton a été mis en vert avec une encoche de validation. (Même après un refresh de la page)

Test N ° 4 : Vérifier l'affichage des produits sur la page panier

Vérifier que tous les éléments du LocalStorage sont bien affichés sur la page panier.

Test à effectuer : boucle FOR qui affiche chaque produit dans la liste

Réussi : Si les produits s'affichent sur la page panier dans la liste.

Test N ° 5 : Vérifier la suppression d'un produit du panier

Vérifier que la suppression d'un produit du panier est fonctionnelle.

Test à effectuer : Event Listener sur le bouton de suppression avec la suppression côté front (affichage) et côté back (LocalStorage), et mise à jour de l'interface vers « Panier vide » si besoin.

Réussi : Si le produit disparaît de la liste, s'il disparaît du LocalStorage et que le prix total est actualisé.

Test N ° 6 : Vérifier les données du formulaire

Vérifier que le type, le required, et le pattern e-mail sont bien fonctionnels.

Test à effectuer : Test de tous les inputs du formulaire vide puis avec différents caractères.

Réussi : Si les champs du formulaire sont obligatoires et que les données sont validées.

Test N ° 7 : Vérifier la requête POST de la commande

(La fonction « XHRFetch » a déjà été validée une première fois à l'étape 1)

Test à effectuer : console.log(reponse) et outil de développement > Network > XHR, et LocalStorage clear avec store.resetCart().

Réussi : Si la réponse est conforme à la commande, et qu'on observe une validation dans Network > XHR.

Test N ° 8 : Vérifier que l'orderId et le prix total sont bien affichés sur la page validation

Une fois que la réponse de la requête POST a été validée, transmission via query string parameters.

Test à effectuer : Redirection vers la page de validation avec les query string parameters, orderId et finalPrice,

Réussi : Si la page indique bien le numéro de commande ainsi que le prix total de la commande.