

Plan de test d'orinoco (Site E-commerce vente d'appareil photos)

Objectif du plan de test

Notre plan de test nous permet de trouver les anomalies avant que l'utilisateur ne les trouve, et de vérifier le bon fonctionnement de notre site internet

TEST D'INTEGRATIONS

Test n°1 : L'utilisateur se connecte sur index.html les caméras s'affichent avec la description, l'image, le titre, le prix :

Vérifier que la fonction sur le script: index.js ligne 57 products (). then(function(products)

Résultat attendu : Les produits s'affichent avec l'image, le nom et le prix

Test n°2 : Chaque appareil photo est muni d'un bouton qui renvoie vers un lien unique et qui renvoie sur la page products.html

On récupère dans le script "produits.js" de ligne 18 à 38

Résultat attendu : On obtient l'id de la caméra qui apparaît dans l'URL et il n'y a qu'une caméra présente avec son image, titre, prix, description, bouton ajout au panier, menu déroulant des lentilles

Test n°3 : Quand l'utilisateur click sur le bouton "ajouter au panier" dans le local storage un tableau apparaît contenant un ID correspondant à l'id dans l'URL

products.js : "ajoutPanier,addEventListener('click, function()'" ligne 110 à 118

Resultat attendu : Dans le LocalStorage un tableau contenant un seul id correspondant à l'id de l'URL

Test n°4 : L'utilisateur qui a maintenant un produit dans son panier, si l'utilisateur souhaité rajouter un autre produit celui-ci doit retourner dans la page index.html et de rajouter un autre produit

Vérifier que la fonction sur le script:products.js : fonction nombreProduit() ligne de 121 à 132

Résultat attendu : dans le LocalStorage tableau contenant deux id correspondant à l'id de l'URL avec en dernier l'id correspondant à l'id dans l'URL

Test n°5 : Le panier présente les caméras (image, prix unitaire et total et quantité)

Vérifier que la fonction sur le script: panier.js : ligne de 24 à 45

Résultat attendu : Deux articles sont présents ou moins

Test n°6 : L'utilisateur arrivé sur le panier avec 2 caméras il souhaite supprimer la dernière caméra qu'il l'ajoute on appelle la fonction deleteButton

Vérifier que la fonction sur le script: panier.js ligne de 104 à 156

Résultat attendu : La quantité pour chaque objet est passé de 2 à 1 le total est mis à jour

Test n°7 : Si l'utilisateur actualise la page avec les deux caméras doit être toujours présent

Vérifier que la fonction sur le script : panier.js : fonction chargementPanier ligne de 3 à 13

Résultat attendu : rien ne doit changer

Test n°8 : Si l'utilisateur supprime tous ses articles à l'aide de la fonction delete Button

Vérifier que la fonction sur le script : panier.js ligne de 104 à 156

Résultat attendu : la page doit afficher "votre panier est vide" à la place des articles

Test n°9 : Si l'utilisateur clique sur valider "mon panier "alors que le panier est vide une alerte doit s'afficher

Vérifier que la fonction sur le script : panier.js : fonction achat ligne de 163 à 172

Résultat attendu : Un message d'alerte s'affiche "votre panier est vide, impossible de passer une commande"

Test n°10 : Si l'utilisateur à un objet dans son panier et souhaite valider son panier une alerte de formulaire doit s'afficher en rouge

Vérifier que la fonction sur le script : panier.js: ligne 251 à 278 " return event.preventDefault"

Résultat attendu : Indication disant "format de votre prénom incorrect..." affiché en rouge"

Test n°11 : L'utilisateur valide le formulaire et clique sur valider mon panier

Vérifier que dans le panier.js: variable request ligne 283 à 296
"

Résultat attendu : l'utilisateur doit être redigé vers la page de confirmation.html

Test n°12 : Une fois que l'utilisateur à valider son panier

Vérifier que dans la page confirmation.js : fonction commande: ligne de 4 à 35"

Résultat attendu : Un id de commande et total payé doit apparaître

