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 trouves, et de vérifier le bon fonctionnement de notre site internet

Test UNITAIRES

***Test n°1 : Vérifier que la requête API AJAX***

Vérifier que la fonction ***« cam () »***obtient une réponse positive de l’API

**Comment :** ***console.log(products) :***

Si OK : une réponse 200 et contient la liste des produits

***Test n°2 : Vérifier que l’erreur de la requête API AJAX***

Vérifier que la fonction ***« cam () »*** obtient une erreur dans ma réponse de l’API

**Comment :** par une *alert(« Nous sommes désolés, le serveur ne répond pas ! »)*

**Réussi :** Si code erreur http 400, 404 ou 500 dans la console

***Test n°3 : Vérifier que les informations des « caméras » sont bien récupérées***

***Comment :*** ***console.log(products)***

**Réussi :** Si le tableau de caméras apparait dans la console

***Test n°4 : Vérifier les éléments ajoutés dans la page index***

Vérifier que la fonction ***« products.forEach »*** affiche les éléments dans la page index.html

**Comment :** Exemple : pour NAME => ***var nom =document.createElement(h3)***

***Nom.textContent = camera.name***

**Réussi :** Si les articles d’affiche dans la page index.html

**Test n°5 : Vérifier la requête API AXAX pour ID du produit**

Vérifier que la fonction ***« article () »*** obtient une réponse positive de l’API et contient les infos du produit.

**Comment:** ***Network => Name => Headers => statutCode 200 OK***

**Réussi:** Si une réponse JSON contient les infos du produit dans le network => Preview

**Test n°6 : Vérifier l’affichage des infos de l’article sur la page produit**

Vérifier que la fonction ***« affichageProduit() »*** affiche les éléments dans la page ‘produits.html.

**Comment :** Exemple pour Name => ***var nom = document.creatElement(h3)***

***Nom.textContent= camera.name ;***

***Nom.id = ‘nom’ ;***

**Réussi :** Si les articles s’affichent bien dans la page « produits.html ».

**Test n°7 : Panier envoyé au LocalStorage**

Vérifier que la fonction ***« ajoutPanier () »*** envoie un tableau contenant les produits au LocalStorage lorsque l’utilisateur clic du bouton « ajouter au panier »

**Paramètres :** ***localStorage.setItem(‘panier’.JSON.stringify(panier)) ;***

**Réussi :** Si dans le localStorage s’affiche un tableau contenant les produits mis au panier grâce au clic du bouton ***« ajoutPanier ».***

**Test n°8 : Vérifier le mesage de réussite**

Alerter le client que l’article a bien été ajouté au panier lors du clic du bouton « ajouter au panier »

**Comment :** ***alert(« Article ajouté au panier) »***

**Réussi :** Si l’alerte s’affiche sur le navigateur.

**Test n°9 : Vérifier l’affichage des articles mis au panier sur la page panier**

Vérifier la fonction ***« affichagePanier () »*** affiche les éléments dans la page ‘panier.html’.

**Comment :** Exemple pour Name => ***var nom= document.creatElement(‘h3’)***

***Non.textContent = camera.name ;***

**Réussi :** Si les articles s’affichent dans la page ‘panier.html ‘ sous forme de petite fiche article

Et si le panier est vide : affiche la phrase ***« Votre panier est vide »***

**Test n°10 : Vérifier le coût total des articles ajoutés**

Vérifier que l’affichage du cout total apparait bien dans la page ‘panier.html’

**Paramètres :** ***var total = localStorage.getItem(‘prixTotal’) ;***

***Var prixTotal = document.getElementById(‘total’) ;***

**Réussi :** Si le prix total s’affiche bien au bas de la page ‘panier.html’.

**Test n°11 : Vérifier le bouton supprimer le produit dans le panier**

Vérifier la fonction *« deletebuttons() »* en supprimant un produit dans le panier.

**Comment :** En sélectionnant dans le DOM le nom du produit, et en le supprimant du LocalStorage

Ensuite en le supprimant de l’affichage de la page ‘panier.html’

**Réussi :** Si au clic du bouton « supprimer l’article » l’article se supprime, et en créant une alerte sur le navigateur client « Nom de la caméra a été supprimer de votre panier »

**Test n°12 : Vérifier la requête POST de la commande**

Vérifier que la fonction ***« achat »*** poste bien la requête lorsque le formulaire est bien validé

**Comment:** ***Network =>Name => Headers => StatutCode : 201 created***

**Réussi:** Si ORDER renvoie une réponse 201

**Test n°13 : Vérifier orderId et prix sont bien affiché sur la page commande**

Vérifier la fonction ***« commande () »*** affiche bien le numéro de commande généré par la requête POST et le cout total dans la page de confirmation de commande dans la page ‘commande.html’

**Comment :** ***var div = document.creatElement(‘div’) ;***

***div.textContent = « merci pour votre commande …. » ;***

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

**Test n° 14 : Vérifier que les inputs du formulaire sont bien obligatoires**

Vérifier que les données des inputs que les utilisateurs remplirons soient bien obligatoire et que l’envois de la requête POST soit bien stoppé si un input n’est pas rempli.

**Comment :** En mettant dans la balise de l’input ***« required »*** rends le champ obligatoire, et en ajoutant dans le fichier ***JS « return event.preventDefault() ; »*** cela empêche le run de la fonction si le champ n’est pas correctement rempli.

**Réussi :** Si au clic du bouton « valider commande » rien ne se passe si un champ du formulaire n’est pas rempli. (Ou mal rempli).