**PLAN DE TEST D'ORINOCO *(VENTE D’OURSON EN PELUCHE)***

**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 e-commerce «Orinoco».

**TESTS UNITAIRES**

**Test N ° 1 : Vérifier la requête API AJAX**

Vérifier que la fonction « teddy() » obtient une réponse positive de l'API.

Comment : console.log(products);

Réussi : si une réponse 201 contient la liste des produits.

**Test N ° 2 : Vérifier l'erreur de la requête API AJAX**

Vérifier que la fonction « teddy()  » obtient une erreur dans la réponse de l’API.

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

Réussi : si code d'erreur http 400, 404 ou 500 dans la console.

**Test N ° 3 : Vérifier que les informations du Teddy sont bien récupérées**

Vérifier que les informations des Teddys sont bien récupérées du serveur vers notre site e-commerce.

Comment : console.log(products);

Réussi : si le tableau de teddy apparait dans la console.

**Test N ° 4: Vérifier que le prix du Teddy est enregistré**

Vérifier que les Teddys enregistre bien leur prix.

Comment : console.log(products[i].price);

Réussi : si teddy avec son prix dans la console.

**Test N ° 5 : Vérifier les éléments ajoutés dans la page d'index**

Vérifier que la fonction « products.forEach() » affiche les éléments dans la page ‘index.html’.

Comment : exemple pour Name => var nom = document.creatElement(‘h4’) ;

nom.textContent = teddy.name ;

Réussi : Si les articles s’affichent dans la page ‘index.html’.

**Test N ° 6 : Vérifier la requête API AJAX pour obtenir l'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 .

**Tests N °7 : 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 = teddy.name ;

nom.id = ‘nom’ ;

Réussi : Si les articles s’affichent dans la page ‘produits.html’.

**Test N ° 8 : 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 « Ajouter au panier ».

**Test N ° 9 : Vérifier le message de réussite**

Alerter le client que l’article à bien été 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 ° 10 : Vérifier l’affichage des articles mis au panier sur la page panier**

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

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

nom.textContent = teddy.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 ° 11 : Vérifier le coût total des articles ajoutés**

Vérifier que l’affichage du coût 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 ° 12 : Vérifier le bouton supprimer le produit dans le panier**

Vérifiez 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 Du Teddy a été supprimer de votre panier ».

**Test N ° 13 : Vérifier la requête POST de la commande**

Vérifier que la fonction « achat() » poste bien la requête lorsque le formulaire est bien valide.

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

Réussi : Si ORDER renvoie une réponse 201.

**Test N ° 14 : Vérifier la requête POST de la commande**

Vérifier que la fonction « achat() » poste bien un objet CONTACT et un tableau de PRODUITS avec les ID des produits.

Comment : Network => Name => Headers => Request Payload : {Contact…{} : Products…[]}.

Réussi : Si ORDER renvoie une réponse 201 ainsi que l’objet des données envoyées.

**Test N ° 15 : 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 coût 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… BLablABLA » ;

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

**Test N ° 16 : Vérifier que les inputs du formulaire sont bien obligatoires**

Vérifier que les données des inputs que les utilisateurs remplirons soient bien obligatoire. 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).

**TESTS D'INTÉGRATION**

**Test N ° 1: Vérifier le message d'erreur en cas de mauvaise communication avec l'API**

*Vérifier si un message d'erreur apparaît lorsque le projet à une mauvaise communication avec l'API.*

**Test N ° 2: Liste des produits en cas de réponse positive de l'API**

*Vérifier si la liste des produits apparaît lorsque l'API renvoie une réponse positive.*

**Test N ° 3: Vérifier que le produit cliqué apparaît bien sur la page produit**

*Vérifier si le produit cliqué sur la page d'index s'affiche sur la page du produit.*

**Test N ° 4: Un message de vérification de succès apparaît lorsque le produit est ajouté au panier**

*Vérifier si un message de réussite apparaît en haut de l'écran lorsqu'un produit est ajouté au panier.*

**Test N ° 5: Vérifier l'affichage lorsque le chariot est vide**

*Vérifier si un message apparaît sur la page du panier lorsque le panier est vide.*

**Test N ° 6: Vérifier l'affichage lorsque le chariot n'est pas vide**

*Vérifier si la liste des produits apparaît sur la page du panier lorsque le panier n'est pas vide.*

**Test N ° 7: Vérifier le message d'erreur lors d'une tentative de confirmation sans produit**

*Vérifier si un message d'erreur apparaît lorsque l'utilisateur souhaite confirmer la commande sans produit dans le panier.*

**Test N ° 8: Vérifier le message d'erreur lorsque le prénom est manquant ou incorrect**

*Vérifier si un message d'erreur apparaît lorsque le prénom est manquant ou incorrect dans le formulaire pendant que l'utilisateur souhaite confirmer la commande.*

**Test N ° 9: Vérifier le message d'erreur lorsque le nom est manquant ou incorrect**

*Vérifier si un message d'erreur apparaît lorsque le nom de famille est manquant ou incorrect dans le formulaire pendant que l'utilisateur souhaite confirmer la commande.*

**Test N ° 10: Vérifier le message d'erreur lorsque l'e-mail est manquant ou erroné**

*Vérifier si un message d'erreur apparaît lorsque l'e-mail est manquant ou incorrect dans le formulaire pendant que l'utilisateur souhaite confirmer la commande.*

**Test N ° 11: Vérifier le message d'erreur lorsque l'adresse est manquante ou erronée**

*Vérifier si un message d'erreur apparaît lorsque l'adresse est manquante ou incorrecte dans le formulaire pendant que l'utilisateur souhaite confirmer la commande.*

**Test N ° 12: Vérifier le message d'erreur lorsque la ville est manquante ou erronée**

*Vérifier si un message d'erreur apparaît lorsque la ville est manquante ou incorrecte dans le formulaire pendant que l'utilisateur souhaite confirmer la commande.*

**Test N ° 13: Vérifier la confirmation de commande**

*Vérifiez que l'utilisateur soit bien redirigé vers la page de confirmation de commande. Mais aussi si le Numéro de commande généré par le serveur, ainsi que le coût total de la commande y figure bien.*