



## Explication Plan de tests

Ce plan de test nous permettra de repérer les anomalies du site avant que les utilisateurs de notre site ne les découvrent. Il s'agit donc d'une vérification des différents éléments du site Orinoco.

### - 1 Affichage dynamiques des produits

L'utilisateur se connecte sur index.html. Les ours s'affichent avec leur nom, image, prix et un bouton de description.

**Documents concernés :** index.html et index.js

**Lignes testées :** Index.js => ligne 3 à 28 => (generateLine)

**Résultat :** Nous nous attendons à ce que notre page affiche le nombre d'articles dans le serveur. Si il n'y a pas d'articles dans le serveur, aucun article ne sera affiché dans la page index.html.

### - 2 Utilisation des url dynamiques

Quand l'utilisateur clique sur un Ours, la page produit.html s'affiche avec les informations du produit sélectionné.

**Documents concernés :** index.html, produit.html et produit.js.

**Lignes testées :** produit.js => ligne 9 à 13 => (recuperationUrl)

**Résultat :** Cette fonction doit récupérer l'URL perso de la page produite et de l'api du produit.

### - 3 Contrôle des informations de l'article

Vérifier que la page produit.html affiche correctement les éléments de l'Ours sélectionné

**Documents concernés** : index.html, produit.html et produit.js

**Lignes testées** : produit.js => ligne 20 - 43 => (generateLine)

**Résultat** : La ours s'affiche correctement avec son nom, image, prix et description. Tout cela, récupéré dans le serveur grâce à l'id de l'ours dans l'url.

### - 4 Ajout de la couleur et de la quantité sélectionnée

Vérifier que la couleur et la quantité se rajoutent bien dans le LocalStorage lorsque nous cliquons sur le bouton « Ajouter au Panier ».

**Documents concernés** : produit.js

**Lignes testées** : produit.js => ligne 45 à 57 => (changeValeur) et (changeCouleur)

**Résultat** : Dans le LocalStorage, nous devons avoir la couleur et la quantité lorsque l'utilisateur clique sur le bouton « Ajouter au Panier ».

## - 5 Ajout des l'articles dans le LocalStorage

Vérifier que l'ours se rajoute dans le LocalStorage lorsque nous cliquons sur le bouton « Ajouter au Panier ».

**Documents concernés** : index.html, produit.js

**Lignes testées** : produit.js => ligne 61 à 79 => (add)

**Résultat** : Dans le LocalStorage, nous devons avoir un Objet contenant les ours mis dans le panier lorsque l'utilisateur clique sur le bouton « Ajouter au Panier ». Celui-ci se mettra à jour lorsque nous rajouterons des produits dans le panier.

## - 6 Vérification du bouton corbeille

Nous devons vérifier si le bouton : vider le panier dans la page panier.html fonctionne correctement.

**Documents concernés** : panier.js

**Lignes testées** : panier.js -> ligne 12 à 21 => (deleteItem)

**Résultat** : Lorsque l'on clique sur ce bouton en forme de corbeille, nous voyons bien que l'article concerné quitte le panier, que le montant total est mis à jour et que le LocalStorage est mis à jour.

## - 7 Vérification du bouton : vider le panier

Nous devons vérifier si le bouton : vider le panier dans la page panier.html fonctionne correctement.

**Documents concernés** : panier.js

**Lignes testées** : panier.js -> ligne 97 à 101 => (supArticle)

**Résultat** : Lorsque l'on clique sur ce bouton, le LocalStorage doit se vider et l'utilisateur doit être redirigé vers la page d'accueil.

## - 8 Envoi des données au serveur

retourne au serveur la constante data avec la méthode post. Elle contient l'Id des produits et les informations de contact de l'utilisateur.

**Documents concernés** : panier.js

**Lignes testées** : panier.js -> ligne 105 à 124 -> (retourFormulaire)

**Résultat** : Le serveur doit retourner un numéro de commande.

## - 9 Vérification des entrées des formulaires

Nous vérifierons les saisies de l'utilisateur dans le formulaire avant l'envoi de la commande au serveur.

**Documents concernés :** panier.js

**Lignes testées :** panier.js => ligne 142 à 200 => "validPrenom validNom validAdresse validVille validEmail"

utilisation des expressions régulières (Regex).

**Résultat :** L'utilisateur ne peut indiquer des chiffres ou caractères spéciaux dans le champ nom et prénom.

Il ne peut pas entrer de caractères spéciaux dans le champ adresse.

Il doit indiquer une adresse mail sous un format valide.

Si le formulaire est mal rempli il recevra un message d'erreur et le formulaire ne sera pas validé.

## - 10 Vérification du drop-down

Lorsque l'utilisateur ajoute des articles au panier dans la page index.html et produit.html, ces articles doivent tous se retrouver dans le drop-down présent en haut à droite de la page

**Documents concernés :** count.js

**Lignes testées :** count.js -> ligne 11 à 28 -> (generateLineUp) et (notGenerate)

**Résultat :** Génère un panier en haut à droite de la page index.html et produit.html qui affiche à l'utilisateur son panier actuel.

Quand le panier est vide elle doit retourner la phrase "panier vide".

## - 11 Vérification du panier

Au clic sur le bouton panier la page doit être différente si le panier est vide ou si il contient des articles.

**Documents concernés :** count.js

**Lignes testées :** count.js -> ligne 47 à 53 -> (verifPanier)

**Résultat :** La page doit retourner la page panierVide.html si le panier ne contient pas d'article et si le panier contient des articles elle doit retourné la page panier contenant le panier de l'utilisateur.