|  |
| --- |
| Plan de test Vintageori |

|  |
| --- |
| Ce plan de test servira à vérifier que toute la fonctionnalité de l’application fonctionne correctement. IL sera découpé en 5 partie (chaque partie représentant une page javascript) |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **recupApPhoto.js** | | | | |
|  | Description | Paramètres | Résultat attendu | Résultat si erreur |
| Fetch API | Vérifier que la fonction fetch(urlAPI) reçoive une réponse positive de l’API | Méthode GET url"http://localhost:3000/api/cameras" | Une réponse contenant la liste des produits | Une erreur 400 et aucune réponse de l’API |
| Response.json | Vérifier que la reponse.json() retourne les informations désirées | Vérifier qu’on reçoit bien une réponse en entrant l’url | Un retour contenant les informations contenues dans l’API | Erreur 400 ressource non trouvé |
| Test unitaire pour  les clés de l’appareil photo sélectionné | Vérifier que chaque clé d’appareil photo renvoi la  valeur appropriée | /\* Récupération de l'id du produit sélectionné \*/  const productId = window.location.search.substr(1);  const url = `http://localhost:3000/api/cameras/${productId}`; | Chaque produit doit avoir les mêmes clés  / valeurs | Des différences entre les produits |
| .product\_\_btn | Vérifier qu’au clic sur « plus de détails » (.product\_\_btn) on accède bien à la page produit détaillés | function displayAllProducts(products)  ligne 28 | La nouvelle page va afficher l’url attendu contenant l’id de l’appareil photo en question | Url incorrecte qui n’a soit pas le bon ID soit aucun ID |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **produitdetaille.js** | | | | |
|  | Description | Paramètres | Résultat attendu | Résultat si erreur |
| Fetc API &  response.json() | Même chose que pour l'index | | | |
| Contenu  dynamique | Vérifier que la page charge l'appareil photo  approprié en fonction de l'ID présent dans  l'url | URL & endpoint ${id} //  function displayProduct(product) pour la Fonction d'affichage du produit !  insertAdjacentHTML pour le  contenu | En fonction de l'appareil photo sélectionné dans  l'index, la page actuelle devrait générer le  contenu pour l'appareil en question | Une page d'erreur est générée qui  demande à l'utilisateur de revenir à  l'accueil |
| Choix des lentilles | Vérifier que chaque appareil photo a les bons  choix de lentilles | function (lense)  Ligne 48à 54 | En fonction de l'appareil photo, le type de lentilles  devraient correspondre à celles  renvoyées par l'API | Absence de type de lentille ou pas de type de lentille |
| Bouton Panier | Vérifier qu’au clic sur « ajouter au panier » le produit sélectionner se retrouve bien dans le panier et rajoute le  produit au localStorage lorsqu'il est cliqué | function addToCart (product)  Ligne 63  Création d'un localStorage, ajout de  toutes les valeurs du produit | Au clique sur « ajouter  au panier », toutes les données du produit  sont ajoutées au localStorage | Le bouton n'éxécute rien OU rien n'est  ajouté au localStorage |
| Contrôle des  Quantités et Affichage du nombre de produit sélectionné donc à commander | S'assurer que la quantité ajouté au panier ne puisse pas  être inférieur à 1 et Vérifier que le nombre des produits rajouter au panier s’afficher correctement sur la page panier et la page produit détaillé | function addToCart (product)  Pamis les valeurs du produit ajouter au local storage rajouter la valeur « quantité :1 »  Puis dans cette fonction rajout de la fonction function verifPanier() | Au clique sur « ajouter au panier » le nombre de produit commandé dans le panier augmentera de 1 et augmentera de 1 à chaque fois qu’on recliquera sur ce bouton | Aucun rajout de produit |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **nombreArticleDansLePanier.js** | | | | |
|  | Description | Paramètres | Résultat attendu | Résultat si erreur |
| Affichage du nombre de produit sélectionné donc à commander | Vérifier que le nombre des produits rajouter au panier s’afficher correctement sur la page panier et la page produit détaillé | Dans la fonction function addToCart (product) rajout de la fonction function verifPanier() | Affiche le nombre de produits dans le panier. Affiche 0 s'il n'y en a pas | Ne met pas de quantité |
| Problème de conexion au serveur | Vérifier qu’un message d’erreur s’affiche si la connexion au serveur pas pu se faire | function displayError() | Affiche message d’erreur | N’affiche rien |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **panier.js** | | | | |
|  | Description | Paramètres | Résultat attendu | Résultat si erreur |
| Affichage si panier  vide | Vérifier que la page affiche le contenu  voulu si le panier est vide | function displayCart() si le panier est vide  inserAdjacentHTML pour le contenu | Un message indiquant le fait que le panier  soit vide, modification visuelle de la page | Le message ne s'affiche pas, la page  n'est pas modifiée |
| Création du  contenu | Vérifier que le contenu généré via  inserAdjacentHTML soit correct et que toutes les  valeurs soient exactes | inserAdjacentHTML avec des ${clés} pour afficher  le contenu dynamiquement | chaque ${clé} renvoi la valeur attendue | Les valeurs sont incorrectes OU aucun  contenu n'est généré |
| Modification quantité produit sélectionné et présent dans le panier | Vérifier que l’on peut soit augmenter ou diminuer la quantité du produit présent dans le panier  Vérifier la fonctionalité de function addOneProduct() | Augmenter de 1 la quantité d’article commandé => Panier.js function addOneProduct()  Ligne 135 à 139  Diminuer de 1 la quantité d’article commandé  => Panier.js function removeOneProduct()  Ligne | On pourra modifier la quantité | On ne peut modifier la quantité |
| Suppression d’un produit dans le panier | Vérifier que l’on peut supprimer un produit dans le panier au clique sur < button class="cart-section\_\_delete> et si c’est le dernier produit sa afficherai « panier vide, revenir à la page d’accueil » au clique <button class="cart-section\_\_cancelCart">  Verification de la fonctionnalité des fonction function deleteProduct() et function refreshSectionAndCart() | Supprime un article dans le panier  =>panier.js function deleteProduct()  Ligne 167  Affiche panier vide, revenir à la page accueil  =>panier.js function refreshSectionAndCart()  Ligne 184 | Suppression du produit dans le panier et si c’est le dernier  Et annule la commande si le dernier produit est enlevé du panier | Rien ne se passe |
| Annulation de tout le panier | Vérifier que lorsque je clique sur « annuler le panier » dans la page panier contenant une commande, on revoit apparaitre « panier vide, retour à la page d’accueil » | panier.js function cancelCart()  Ligne 177à 181 | Supprime la commande  Et raffraichi la page panier et , on revoit apparaitre « panier vide, retour à la page d’accueil » | Rien ne se passe |
| Prix total de la commande calculé | Vérifier que les prix de chaque produit sélectionné peuvent s’additionner et donner un total sur la page panier | function displayCart()  Ligne 32à 34 | On voit apparaitre la somme du prix de tout nos produits dans le panier | Rien ne passe donc il y a une erreur |
| Formulaire | Vérifier que tous les inputs du formulaire  soient fonctionnels et appropriés | Formulaire avec input text pour "prénom /  nom / adresse / ville" & input email pour  l'adresse email + contrôle via "pattern" | L'user peut renseigner des informations  dans le formulaire, ces informations sont  limitées par les patterns | Le formulaire renvoi des erreurs lorsque  l'user essaie de renseigner des  informations |
| Création de  "contact" | Vérifier que l'objet contact soit bien crée  et contienne toutes les données du  formulaire | Déclaration contact = { données du  formulaire } | L'objet contact renvoi les données du  formulaire (prénom, nom, adresse, ville &  email) | Erreur 400, les informations du formulaire  ne sont pas sauvegardées dans l'objet  contact |
| "request" | Vérifier que "request" soit fonctionnel et  qu'on reçoive bien l'orderID en retour | Requête de type post avec l'url "/order",  body contenant "contact products",  header 'Content-Type': 'application/json' | La requête s'envoi avec succès | La requête post échoue, Erreur 400 |
| Réponse de  "fetch" | Vérifier que la response.json() de fetch  renvoi les bonnes informations | Réponse provenant de "request" | La response.json() retourne un orderID  unique et différent à chaque requête | Réponse vide OU pas de réponse |
| Bouton « Valider le panier » | Vérifier que l'interaction avec le bouton  soit fonctionnelle | Fonction postOrder(contactProducts) | Si tout est bien rempli et fonctionne correctement ,la  fonction s'exécute :  Envoi au serveur l'objet contact et le tableau d'id products au format string  Enregistrement de l'objet contact et l'orderId reçus du serveur, ainsi que le total de la commande sur le localStorage. vide le localStorage,  fait  apparaître un message de confirmation | Une des informations n'est pas récupérée  ou une partie de la fonction ne s'exécute  pas |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Confirmation.js** | | | | |
|  | Description | Paramètres | Résultat attendu | Résultat si erreur |
| Récupération du  localStorage | Vérifier que la déclaration du localStorage  soit fonctionnelle et que les informations à  l'interieur soient celles attendues | \* Récupération des différents éléments dans le localStorage afin de les afficher au chargement de la page \*/  const confirmation = document.querySelector("#confirmation-section");  const contact = JSON.parse(localStorage.getItem("contact"));  const orderId = JSON.parse(localStorage.getItem("orderId"));  const total = JSON.parse(localStorage.getItem('total')); | Un array contenant : l'orderID unique pour  la commande && le coût total de la  commande | Pas d'orderID ou prix total incorrect OU  aucune information |
| Confirmation commande | Vérifier que lorsque l’on confirme une commande (en aillant remplit correctement notre formulaire, en cliquant sur « valider panier ») on reçoit en retour un message avec nos coordonnées, le prix total de notre commande et un numéro de la commande comme preuve d’achat | confirmation.js ligne 8 à 19  Le message renvoyer par le serveur contenant (coordonnées, prix total commande et numéro de commande)  insertadjacentHTML pour le contenu contenant le message de confirmation de commande | Le message de confirmationde commande contenant les informations sur nos coordonnées, le prix total et un numéro de la commande comme preuve d’achat | Pas de création de contenu OU les  variables sont incorrectes |