|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Fichier JS** | **Lignes de code testées** | **Fonction testée** | **Résultat attendu** | **Comment vérifier le résultat attendu** | **Problème possible** |
| 1 | index.js | 4 à 8 | main | 1. Récupère les données du service web et la stocke dans la variable ‘products’. 2. Construit le HTML avec les données ‘products’ | Observer le contenu de la variable ‘products’ via un console.log | Voir ligne 2 |
| 2 | index.js | 11 à 22 | getProducts | Fait une requête vers le service web et récupère les données | Observer la réponse de la requête fetch via un console.log | -Erreur serveur |
| 3 | index.js | 24 à 37 | displayProduct | Après avoir récupérer les données, permet d’afficher le résultat dans la page HTML | Observer directement le rendu sur la page web | -Erreur sur le nom des propriétés |
| 4 | product.js | 7 à 19 | productPage | 1. Récupère les données du produit 2. Construit le HTML de la page 3. Ajoute le produit au panier | Observe via un console.log le contenu de la variable ‘product’ | Voir lignes 5, 6, 7 |
| 5 | product.js | 23 à 36 | loadProduct | Fait une requête vers le service web en récupérant l’Id du produit via l’URL | Observe la réponse de la requête fetch via un console.log | -Erreur serveur |
| 6 | product.js | 38 à 57 | displayProductPage | Après avoir récupérer les données, permet d’afficher le résultat dans la page HTML | Observer directement le rendu sur la page web | -Erreur sur le nom des propriétés |
| 7 | product.js | 60 à 100 | addProduct | 1. Créer un objet tableProduct et y insérant les données du produit en valeur 2. Ajoute le l’objet tableProduct dans le localStorage 3. Incrémente le prix total dans le localStorage | 1. Observe le contenu de l’objet avec un console.log 2. Observe « monPanier » et « prixTotal » dans le localStorage du navigateur | -le nouveau produit ajouté écrase le panier au lieu d’être ajouté à un panier déjà rempli dans le localStorage  -Ecrasement du prix total au lieu de l’incrémenter dans le localStorage |
| 8 | basket.js | 1 à 46 | displayBasket | 1. Récupère le panier et le prix total stocké dans le localStorage 2. Si le localStorage est vide, envoie un message sur la page HTML que le panier est vide 3. Si le localStorage n’est pas vide, insère sur la page HTML le contenu du panier et le prix total | 1. Si le panier est vide, affiche dans la console : vide 2. Si le panier contient un ou des produits affiche dans la console : rempli 3. Observe le contenu HTML directement sur la page web | -Erreur sur le nom des propriétés |
| 9 | basket.js | 48 à 50 | deleteOrder | Supprime le contenu du localStorage | Observe directement le localStorage dans le navigateur | Aucune erreur possible |
| 10 | basket.js | 55 à 84 | sendForm | 1. Assure que tous les champs soient remplis par le bon type 2. Récupère les id de chaque produits dans le panier stocké dans le localStorage et les insèrent dans un tableau ‘products’ 3. Crée un objet ‘contact’ avec les informations entrées dans le formulaire 4. Appelle la fonction collectOrder | 1. Si le formulaire est mal rempli, la page affiche un message d’erreur 2. Observe le contenu panier dans le localStorage via un console.log 3. Observe le contenu du tableau ‘products’ via un console.log 4. Observer le contenu de l’objet ‘contact’ via un console.log |  |
| 11 | basket.js | 86 à 95 | createOrder | Fait une requête avec la méthode ‘post’ afin d’envoyer l’objet ‘contact’ et le tableau ‘products’ vers le service web et récupère une réponse | Observe le résultat de la requête dans la fonction collectOrder (ligne12) | -Erreur dans l’url  - Erreur dans le nom des propriétés de l’objet ‘contact’ |
| 12 | basket.js | 97 à 122 | collectOrder | 1. Appelle la fonction createOrder et récupère la réponse de la requête 2. Stocke le numéro de commande contenu dans la réponse dans la variable orderId et l’envoie dans le localStorage sous la clé ‘orderId’ 3. Supprime le panier du localStorage 4. Redirige vers la page confirmation.html | 1. Affiche la réponse de la requête via un console.log de la variable ‘order’ 2. Le catch de la ligne 119 permet de nous renvoyer l’erreur dans la console | -Erreur dans la requête  -Erreur dans la redirection vers la page confirmation |
| 13 | Confirmation.js | 10 | displayConfirmation | 1. Affiche le numéro de commande et le prix total récupéré dans le localStorage 2. Efface tout le contenu du localStorage 3. Empêche l’accès autrement que par la validation du formulaire de la page basket.html | Affiche le numéro de commande et le prix total via un console.log ou sur la page web directement |  |