

Fichier JS	Lignes de code testées	Fonction testée	Résultat attendu	Comment vérifier le résultat attendu	Problème possible
common.js	17 à 24	basketheader	La fonction doit afficher l'état du panier dans le header : le nombre d'article dans le panier ou « Panier (vide) » s'il n'y a aucun article	Appeler la fonction avec le lien du fichier dans lequel le panier sera affiché comme argument et en attribuant un nombre d'article à la variable numberOfItems Input : string / chemin + nom du fichier (de la redirection : basket.html) Output : affichage panier vide ou mention du nombre d'articles dans le panier	Affichage d'un panier vide si des articles sont présents dans ledit panier et inversement Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur. Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.
common.js	27 à 36	basketUpdate	La fonction met à jour la variable numberOfItems qui représente le nombre d'articles dans le panier	Définir un tableau d'objets (ayant chacun un seul attribut itemQuantity) basketDatas vide et contenant plusieurs valeurs dont la somme est connue Input : tableau d'objets / stocké dans basketStorage (localStorage) / liste des produits du panier / id, option, quantité Output : nombre / stocké dans basketLevel(localStorage) / quantité d'articles dans le panier	La somme des valeurs pourrait être erronée L'enregistrement dans le localStorage pourrait ne pas fonctionner Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur. Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.
common.js	42 à 55	getDatas	La fonction doit charger les données stockées dans le fichier disponible en suivant l'url passé en argument	Comparer le résultat en utilisant une url fournie par le backend et celui rendu par l'application Postman Input : string / url de la catégorie dont on veut avoir les produits et leurs données Output : objet / liste des produits et de leurs données	Pas de réponse du serveur ou serveur inaccessible, indisponible Données retour erronées ou absentes Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur.
common.js	58 à 72	postDatas	La fonction doit envoyer une requête et un tableau de données à l'url passée en argument	Comparer le résultat renvoyé par Postman et celui donné par l'utilisation d'une url fournie par le backend et un tableau d'objet dont le format est disponible dans le backend Input : string + objet (objet + tableau) / url de la catégorie concernée par un des produits commandé s+ [{firstname, lastname, address, city, email de l'utilisateur}, {id du produit commandé}] Output : objet / liste des caractéristiques du produit en input + numéro de la commande Editer la variable basketDatas (stockage des données du panier dans le localStorage) ou la visualiser dans les devtools de Chrome	Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur. Pas de réponse du serveur ou serveur inaccessible, indisponible Données retour erronées ou absentes Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur. Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.
product.js	37 à 72	sendToBasket	La fonction collecte les données saisies par l'utilisateur lorsqu'il a validé son choix pour le transfert d'un produit dans le panier et les stocke dans le localStorage	Le contenu de la variable basketDatas (stockage des données du panier dans le localStorage) ou la visualiser dans les devtools de Chrome Input : tableau d'objet, objet / tableau stocké dans basketStorage (localStorage) et objet (id, choix option, quantité du produit) Output : tableau d'objet / stocké dans basketStorage (localStorage)	Le contenu de la variable pourrait être erronée L'enregistrement dans le localStorage pourrait ne pas fonctionner Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur. Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.

basket.js	6 à 28	totalAmount	La fonction calcule et met à jour le montant total des articles présents dans le panier, et stocke la valeur dans le localStorage	<p>Editer la variable amountStorage (stockage du montant du panier dans le localStorage) ou la visualiser dans les devtools de Chrome. Les modifications doivent être également prises en compte sur la page</p> <p>Input : tableau d'objet / tableau stocké dans basketStorage (localStorage) – liste des produits du panier(id, option, quantité) Output :objet/ stocké dans amountStorage (localStorage) et mise à jour de l'affichage du montant total du panier</p>	<p>Le contenu de la variable pourrait être erronée L'enregistrement dans le localStorage pourrait ne pas fonctionner</p> <p>Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur.</p> <p>Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.</p>
basket.js	31 à 54	changeItemQuantity	La fonction calcule et met à jour le nombre total des articles présents dans le panier, stocke la valeur dans le localStorage et met à jour la visualisation de l'état du panier dans le header	<p>Editer la variable basketLevel (stockage du nombre d'articles du panier dans le localStorage) ou la visualiser dans les devtools de Chrome. Les modifications doivent être également prises en compte sur la page</p> <p>Input : tableau d'objet + objet/ tableau stocké dans basketStorage (localStorage) – liste des produits du panier(id, option, quantité) + objet stocké dans basketLevel (localStorage) – nombre d'items dans le panier Output :tableau d'objet, objet/ update des deux inputs + mise à jour de l'affichage du nombre d'items dans le panier du header</p>	<p>Le contenu de la variable pourrait être erronée L'enregistrement dans le localStorage pourrait ne pas fonctionner</p> <p>Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur.</p> <p>Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.</p>
basket.js	57 à 80	removeItem	La fonction supprime un élément du panier et met à jour les données du localStorage ainsi que l'affichage de la page	<p>Editer la variable basketDatas (stockage des données du panier dans le localStorage) ou la visualiser dans les devtools de Chrome. Les modifications doivent être également prises en compte sur la page</p> <p>Input : tableau d'objet / tableau stocké dans basketStorage (localStorage) – liste des produits du panier(id, option, quantité) Output :tableau d'objet, objet/ update du input + update de basketLevel (localStorage) + mise à jour de l'affichage du nombre d'items dans le panier du header</p>	<p>Le contenu de la variable pourrait être erronée L'enregistrement dans le localStorage pourrait ne pas fonctionner</p> <p>Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur.</p> <p>Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.</p>
basket.js	122 à 171	formValidation	La fonction collecte les données saisies par l'utilisateur dans le formulaire de commande et les envoie au serveur qui retourne des données produit et un n° de commande	<p>Comparer le résultat renvoyé par Postman et celui contenu dans la variable response (à éditer) Editer la variable basketDatas (stockage des données du panier dans le localStorage) ou la visualiser dans les devtools de Chrome. Les modifications doivent être également prises en compte sur la page</p> <p>Input : préremplir le formulaire et inclure des erreurs pour tester les différentes conditions regex Output : affichage de fenêtres d'alerte et de confirmation si aucune erreur détectée</p>	<p>Pas de réponse du serveur ou serveur inaccessible, indisponible Données retour erronées ou absentes</p> <p>Les données de développement fournissent 3 catégories de 5 produits. Effectuer un contrôle visuel et systématique des écrans renvoyés pour détecter d'éventuelles erreur.</p> <p>Utiliser également la console et les outils devtools de visualisation du localStorage pour la détection d'erreur.</p>