

	Functionalité	Action	Résultat attendu	Résultat observé
1	Une page d'accueil montrant (de manière dynamique) tous les articles disponibles à la vente.	Ouvrir sur la page d'accueil du site web dans un navigateur	Affichage de l'ensemble des produits	pas d'affichage des produits
2	La page Produit présentant un seul produit ; Présence d'un menu déroulant permettant à l'utilisateur de choisir une option de personnalisation, ainsi qu'un input pour saisir la quantité.	Ces éléments doivent être pris en compte dans le panier.	Affichage de tous les éléments prévus. Bonne fonctionnalité des différents input	-Pas image produit présente, mais logo kanap. Description a revoir en fonction du produit. -input fonctionnent mais les paramètres se transmettent ils au panier?
3	Page Panier: l'utilisateur va pouvoir modifier la quantité d'un produit de son panier. L'utilisateur aura aussi la possibilité de supprimer un produit de son panier, le produit devra donc disparaître de la page.	à la modification des input, le total du panier devra bien se mettre à jour.	Les inputs des utilisateurs doivent être analysés et validés pour vérifier le format et le type de données avant l'envoi à l'API. Il ne serait par exemple pas recevable d'accepter un prénom contenant des chiffres, ou une adresse e-mail ne contenant pas de symbole "@". En cas de problème de saisie, un message d'erreur devra être affiché en dessous du champ correspondant.	Input fonctionnent. Message d'erreur ne s'affiche pas. Les modifications ne s'opèrent pas. L'image du produit ne s'affiche pas : Erreur
4	Page confirmation: Sur cette page, l'utilisateur doit voir s'afficher son numéro de commande.	Affichage numéro de commande	Il faudra veiller à ce que ce numéro ne soit stocké nulle part.	numero de commande est actuellement du html
L.3	script.js: kanapItems	fonction pour récupérer les items de la liste de l'API	récupère la data de l'api et la transfère à la fonction dataToCards	✓
L.22	script.js: dataToCards	fonction pour gérer l'affichage des cartes "Produits" sur la page accueil via l'API	utilise la data de l'Api pour générer les cards affichées sur la page accueil	✓
				✓
L.15	product.js: searchKanapProduct	Récupération des données de l'API en fonction de l'ID des Produits	recupère les données de l'API puis les transfère à la fonction kanapProduct	✓
L.36	product.js: kanapProduct	Fonction qui gère l'affichage du produit, via les données récupérées sur l'API, pour la page produit/product.html	utilise la data de l'API pour générer le produit pour l'affichage	✓
L.189	product.js: pushProductInStorage	Vérifie et met à jour le localStorage en fonction des paramètres sélectionnés par l'utilisateur	Conversion et comparaison des données puis sauvegarde vers localStorage	✓
L.248	product.js: insertAt	Fonction pour insérer un élément à un endroit spécifique du tableau	Ajout dans un tableau d'un nouvel élément (à index+n du premier)	✓
L.254	product.js: docReady	Fonction pour s'assurer que le document est correctement généré avant de récupérer les données pour les ajouter au panier.	valide que le DOM est disponible pour s'assurer que les éléments (boutons) sont correctement chargés, avant d'y ajouter un addEventListener	✓
				✓

L.8	<code>cart.js: getDataFromLocalStorage</code>	Récupération des données du panier depuis le Local Storage	Pour confirmation.html : Affichage du Numéro de commande. Pour cart.html : Vérifie et récupère les données du panier et les transfère à la fonction cartElement	✓
L.36	<code>cart.js: cartElement</code>	Gère l'affichage des produits du panier, sur la page panier	Recupère les données pour afficher les produits sur la page cart	✓
L.178	<code>cart.js: showTotal</code>	Calcul et affichage du prix total, sur la page panier	Dans le tableau panier (arrayCart), on récupère et multiplie pour chaque élément du panier: le prix du produit et la quantité choisie. On réalise ensuite la somme des résultats obtenus. Enfin on les affiche sur la page panier.	✓
L.219 à 290	<code>cart.js: inputFirstName, inputLastName, inputAddress, inputCity, inputMail</code>	Saisie et gestion des coordonnées	Récupère et vérifie, dans les input, les données saisies par l'utilisateur (grâce aux regex)	✓
L.310	<code>cart.js: btnCommander.addEventListener</code>	Validation de la commande	Au clic sur le bouton commander: création de l'objet commande. Ensuite requête post pour récupérer l'orderId par l'API. Si la requête passe, redirection vers la page confirmation. L'orderId est affiché mais n'est pas conservé/stocké (dans le localStorage).	✓