

	Fonctionnalité	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 intégrer les éléments de l'API à la page index	utilise la data de l'Api pour générer la page accueil	✓
L.5	product.js: urlSearchParams	On récupère avec searchParams, à l'adresse url où l'on se trouve, un objet URLSearchParams	récupération objet URLSearchParams	✓
L.6	product.js: productId	Afin de récupérer l'ID du Produit Kanap sur lequel l'utilisateur a cliqué, l'objet URLSearchParams nous permet d'accéder aux arguments décodés de la requête GET contenu dans l'URL.	récupération de l'ID du produit Kanap, sur lequel l'utilisateur a cliqué	✓
L.15	product.js: searchKanapProduct	Récupération data de l'API en fonction de l'ID des Produits	recupère la data puis transfère à la fonction kanapProduct	✓
L.36	product.js: kanapProduct	Fonction pour intégrer les éléments de l'API à la page product	utilise la data de l'Api pour générer la page product	✓
L.188	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, lorsque l'utilisateur ajoute un élément au panier. Au clic, sur le bouton, récupère et vérifie les valeurs sélectionnées par l'utilisateur	✓

L.200	product.js: pushProductInStorage	Vérifie et met à jour les paramètres, entre le localStorage et les sélections de l'utilisateur	Conversion datas + sauvegarde vers localStorage ET comparaison avec datas déjà présente dans localStorage	✓
L.241	product.js: insertAt	fn pour insérer un élément à un endroit spécifique du tableau	Ajout dans [arrayKanap] du nouveau produit (à index+1 du premier)	✓
L.8	cart.js: getDataFromLocalStorage	Récupération des données panier via le Local Storage	Pour confirmation.html : Affichage Numero de commande. Pour cart.html : Vérifie et récupère données du panier et les transfère à la fonction elementDuPanier	✓
L.36	cart.js: elementDuPanier	Génère <article> = le panier de la page panier	Recupère les données pour générer la page cart	✓
L.176	cart.js: showTotal	Gestion et affichage Somme et Total	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	cart.js: inputFirstName, inputLastName, inputAddress, inputCity, inputMail	Saisie et gestion des coordonnées	Récupère et vérifie, dans les input, les données saisies par l'utilisateur (regex)	✓
L.310	cart.js: btnCommander.addEventListener	Validation de la commande	Au clic sur bouton commander: création de l'objet de la commande, puis requête post pour récupérer l' orderId par l'api, puis Si ok, redirection vers page confirmation. orderId est affiché mais n'est pas conservé/stocké (localStorage).	✓