

Fonctionnalité	Action	Résultat attendu	Résultat observé
Une page d'accueil montrant (de manière dynamique) tous les articles disponible à la vente. <a href="#">Script.js</a>	Ouvrir sur la page d'accueil du site web dans un navigateur.	Affichage de l'ensemble des produits.	Ok
Une page produits montrant (de manière dynamique) toutes les informations d'un article sélectionné. <a href="#">Product.js</a>	S'ouvre au « clic », sur un produit de la page d'accueil.	Affichage du produit, avec toutes les informations le concernant. Et avec la possibilité de choisir le nombre et la couleur de celui-ci.	Ok
Choix d'une couleur pour un produit, sur la page produit. <a href="#">Product.js</a>	Au « clic » sur la sélection de couleur : possibilité de choisir une couleur parmi les différentes propositions ; Menu déroulant.	Affichage du nom de la couleur choisis.	Ok
Choix du nombre d'article à sélectionner, sur la page produit. <a href="#">Product.js</a>	Au « clic » sur la flèche du haut : augmentation du nombre d'article de 1. Au « clic » sur la flèche du bas : diminution du nombre d'article de 1.	Affichage du nombre d'article sélectionné.	Ok

Ajout d'un article au panier, sur la page produit. <a href="#">Product.js</a>	Au « clic » sur « Ajouter au panier » : Ajoute l'article avec sa couleur et son nombre choisis, dans le LocalStorage, pour être affiché au panier. Si pas de sélection de couleur ou de nombre : message d'erreur et pas d'ajout au panier.	Affichage de l'article sélectionné sur la page du panier. Si la page panier et déjà ouverte : rafraîchissement de la page en question avec le nouvelle article.	Ok
Liste des produits choisis par le client. <a href="#">Carte.js</a>	Au « clic » sur « Panier » dans le menu d'en-tête : redirection vers la page du panier. Récupération des produits dans le LocalStorage pour afficher les produits.	Affichage des produits sélectionnés, de leur nombre, de leur pris, du prix total, avec la possibilité de changer leur nombre ou de les supprimer.	Ok
Suppression d'un produit au panier. <a href="#">Carte.js</a>	Au « clic » sur « supprimer », la page du panier se rafraîchit et le produit à « supprimer » n'est plus présent.	Le produit supprimer n'est plus affiché sur la page panier. N'existe plus dans le LocalStorage.	Ok
Modification de la quantité d'un produit du panier	Au « clic » sur la flèche du haut : augmentation du nombre d'article de 1. Au « clic » sur la flèche du bat : diminution du nombre d'article de 1.	Affichage du nombre d'article sélectionné. Affichage des prix évoluant en fonction du nombre d'article	Ok

Calcul du prix total d'un type de « kanap » sélectionné. <a href="#">Carte.js</a>	Le prix d'un type de « kanap » est calculé en fonction du nombre sélectionné. Au changement de la quantité de ce type de « kanap » son prix change également.	Affichage du prix total pour chaque type de « kanap » sélectionné. Affichage du prix évoluant en fonction d'une nouvelle quantité choisie.	Ok
Calcul du prix Total de tous les « kanaps » du panier. <a href="#">Carte.js</a>	Le prix total de la commande est calculé en fonction de tous les « kanaps » sélectionnés et de leurs prix respectifs. Au changement de la quantité, le prix total change.	Affichage du prix total de la commande. Ce prix évolue en fonction d'une nouvelle quantité choisie.	Ok
Envoi des informations du client au serveur. <a href="#">Carte.js</a>	Au « clic » sur « commander ! » : les informations du client sont envoyées vers l'API. Pour que l'API puisse comprendre ces informations, il faut un objet « contact » avec les informations du client, ainsi qu'un tableau de « string » « products », avec les identifiants des produits choisis. Cela permettra à l'API de pouvoir envoyer un numéro de commande. Cela est effectué avec un « fetch », de type « POST ». Le numéro de commande est récupéré dans une promise.	Lors des informations envoyées vers le serveur, si aucune erreur n'est présente, un numéro de commande est récupéré. Celui-ci sera affiché sur la page de commande.	Ok

Sécurisation de l'envoi des infos du client. <a href="#">Carte.js</a>	Au « clic » sur « commander ! », si les informations du formulaire ne sont pas remplies ou sont erronées, la commande n'est pas envoyée vers l'API. De plus des messages d'erreurs sont affichés. Pour cela des comparaisons avec des « regex » sont effectuées.	Affichage d'un message d'erreur général. Affichage d'un message d'erreur en dessous de la ligne concernée du formulaire.	Ok
Génération du numéro de commande et d'un identifiant <a href="#">Carte.js</a>	Lorsque l'API reçoit la commande du client, celui-ci renvoie un numéro de commande que l'on va stocker dans le LocalStorage. Un numéro d'identifiant est créé en fonction du jour où la commande est effectuée et du formulaire. Suppression des « kanaps » du LocalStorage.	Le numéro de commande est stocké dans le LocalStorage pour être affiché sur la page de confirmation. L'identifiant est affiché sur l'URL de la page de commande, lors de l'appel de celle-ci.	Ok
Redirection vers la page de confirmation <a href="#">Carte.js</a> & <a href="#">confirmation.js</a>	Au « clic » sur « commander ! », après vérification du formulaire, réception du numéro de commande et création de l'identifiant, l'utilisateur est redirigé vers la page de confirmation. Après être récupéré dans le LocalStorage le numéro de commande est affiché. L'identifiant est affiché dans l'URL. Suppression du numéro de commande du LocalStorage.	Affichage du numéro de commande, sur la page de confirmation. Affichage de l'identifiant, dans l'URL de la page de confirmation.	Ok