

N°	page testée	Test	Ligne testées	Résultat attendu	Résultat
1	index.html	L'utilisateur se connecte sur index.html. Les caméras s'affichent avec image, titre, prix et description	index.js : Fonction generationArticle();	5 caméras s'affichent avec image, titre, prix et description.	OK
2	index.html	Chaque caméra est muni d'un lien unique qui renvoie vers sa page product.Html unique	index.js : Fonction urlDynamique();	chaque lien est unique et renvoi vers le bon objet.	OK
3	product.html	L'utilisateur clique sur le lien objet sur index.html et arrive sur la page product.html du produit.	product.js : ligne 54 à 65	l'id de la caméra apparait dans l'URL, il n'y a qu'une caméra présente avec image, titre, prix, bouton "ajouter au panier", description, menu déroulant de personnalisation de lentilles affichés	OK
4	product.html	L'utilisateur arrive avec un panier vide sur le site (var panier = [ ]; et localStorage.clear();). Clique sur le bouton "ajouter au Panier". Ajouter un console.log ligne 45 tel que : console.log(localStorage.getItem('item')); Si la console affiche un tableau contenant un id correspondant à l'id dans l'Url,	product.js : event "click" ligne 41 à 47	La console affiche un tableau contenant un seul id correspondant à l'id dans l'Url.	OK
5	product.html	L'utilisateur a maintenant un objet dans son panier. Il se dirige vers une page product.html autre que celle de l'objet présent dans son panier. Et ajoute cet objet au panier	product.js : variable panier, ligne 37	La console affiche un tableau contenant 2 id avec en dernier l'id correspondant à l'id dans l'Url.	OK
	panier.html	Le panier présentent les caméras (image, prix unitaire, sous-total objet, quantité)	panier.js : ligne 31 à 37 (visibilité des blocs, nouveau Set); ligne 109 à 244 (création de l'article)	Deux caméras sont présentes.	OK

6	panier.html	L'utilisateur arrive sur le panier avec 2 caméras. Il supprime la dernière caméra qu'il a ajoutée avec le bouton -1 (article du bas). On vérifie alors que les identifiants bouton sont uniques et liés à l'objet	panier.js : ligne 136:192 à 136:222 pour l'id, ligne 193 à 195	L'article disparaît, le total est mis à jour. Il ne reste qu'un objet, le total à payer, le bouton pour vider le panier et la section formulaire.	OK
7	panier.html	L'utilisateur revient sur la page panier avec cette fois-ci deux caméras en deux exemplaires. Il clique sur le -1 sur chaque objet	panier.js : ligne 172 à 191	La quantité pour chaque objet est passée de 2 à 1. le sous-total de chaque objet est mis à jour, le total est mis à jour.	OK
8	panier.html	L'utilisateur actualise la page avec les deux caméras toujours présentes dans son panier	panier.js : ligne 197 à 203 ne s'exécutent pas	Rien ne doit changer	OK
9	panier.html	L'utilisateur supprime tous ses articles à l'aide de la fonction -1.	panier.js : ligne 193 à 195 (enlever un article); et 197 à 203	La page doit afficher "votre panier est vide !!" à la place des articles. Tout autre chose à part le header doit avoir disparu	OK
10	panier.html	L'utilisateur actualise la page après avoir vidé son panier avec la fonction -1	panier.js : condition ligne 197	La page doit seulement afficher "votre panier est vide !!"	OK
11	panier.html	L'utilisateur revient sur la page panier avec cette fois-ci deux caméras en deux exemplaires. Il retire la dernière caméra ajoutée du panier à l'aide de la fonction retirer du panier (icône poubelle)	panier.js : ligne 212 à 242	L'article seul doit disparaître. Seule une caméra doit rester dans le panier. le Total est mis à jour.	OK
12	panier.html	L'utilisateur actualise la page	panier.js : condition ligne 233 non exécutée	Rien ne doit changer	OK
13	panier.html	L'utilisateur supprime le dernier article présent à l'aide de la fonction "retirer du panier" (icône poubelle)	panier.js : ligne 233 à 238	Doit seulement afficher "votre panier est vide !!"	OK

14	panier.html	L'utilisateur actualise la page après avoir supprimé la dernière caméra du panier	panier.js : condition ligne 233 exécutée	Rien ne doit changer	OK
15	panier.html	L'utilisateur ajoute deux caméras deux fois et les supprime toutes les deux avec la fonction retirer du panier	panier.js : ligne 212 à 242	La page doit seulement affiché "votre panier est vide !!"	OK
16	panier.html	L'utilisateur actualise la page	panier.js : condition ligne 233 exécutée	Rien ne doit changer	OK
17	panier.html	L'utilisateur ajoute chaque caméra une fois dans le panier. Il décide ensuite de voyager vers toutes les pages produits via le lien présent sur le nom de caméra	panier.js : fonction urlDynamique();	Tous les liens doivent être valide et renvoyer vers la page produit de la caméra en question	OK
18	panier.html	L'utilisateur clique sur "vider son panier"	panier.js : ligne 247	Une boîte de confirmation doit apparaitre	OK
19	panier.html	L'utilisateur décide d'appuyer sur l'option "annuler" de la boîte de confirmation	panier.js : ligne 247	La page panier doit rester telle quelle.	OK
20	panier.html	Après annulation, l'utilisateur enlève un article avec la fonction -1, un autre avec la fonction retirer du panier	panier.js : ligne 171 à 205 et 212 à 242	Il doit rester trois articles, le total doit avoir été mis à jour.	OK
21	panier.html	L'utilisateur reclique sur vider le panier. Et clique sur OK lors de la confirmation.	panier.js : ligne 248	Une boîte d'alerte arrive "1, 2, 3, DJANGO !".	OK

22	panier.html	L'utilisateur clique sur OK lors de la boîte d'alerte	panier.js : ligne 250 à 259	Le panier est vide. la page affiche "Votre panier est vide !!"	OK
23	panier.html	L'utilisateur actualise la page	panier.js : ligne 263 à 267.	Rien ne doit changer	OK
24	panier.html	L'utilisateur revient sur la page avec au moins un article dans son panier et clique sur le bouton de soumission de formulaire vide.	panier.js : ligne 41 à 43	Une boîte de confirmation doit apparaître sans actualisation de la page	OK
25	panier.html	L'utilisateur clique sur "OK" lors de l'apparition de la boîte de dialogue	panier.js : ligne 99 à 105	Une boîte d'alerte "les données ne sont pas valides" doit apparaître.	OK
26	panier.html	L'utilisateur remplit chaque champ du formulaire avec une erreur. tel que: Nom : Dup0nt (contient un numéro); Prénom : J3an (contient un numéro); ville: 4rbre (contient un numéro); adresse: 01_rue du~colibri; (contient des caractères spéciaux non acceptés); email: aaa@bbb,ccc (contient une virgule).	regex.js : le fichier entier	A côté de chaque Label, le label + "invalidé" doit apparaître en rouge après le retraitement du focus du champ	OK
27	panier.html	L'utilisateur envoie son formulaire	panier.js : ligne 99 à 105	Une boîte d'alerte "les données ne sont pas valides" doit apparaître.	OK
28	panier.html	L'utilisateur vide tous les champs.	regex.js : le fichier entier	La mention invalide doit rester	OK
29	panier.html	L'utilisateur remplit 4 champs sur 5 et envoie le formulaire en acceptant la boîte de dialogue	panier.js : ligne 99 à 105	Une boîte d'alerte "les données ne sont pas valides" doit apparaître.	OK

30	panier.html	L'utilisateur remplit les 5 champs, clique sur "Valider" mais s'aperçoit d'une faute dans son adresse. Il clique alors sur annuler via la boîte de confirmation	panier.js : ligne 43 et 44	Une alerte "validation annulée" arrive.	OK
31	panier.html	L'utilisateur clique sur OK et modifie les champs erronés puis renvoie le formulaire	panier.js : ligne 43	La boîte de confirmation apparaît	OK
32	panier.html	L'utilisateur clique sur OK et envoie le formulaire	panier.js : ligne 48 à 93	Doit être redirigé vers la page confirmation.html	OK
33	confirmation.html	L'utilisateur arrive sur confirmation.html	panier.js : ligne 89 et 90; confirmation.js : tout le fichier	Un id de commande et le total payé doit apparaître.	OK
34	confirmation.html	L'utilisateur actualise la page	confirmation.js : tout le fichier	Rien ne doit changer	OK
35	confirmation.html	L'utilisateur refait la même commande	confirmation.js : ligne 1	Seul l'ID doit changer	OK
36	confirmation.html	L'utilisateur fait une commande différente pour un prix différent	confirmation.js : ligne 1 et 2	L'id et le prix doivent changer	OK
37	panier.html	Après avoir passé une commande, l'utilisateur revient vers le panier	panier.js : ligne 87	Le panier doit être vide	OK

38	index.html product. html panier.html	Le serveur est déconnecté	index.js : catch ligne 62; product.js : catch ligne 66; panier.js : catch ligne 244	Doit afficher une boîte d'alerte	OK
----	-----------------------------------------	---------------------------	-------------------------------------------------------------------------------------------	----------------------------------	----