Planification de tests unitaires

La planification a été réalisée pour couvrir un minimum de 80% du code pour le front-end.

Le plan est divisé en plusieurs sections :

- Test de communication avec l'API
- Affichage de l'ensemble des produits sur la page Accueil puis d'un seul produit sur une page Produit
- Interaction de l'utilisateur
- Enregistrement des produits ajoutés au panier et affichage sur la page panier
- Vérification de l'animation sur l'icône panier
- Vérification des champs du formulaire
- Affichage de la page confirmation

Le plan comporte un champ nature du test qui décrit le type de test à effectuer, un champ description qui explique le fonctionnement de celui-ci et un dernier champ qui indique le ou les fichiers concernés ainsi que le ou les numéros de lignes correspondant au code.

Test de communication avec l'API		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Vérifier la communication entre l'API et le site	Pour vérifier que la communication est établie, nous utilisons un système de promesses qui renvoie le **Status** de la requête. On affiche alors un retour dans un console.log avec un **.then** si la promesse est tenue ou dans un **.catch** s'il y a une erreur.	Les fichiers **main.js (L.31 et 33), produit.js (L.32 et 34)** et **panier.js (L.109 et 114)** sont testés.

Affichage de l'ensemble des produits sur la page Accueil puis d'un seul produit sur une page Produit		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Permettre l'affichage de l'ensemble des produits de la page d'Accueil	On réalise une requête avec la **méthode GET** pour récupérer la liste des produits, puis on utilise une boucle **forEach** pour parcourir le tableau que l'API fournie et afficher l'ensemble des éléments sous forme d'une liste.	Fichier **main.js (L.6 à 19).**
Permettre l'affichage d'un produit sélectionné sur une page dédiée avec ses infos de personnalisations	On réalise une requête avec la **méthode GET** au moment où l'utilisateur **clic** sur le produit, puis on utilise une fonction (**presentationCamera**) qui va parcourir le tableau du produit en sélectionné grâce à son ID et créer une page d présentation avec l'image, le nom, le prix et un bandeau de personnalisation.	Fichier **produit.js (L.13 à 21 et L.38 à 74).**

Interaction de l'utilisateur		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Permettre la redirection vers la page Produit au clic sur un produit depuis la page Accueil	Lorsque l'utilisateur est sur la page d'Accueil, il devra effectuer un **clic** sur un produit pour être redirigé vers une page incluant celui-ci. On utilise la classe Bootstrap « **streched-link** » et on attribue le numéro de l'ID de la caméra en question pour effectuer une redirection vers le produit ciblé.	Fichier **main.js (L.18).**

Afficher les options de personnalisations au clic sur le bandeau	Lorsque l'utilisateur est sur la page Produit, il pourra sélectionner une option de personnalisation au moyen d'un bandeau qui fera défiler les différentes possibilités. Le bandeau est défini par une classe Bootstrap « **browser-default** » et affiche le contenu des options grâce à une boucle **forEach** qui récupère les éléments depuis le tableau de l'API.	Fichier **produit.js (L.44 à 55).**
Ajouter un élément au panier	Lorsque l'utilisateur est sur la page Produit et qu'il **clic** sur le bouton « **Ajouter au panier** » on effectue une fonction (**addCart**) qui va ajouter le produit en écoutant **l'évènement au clic**.	Fichier **produit.js (L.116 à 138).**
Vider le panier	Si l'utilisateur souhaite vider son panier, il effectuera un **clic** sur le bouton « **Vider le panier** ». On écoutera **l'évènement au clic** et on effacera le **local Storage** afin de purger l'ensemble des éléments dans le panier.	Fichier **panier.js (L.30 à 32).**
Passer commande	Lorsque l'utilisateur aura correctement compléter les informations demandées dans les champs du formulaire, il devra effectuer un **clic** sur le bouton « **Passer commande** ». On écoutera **l'évènement Submit** qui permettra l'envoi des données du formulaire et des produits.	Fichier **panier.js (L.95).**

Enregistrement des produits ajoutés au panier et affichage sur la page panier		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Enregistrer les produits	Lorsque l'utilisateur est sur la page Produit et qu'il **clic** sur le bouton « **Ajouter au panier** » on effectue une fonction (**cameraInStorage**) qui va créer un tableau contenant les informations du produit sélectionné et avec une condition **if** on vérifiera si le produit est déjà présent dans le panier, et si celui-ci est déjà enregistré, on utilisera une boucle **forEach** pour incrémenter le tableau le tableau.	Fichier **produit.js (L.78 à 96).**
Afficher les produits enregistrés sur la page Panier	Lorsque les produits ont été ajoutés au panier, on récupère les infos sur la page Panier avec une fonction (**cart**) qui va afficher les infos contenues dans le **local Storage**.	Fichier **panier.js (L.2 à 15).**
Afficher le prix total	Le total de chaque produit en fonction de la quantité sera obtenu par la multiplication de la « **Quantité** » et de son « **Prix** ». Le prix total de la commande sera obtenu par l'addition des lignes « **td** » intitulé « **Total** ».	Fichier **panier.js (L.14 à 27).**

Vérification de l'animation sur l'icône panier		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Affichage du nombre de produit sur le l'icône panier	Lors de l'ajout d'un produit dans le panier, on récupère le nombre déjà présent dans celui-ci garce à deux fonctions. La première récupère et incrémente la valeur puis la stocke dans le **local Storage** (**numberInCart**), et la seconde lit le nombre de produit stocké (**cartNumbers**).	Fichier **produit.js (L.100 à 111).** Fichier **animationPanier.js**

Vérification des champs du formulaire		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Affichage d'une erreur si non-respect du format du champ	Lorsque l'utilisateur saisi les informations sur le formulaire, il doit respecter les champs de celui-ci comme indiqué dans les **placeholder.** On vérifie les informations saisies en écoutant **l'évènement keyup** dans une boucle **for** et conditionne avec un **if** pour retourner une **animation bordure rouge** sur le champ si celui-ci est mal rempli. On bloque également l'envoi du formulaire si les conditions ne sont pas remplies.	Fichier **panier.js (L.45 à 54).** Fichier **panier.html** (**L.75 à 103**).

Affichage de la page confirmation		
Nature du test	**Description du test**	**Fichier et numéro de ligne du code**
Récupérer l'ID des produits et les valeurs saisies dans le formulaire	Pour obtenir un numéro de confirmation et orienter vers la page Confirmation, on récupère l'ID des produits présents sur la page panier qu'on conditionne avec un **if** pour vérifier que la page contienne un produit puis on effectue une boucle **for** pour récupérer les ID dans un tableau. On récupère également les valeurs du formulaire dans un **objet contact**.	Fichier **panier.js (L.65 à 67 et L.99 à 104).**
Envoi des données à l'API et récupération d'un ID de confirmation de commande	L'utilisateur envoi les données à l'API avec la **méthode POST** appelé par la fonction (**post**) à l'écoute de **l'évènement Submit**. Si les données sont correctes, on reçoit un ID de confirmation. Une **promesse** nous retourne un message dans un **console.log** qui indique l'envoi du formulaire sous le numéro ID.	Fichier **panier.js (L.72 à 109).**
Redirection vers la page Confirmation	L'utilisateur est redirigé vers la page confirmation si la **promesse** est tenue par la fonction (**post**).	Fichier **panier.js (L.110 à 112).**
Affichage de la page confirmation	La page confirmation affiche un message ainsi que le « **prix total** » et **l'ID de confirmation de commande** récupéré depuis le **local Storage.** L'utilisateur peut alors retourner sur la page Accueil, on écoutera **l'évènement au clic** ou la fonction (**setTimeout**) renverra directement sur la page Accueil au bout de quelques secondes.	Fichier **confirmation.js**