**Projet 5 du parcours Développeur web d'OpenClassrooms :**

**Construire un site e-commerce ORINOCO**

Il s'agit d'un MVP d'une application thématique ne vendant qu'un seul type de produits (ici des appareils photos) ; la partie back-end est fournie et l'objectif est de créer la partie front-end par l’intermédiaire d'une API fournie.

**Compétences évaluées :**

* Interagir avec un web service avec JavaScript
* Créer un plan de test pour une application
* Valider des données issues de sources externes
* Gérer des événements JavaScript

**Prérequis :**

* Cloner ce repository : https://github.com/OpenClassrooms-Student-Center/JWDP5 et créer un dossier back-end.
* Se placer dans le dossier back-end.
* Installer **Node.js** et **npm**.
* Lancer **npm** avec la commande "***npm install***" pour installer les dépendances
* Taper la commande "***node server.js***" pour lancer le serveur.
* Une fois le serveur lancé, écrire l'url de l'API : <http://localhost:3000/api/cameras>

**Spécificités techniques :**

*L'application web doit être composée de 4 pages :*

* une page de vue sous forme de liste, montrant tous les articles disponibles à la vente.
* une page “produit”, qui affiche de manière dynamique l'élément sélectionné par l'utilisateur et lui permet de personnaliser le produit et de l'ajouter à son panier.
* une page “panier” contenant un résumé des produits dans le panier, le prix total et un formulaire permettant de passer une commande. Les données du formulaire doivent être correctes et bien formatées avant d'être renvoyées au back-end. Par exemple, pas de texte dans les champs date.
* une page de confirmation de commande, remerciant l'utilisateur pour sa commande, et indiquant le prix total et l'identifiant de commande envoyé par le serveur.

**Information complémentaires :**

Pour le MVP, la personnalisation du produit ne sera pas fonctionnelle : la page

Contenant un seul article aura un menu déroulant permettant à l'utilisateur de choisir une option de personnalisation, mais celle-ci ne sera ni envoyée au serveur ni reflétée dans la réponse du serveur.

Le code source devra être indenté et utiliser des commentaires. Il devra également utiliser des fonctions globales.

Concernant l’API, des promesses devront être utilisées pour éviter les rappels.

Les inputs des utilisateurs doivent être validés avant l’envoi à l’API.

**Outils utilisés :**

* VISUAL STUDIO
* HTML / CSS3 / SASS / JAVASCRIPT
* GIT/GITHUB