

Travail d'intégration

Javascript - React
CF2M - 2024 - Webdev

Simulateur de boutique en ligne | T.I :

Objectifs :

Créer une application web de simulation de boutique en ligne où l'utilisateur peut :

- Parcourir un catalogue de produits.
- Ajouter des articles à son panier.
- Compléter une commande avec un formulaire de confirmation.

L'application doit être dynamique grâce à l'intégration de JavaScript, et respecter les bonnes pratiques de séparation des fichiers (HTML, CSS, JavaScript).




1. Fonctionnalités (50 points)

a) Catalogue de produits (25 points)

- Afficher une liste d'articles avec :
 - Les informations principales : prix, description, image, et quantité restante.
 - Une icône pour sélectionner un article et une icône pour afficher les détails de chaque produit.
 - Lorsqu'un article est sélectionné, l'utilisateur peut l'ajouter au panier.

b) Panier d'achat (25 points)

- Les articles ajoutés apparaissent dynamiquement dans le panier, avec :
 - Le nombre total d'articles et le prix total mis à jour automatiquement à chaque ajout.
 - Un bouton "Commander" qui redirige l'utilisateur vers un formulaire de confirmation.
 - Le formulaire doit permettre à l'utilisateur d'entrer son adresse et de soumettre la commande.
 - Une fenêtre modale doit s'afficher pour confirmer la réception de la commande une fois le formulaire soumis.
- 

2. Mise en page (20 points)

- Utiliser un fichier CSS séparé ou un framework CSS de votre choix (par exemple Bootstrap, Tailwind).
- L'interface doit être intuitive et facile à utiliser :
 - Design propre et responsive (adapté aux différentes tailles d'écran).
 - Interface claire, chaque élément doit être bien organisé et interactif.

3. Contenu (20 points)

- Utiliser un fichier JSON pour stocker les données des produits. Ce fichier JSON doit contenir les informations sur chaque produit, telles que l'ID, le nom, le prix, la description, l'image, et la quantité restante.
- Le contenu du fichier JSON doit être utilisé pour afficher dynamiquement les produits sur la page.

4. “Méthodes” (10 points)

- Utiliser *fetch()* pour récupérer les informations du JSON
- Utiliser *localStorage* pour stocker les produits dans le panier et autres choix de l'utilisateur

5. Total /120 (20 points bonus possible)





Méthodologie :

Structure de l'application :

a) Catalogue de produits (HTML, CSS, JSON, JS)

- HTML :
 - Créez une section dédiée au catalogue des produits avec des cartes pour chaque article.
 - Chaque produit doit afficher les informations suivantes : image, nom, description, prix, et quantité restante. Ajoutez également un bouton pour afficher les détails du produit et un bouton pour ajouter l'article au panier.
- CSS :
 - Stylisez le catalogue pour que chaque produit soit bien mis en valeur. Utilisez un fichier CSS séparé.
- JSON :
 - Créez un fichier JSON ou une variable contenant les produits. Chaque produit doit inclure un ID, un nom, une description, un prix, une quantité restante, et une image.
- JS :
 - Créez un script qui vous permettra de récupérer les datas du JSON.
 - Créez des scripts pour gérer les interactions avec les articles du catalogues et votre panier (détail & ajout).
 - Gérer les interactions avec la BDD.

b) Panier d'achat (HTML, CSS, JS)


- HTML :
 - Créez une section dédiée au panier sur la page, où les articles ajoutés apparaîtront.
 - Ajoutez un bouton "Commander" qui redirige l'utilisateur vers un formulaire de confirmation.
- CSS :
 - Stylisez le panier pour qu'il soit bien visible et lisible. Utilisez des marges, des bordures, et assurez-vous que le prix total soit mis en valeur.
- JavaScript :
 - Utilisez JavaScript pour gérer les interactions entre les produits et le panier. Créez des fonctions qui :
 - i. Permettent d'ajouter des articles au panier.
 - ii. Mettent à jour le prix total à chaque ajout d'article.
 - iii. Affichent dynamiquement le contenu du panier.





Déploiement :

Pour finir :

- Rendez votre projet accessible sur Github.
 - Le site doit être responsive, développez le en utilisant l'approche *"mobile-first"*.
 - Pensez à mettre votre repo à jour au fur et à mesure du projet (minimum 3 push).
 - *(Bonus) : Lier le projet à une BDD (MySQL, Firebase, ...) pour stocker les articles, les achats et tenir les stocks à jour.*
 - *(Bonus) : Créer un système de connexion/inscription à la boutique.*
 - *(Bonus) : Faire le projet avec Vite / Vite + React*
 - *(Bonus) : Effectuez un déploiement via Github ou votre propre serveur.*
- 



Bon courage !

Deadline / présentation :

Vendredi 13h