Page 1

Étapes clés – Projet 4 du Développeur web web mobile

Implémentez un site e-commerce dynamique

Besoin d'indications pour avancer dans votre projet ? Suivez les étapes clés proposées ci-dessous !

Recommandations générales

Vous allez écrire du code JS qui sera découpé en plusieurs fonctions. L'idée n'est pas de commenter chaque ligne de code, mais de commenter les choses de façon plus générale. Pour cela, il est possible de commenter le code, fonction par fonction, grâce m JSDoc. Pour vous aider, lisez l'article Bien commenter son code.

Étape 1 : Prendre en main les maquettes HTML / CSS

5 % d'avancement

Avant de commencer, il est essentiel de prendre connaissance des éléments mis m disposition, notamment les différentes pages web mises en place.

☐ **Une fois cette étape réalisée, vous aurez :**

- connaissance du fonctionnement / de la structuration des pages HTML / CSS.

☐ **Recommandations :**

- Affichez les 4 pages HTML fournies dans votre navigateur.
- Essayez d'afficher les éléments HTML commentés par Frank dans le code, ceci dans le seul et unique but de voir m quoi vont ressembler les choses m la fin. En effet, avant d'insérer des éléments HTML de façon dynamique dans le DOM grâce m JS, il est nécessaire de savoir quels éléments nous allons devoir insérer.

Points de vigilance :

- Attention m bien analyser les pages HTML / CSS fournies. Tout a été prévu pour que vous n'ayez pas m y retoucher, inutile de recréer des éléments déjm conçus.

Page 2

Étape 2 : Manipuler l'API

10 % d'avancement

Avant d'aller plus loin, il est essentiel de prendre connaissance des autres éléments mis m disposition, tels que l'API avec laquelle il va falloir travailler dans ce projet.

- **Une fois cette étape réalisée, vous aurez** :
  - connaissance du fonctionnement / de la structuration de l'API réalisée par Bilal.

- **Recommandations :**
  - Bien prendre le temps de lire le ReadMe du repo GitHub fourni !
  - Une fois l'API lancée, consultez l'URL renseignée dans les spécifications fonctionnelles et techniques du projet, dans le but de vérifier le bon fonctionnement de celle-ci. L'idée ici est de voir, depuis votre navigateur par exemple, le JSON retourné par l'API, que ce soit lorsqu'on demande m l'API l'ensemble des produits, ou bien seulement un produit précis (via son id).

Points de vigilance :
- Attention m bien lancer l'API avant de vouloir l'utiliser.

Étape 3 : Insérer les produits dans la page d'accueil

20 % d'avancement

Vous pouvez maintenant intégrer l'ensemble des produits de l'API dans la page du site web.

- **Une fois cette étape réalisée, vous aurez** :
  - une page d'accueil contenant les produits de l'API.

- **Recommandations :**

- En JS, commencez par requêter l'API pour lui demander l'ensemble des produits ; récupérer la réponse émise, et parcourir celle-ci pour insérer chaque élément (chaque produit) dans la page d'accueil (dans le DOM).

Points de vigilance :

- Attention, ici il est question d'afficher les produits de façon dynamique, pas de façon statique.
- Attention de bien utiliser l'ensemble des éléments nécessaires pour chaque produit, on a vite fait d'oublier d'utiliser les textes alternatifs, par exemple.

▭ **Ressources :**

- La partie 2 (Communiquez via une API avec un service web) du cours "Écrivez du JavaScript pour le Web" devrait vous aider m récupérer les données depuis l'API.
- Le chapitre "Utilisez la bonne boucle pour répéter les tâches (for, while)" du cours "Apprenez m programmer avec JavaScript" devrait vous aider m parcourir la réponse envoyée par l'API.
- Vous pourrez aussi trouver des informations intéressantes dans le chapitre Modifiez le DOM du cours "Écrivez du JavaScript pour le Web".

Étape 4 : Faire le lien entre un produit de la page d'accueil et la page Produit

30 % d'avancement

Avant de penser m la page Produit, il va falloir prévoir ce qu'il faut sur la page d'accueil pour que, une fois sur la page Produit, vous puissiez savoir lequel des différents produits de l'API il faut afficher.

▫ **Une fois cette étape réalisée, vous aurez** :

- la possibilité d'ouvrir une page Produit en sachant quel produit afficher.

▭ **Recommandations :**

- Renseignez-vous sur le terme "URLSearchParams". C'est grâce m cette notion que votre page Produit va pouvoir "savoir" lequel des différents produits de l'API afficher.
- Pour chacun des produits de la page d'accueil, il va falloir bien paramétrer la balise "a" et son attribut "href".

Points de vigilance :

- Attention m bien utiliser URLSearchParams pour passer l'id d'une page m une autre, et non pas localStorage.

▭ **Ressources :**

- Voici un court article mais relativement clair sur URLSearchParams : Comment récupérer les paramètres d'URL en JavaScript.
- La documentation MDN sur URLSearchParams.

Étape 5 : Récupérer l'id du produit m afficher

35 % d'avancement

Avant de pouvoir afficher les détails d'un produit, il va falloir savoir de quel produit on parle ; nous allons donc récupérer l'id du produit ayant été cliqué sur la page d'accueil.

▭ **Une fois cette étape réalisée, vous aurez :**

- la connaissance du produit m afficher sur la page Produit.

▭ **Recommandations :**

- Vous êtes maintenant en mesure de savoir lequel des produits de l'API nous allons vouloir afficher dans la page Produit. Il va donc falloir récupérer l'id du produit en question dans l'URL (URLSearchParams).

▭ **Ressources :**

- Comme lors de l'étape précédente, l'article sur URLSearchParams vous sera utile : Comment récupérer les paramètres d'URL en JavaScript.

Étape 6 : Insérer un produit et ses détails dans la page Produit

45 % d'avancement

Nous avons maintenant l'id du produit m afficher, ceci permettant de requêter l'API dans le but de récupérer les différentes informations du produit en question.

- **Une fois cette étape réalisée, vous aurez :**
  - une page Produit complétée, m partir des données de l'API.

- **Recommandations :**
  - Interroger l'API pour récupérer les détails du produit.
  - Insérer ces détails dans la page Produit (dans le DOM).

  Points de vigilance :
  - Attention de bien utiliser l'API, l'idée ici est de récupérer un seul et unique produit, et non pas l'ensemble des produits.

- **Ressources :**
  - Se référer aux spécifications fonctionnelles et techniques du projet pour savoir comment requêter l'API.

Étape 7 : Ajouter des produits dans le panier

55 % d'avancement

La page Produit est en place, celle-ci affiche les détails d'un produit cliqué m partir de la page d'accueil. Il faut maintenant gérer la possibilité d'ajouter ce produit au panier.

- **Une fois cette étape réalisée, vous aurez :**
  - la possibilité d'ajouter des produits dans votre panier.

- **Recommandations :**
  - Techniquement parlant, le panier peut être un array qui contiendrait trois choses :
    - l'id du produit ;

- la quantité du produit ;
- la couleur du produit.

- Il est nécessaire d'utiliser localStorage pour pouvoir accéder m cet array depuis la page Panier.
- Lorsqu'on ajoute un produit au panier, si celui-ci n'était pas déjm présent dans le panier, on ajoute un nouvel élément dans l'array.
- Lorsqu'on ajoute un produit au panier, si celui-ci était déjm présent dans le panier (même id + même couleur), on incrémente simplement la quantité du produit correspondant dans l'array.

Points de vigilance :

- Dans localStorage, attention de ne pas multiplier inutilement des éléments identiques.

▭ **Ressources :**

- Vous devriez lire cet article sur localStorage. Voici également la documentation MDN m ce sujet : Window.localStorage

Étape 8 : Afficher un tableau récapitulatif des achats dans la page Panier

65 % d'avancement

Les produits sont ajoutés au panier, mais cela reste encore invisible pour l'utilisateur. Dans cette étape, nous allons afficher le contenu du panier dans la page Panier.

▭ **Une fois cette étape réalisée, vous aurez :**

- une page Panier affichant tous les articles précédemment ajoutés.

▭ **Recommandations :**

- Depuis la page Panier, récupérer le panier (l'array) via localStorage.
- Parcourir l'array.
- Créer et insérer des éléments dans la page Panier.

Points de vigilance :

- Attention de ne pas dupliquer inutilement les éléments dans le tableau récapitulatif (le panier). S'il y a plusieurs produits identiques

(même id + même couleur), cela ne doit donner lieu qu'm une seule ligne dans le tableau.

▭ **Ressources :**

- À nouveau, le chapitre "Utilisez la bonne boucle pour répéter les tâches (for, while)" du cours "Apprenez m programmer avec JavaScript" devrait vous aider.

Étape 9 : Gérer la modification et la suppression de produits dans la page Panier

75 % d'avancement

Les produits présents dans le panier sont affichés sur la page Panier. Maintenant, il faut permettre m l'utilisateur de modifier la quantité ou de supprimer un produit dans le panier.

▭ **Une fois cette étape réalisée, vous aurez :**

- la possibilité, sur la page Panier, de modifier la quantité ou de supprimer un produit.

▭ **Recommandations :**

- Concernant la modification, il va falloir recourir m l'événement de modification (addEventListener de type change) pour observer le changement de la quantité.
- Aussi, la méthode Element.closest() devrait permettre de cibler le produit que vous souhaitez supprimer (où dont vous souhaitez modifier la quantité) grâce m son identifiant et sa couleur.

Points de vigilance :

- Attention m bien penser m modifier le DOM, mais aussi localStorage, sinon les modifications effectuées dans le panier ne seront pas conservées en cas de changement de page / de rafraîchissement de la page.
- Pour récupérer l'ID du produit et modifier sa quantité ou le supprimer, **ne répétez pas les données** du produit dans les éléments enfants. Récupérez plutôt le *data-id* et le *data-color* dans l'élément parent (l'article *cart_item*) grâce m Element.closest(). Ainsi, les données du produit sont centralisées m un seul endroit du code : l'élément englobant toutes les informations du produit.

- **Ressources :**
  - Cette documentation sur la propriété dataset pourrait vous faciliter un peu les choses.
  - Voici un article parlant de l'utilisation de addEventListener de type change.
  - La documentation MDN de la méthode Element.closest().

Étape 10 : Passer la commande

85 % d'avancement

Nous avons presque terminé, l'utilisateur doit pouvoir valider sa commande, c'est l'objectif de cette étape.

- **Une fois cette étape réalisée, vous aurez :**
  - la possibilité, sur la page Panier, de saisir vos coordonnées puis de confirmer votre commande.

- **Recommandations :**
  - Récupérer et analyser les données saisies par l'utilisateur dans le formulaire.
  - Afficher un message d'erreur si besoin (par exemple lorsqu'un utilisateur renseigne "bonjour" dans le champ "e-mail").
  - Constituer un objet contact (m partir des données du formulaire) et un tableau de produits.

  Points de vigilance :
  - Attention m bien vérifier les données saisies par l'utilisateur.
  - Lors de la vérification des données via des regex, attention m bien mener des tests pour vérifier le bon fonctionnement des regex.
  - Ne pas oublier d'afficher un message d'erreur si nécessaire.

- **Ressources :**
  - La partie 2 (Communiquez via une API avec un service web) du cours "Écrivez du JavaScript pour le Web" devrait vous aider m envoyer/récupérer des données m/depuis l'API.
  - L'article Introduction aux expressions régulières ou expressions rationnelles en JavaScript devrait vous permettre de mieux

comprendre comment vérifier les données saisies par un utilisateur. Les regex peuvent parfois être complexes m écrire, ne pas hésiter m mener des recherches m ce sujet (exemple : expressions régulières JavaScript email).
- Pour savoir comment faire une requête POST en JavaScript, lisez la documentation sur ce sujet.
- Se référer aux spécifications fonctionnelles et techniques du projet pour savoir comment requêter l'API.

Étape 11 : Afficher le numéro de commande

90 % d'avancement

Nous voilm au bout des choses, maintenant que nous pouvons passer commande, il ne reste plus qu'm afficher le numéro de ladite commande.

▭ **Une fois cette étape réalisée, vous aurez :**
- après confirmation de la commande, un affichage du numéro de commande en question.

▭ **Recommandations :**
- Effectuer une requête POST sur l'API et récupérer l'identifiant de commande dans la réponse de celle-ci.
- Rediriger l'utilisateur sur la page Confirmation, en passant l'id de commande dans l'URL, dans le but d'afficher le numéro de commande.
- Si ce numéro doit être affiché, celui-ci ne doit pas être conservé / stocké.

Points de vigilance :
- Attention m bien vérifier par deux fois la requête attendue par l'API, aucune erreur ne sera tolérée par celle-ci.
- Attention, si le numéro doit être affiché, celui-ci ne doit pas être conservé / stocké.

Étape 12 : Mettre en place le plan de test d'acceptation

100 % d'avancement

Le site web étant en place, il est maintenant l'heure de mettre en place un plan de test dans le but de vérifier que toutes les fonctionnalités ont bien été implémentées, et que celles-ci sont fonctionnelles.

- **Une fois cette étape réalisée, vous aurez :**
  - un plan de test d'acceptation complété.

- **Recommandations :**
  - Maintenant que l'ensemble du code JS est écrit, il faut mettre en place le plan de test d'acceptation. L'idée principale est de vérifier l'alignement entre le cahier des charges / les spécifications fonctionnelles, et le produit réalisé.

Points de vigilance :
- Attention de ne pas oublier de fonctionnalités dans l'écriture des tests.

- **Ressources :**
  - Vous pouvez trouver de nombreuses documentations concernant les différents tests, et notamment les tests d'acceptation. Voici un article sur les tests d'acceptations.

Projet terminé !