**TP E-commerce en React**

Objectif :

Créer un front-end fonctionnelle en React, utiliser une API pour collecter des informations.

Technologie :

Le projet est en **React** avec une installation avec **Vite.**

Les données proviennent de Fake Store API : <https://fakestoreapi.com/docs>

Les librairies/ Framework à utiliser seront :

* + Formik et Yup pour la création et contrôle de formulaire
  + Axios pour les attaques d’API
  + Tailwind CSS en tant que framework de CSS
  + D’autres librairies pourront être rajouté au choix des développeurs

Outils Obligatoire : GIT avec GITHUB (un repo par groupe), Node,

react, react-router-dom

Outils Recommandé : Figma, Miro, Trello, Méthodologie Agile SCRUM, VSCode, Postman, localstorage

# Besoin :

Les parties à réalisées sont uniquement coté “Utilisateur”,

Le coté “admin” et “vendeur” ne sont pas à réaliser.

* **Page d’accueil :** Mettant en avant les différents produits disponibles :

- catégories (mode homme, mode femme, technologies, bijoux...)

- 5 produits les mieux notés

* **Page inscription :** permet à l’utilisateur de créer un compte, la structure de la donnée à renvoyer dans être comme ci-dessous :

{{

"email":"John@gmail.com",

"username":"johnd",

"password":"m38rmF$",

"name":{

"firstname":"John",

"lastname":"Doe"

},

"address":{

"city":"Lille",

"street":"rue de l'église",

"number":3,

"zipcode":"59000"

}

}

Si vous envoyez une donnée comme ci-dessus, il vous renverra un objet avec un nouvel identifiant. Rien de réel ne sera inséré dans la base de données, donc si vous voulez accéder au nouvel identifiant, vous obtiendrez une erreur 404.

* **Page Connexion :** permet à l’utilisateur de connecter :

{{

"username":"johnd",

"password":"m38rmF$",

}

Il faut utiliser les comptes déjà existant dans l’api : <https://fakestoreapi.com/users>

Créez un système de mot de passe oublié : envoyer une requete de modification, il n’y aura pas de modification dans la base de donnée de l’API

* **Page recherche :** accessible si on fait une recherche par mot clé ou au click d’une catégorie.

Permet d’avoir accès à des filtres :

- par prix ex : 0 à 50€, 50 à 100€ …. (Exemple de fourchette)

- par note min et max : 1,2,3,4,5 etoiles

* **Page panier** : voir les articles ajouter dans le panier, possibilité d’ajouter un article, changer le nombre d’articles, supprimer un article du panier.

Appel API seulement pour récupérer les articles commandés par l’utilisateur en connecté, mais modification du panier enregistré par dans une variable (state)

Attention : Le prix doit changer également.

La validation du panier doit rediriger vers la page paiement

* **Page paiement :** dois afficher l’adresse de facturation (adresse du compte de l’utilisateur connecté),
  + Possibilité de modifier l’adresse, envoyez une requête de changement d’adresse (la modification dans la base de données restera fictive mais enverra un message de modification)

Simulez une page de paiement : codes bancaire et /ou paypal

Valider la transaction (message de validation sans passer par l’API)

* **Page Profil** : Voir les informations du compte et pouvoir les modifier

Les modifications sont à envoyer vers l’API, mais pas de modification dans l’API

* **Page Détail produit :** Détail de toutes les informations du produit :
  + Titre/ intitulé du produit
  + Image
  + Description
  + Catégorie
  + Prix
  + Note moyenne avec nombre de notes

Il doit avoir un bouton pour ajouter au panier (changement du nombre d’articles au logo panier, pas d’appel API nécessaire)

* **Header** (le même sur tout le site) **:** 
  + Logo de votre site
  + Champ de saisie de recherche titre/intitulé par mot clé avec des propositions sous forme de menu déroulant
  + Lien vers Panier
  + Un bouton “Connectez-vous” permettant d’avoir accès à l’Inscription et connexion (mode hors ligne), qui sera remplacé par un avatar permettant de voir son Profil et de se déconnecter (mode connecté)
* **Footer** (le même sur tout le site) :

Ajoutez ce que vous souhaitez : copyright, accès aux réseaux sociaux...

Caractéristiques :

* Le code doit être **commenté !**
* **Le css, les images** et les **polices d’écriture** doivent être dans le dossier **assets** avec un dossier pour chaque type d’éléments

ex: les images dans un dossier **images**

* Les **composants** sont à répartir selon leur utilisation :
  + S’ils sont accessibles par les routes : dossier **pages**
  + S’ils sont importés par d’autres composants : dossier c**omponents**
* Les projets seront mis sur **GITHUB**
  + Il y aura plusieurs branches :
    - **Main/Master** : branche de production
    - **Une branche par développeur :** chaque développeur devra commit et push leur travail quotidiennement.

SI fonctionnalité terminer et testé merger sur la main/master

* Le projet commencera par une réalisation d’une maquette rapide :

Bref aperçu de votre site (charte graphique, couleur, logo...).

* Le projet doit être livré avant le vendredi à 9h, il doit être présent dans la branche main/master de github (le code présent uniquement dans les branches des développeurs sera ignorés à la livraison)

4 documents seront fournis :

* Un cahier de charges (le document sur lequel vous vous trouvez)
* Un diagramme de case d’utilisation
* Un Wireframe
* Un site map