

Évaluation 3

582-B54-RI - INTÉGRATION D'INTERFACE UTILISATEUR



Date de remise : dimanche 1er février 2026 à 23h55

Travail à faire individuellement

I. Travail à faire

Vous devez concevoir à l'aide de Vue.js un site web de vente en ligne. Le site web devra permettre de faire des achats.

Le site devra être fait avec la Composition API.

La personne utilisant votre site pourra :

- Afficher les articles
- Trier les articles par catégorie
- Voir les détails d'un élément choisi
- Se créer un compte/ s'authentifier

Votre site devra communiquer avec l'API « fake store API »

<https://fakestoreapi.com/>

Dans la documentation de l'API, vous aurez les détails pour différentes URLs.

II. Présentation du site

1. Barre de navigation

Votre site doit avoir une barre de navigation permettant d'aller sur chacune des pages principales du site.

2. Page d'accueil

Cette page doit présenter votre site web. Elle doit présenter les différentes catégories de produits.

On peut avoir les catégories au lien suivant :

<https://fakestoreapi.com/products/categories>

3. Page de la liste des produits

Vous devez avoir une deuxième page permettant d'afficher tous les produits de l'API.

Les informations devront être affichées sous forme de grille.

Les cartes doivent comporter les informations suivantes :

- L'image du produit
- Le titre
- Le prix
- L'évaluation (rate)
- La catégorie

Enfin, sur cette page, vous devez permettre de trier les articles par catégories. Vous devez donc avoir une liste déroulante permettant de choisir une catégorie et afficher tous les articles de cette catégorie.

On doit pouvoir revenir sur l'affichage complet des articles.

4. Page de détails

Lorsqu'on clique sur une carte dans la page liste, on doit voir un « zoom » sur cet objet avec plus de détails.

Sur cette page, on doit voir toutes les informations de l'article, y compris la description.

On doit pouvoir ici ajouter l'article au panier.

Cependant, la possibilité d'ajouter un article au panier ne doit se faire **que si le client est authentifié.**

On peut avoir les détails d'un produit via l'URL <https://fakestoreapi.com/products/{id}>

Exemple pour le produit ayant l'id 1 : <https://fakestoreapi.com/products/1>

Une fois un produit ajouté au panier, le nombre de produit du panier doit s'afficher dans la barre de navigation.

5. Page de création de compte

Vous devez permettre à un client de se créer un compte.

On devra entrer le nom et le prénom, l'email et un mot de passe.

Les informations du membre devront être conservées au niveau de **Firebase de Google**.

6. Authentification du client

À tout moment, le client pourra s'authentifier (authentification Firebase de Google) en entrant son email et son mot de passe.

Seul un client authentifier pourra effectuer des achats.

Une fois authentifié, le nom du client apparaitra dans la barre de navigation ainsi que l'option du panier d'achat.

7. Panier d'achat

La page du panier d'achat doit être disponible **uniquement si le client est authentifié**.

Si le panier est vide, une image doit être affichée. S'il y a des articles, on doit voir la liste.

Pour chacune des cartes, on doit voir :

- Le titre
- La description
- Le prix
- La quantité

Sur cette page, on doit pouvoir également le sous-total de l'achat, les taxes et enfin le prix total.

On doit pouvoir également retirer un article du panier d'achat.

Avant le retrait, un message de confirmation doit apparaître.

Une fois le retrait confirmé, le sous-total, les taxes et le prix total doivent se mettre à jour, ainsi que le nombre de produits dans la barre de navigation.

III. Organisation du code

1. Mise en place d'un store

Vous devez mettre en place un store avec Pinia.

Votre store aura la charge de conserver :

- Le panier d'achat de l'utilisateur authentifié
- Les informations de session du client authentifié (session)

Bien entendu, les données conservées dans le store sont celles utilisées pour le bon fonctionnement de tout votre site.

2. Gestion des routes

Vous devez mettre en place les routes.

Vous devez également mettre en place les gardes de routes afin de ne pas pouvoir accéder aux pages qui ne sont pas autorisées hors connexion.

3. Utilisation de composable

Vous devez avoir un composable dans lequel toutes les requêtes vers l'API vont se faire.

Il ne doit pas avoir de requête ailleurs dans votre projet.

4. Les composants

Votre travail doit comporter les éléments suivants :

- Des composants pour chacune des pages
- Les cartes doivent provenir d'un composable dédié à cet effet

IV. Critères de correction

- Le site doit comporter tous les éléments demandés plus haut
- Le site devra être esthétique et responsive
- Le site n'a pas d'erreur dans la console

V. Remise du travail

Vous devez remettre votre travail dans un dossier compressé dans les délais indiqués.

Tout travail remis en retard ne sera pas accepté.

Afin d'alléger votre fichier compressé, vous pouvez supprimer le dossier « node_modules » à l'intérieur du fichier compressé. Il sera réinstallé lors de la correction.

VI. Grille de correction

Barre de navigation	4
Page d'accueil	
Présentation de votre site	3
Catégories des produits	3
Page de liste des produits	
Affichage de tous les produits	5
Affichage sous forme de grille	2
Carte avec informations demandées	5
Tri par catégorie	4
Retour à l'affichage complet	2
Page de détails de produit	
Informations complètes du produit	5
Ajouter au panier	5
Indication de la quantité	5
Bouton ajouter disponible si authentifié	2
Page panier	
Accès seulement si authentifié	4
Image si panier vide	2
Liste des produits avec informations demandées	5
Calcul du sous-total, des taxes et du total	4
Retrait d'un produit du panier	4
Message de confirmation	2
Mise à jour de la barre de navigation	2
Authentification Firebase	
Création compte Firebase	4
Authentification Firebase	4
Nom affiché dans la barre de navigation	2
Organisation du code	
Création et utilisation du store	5
Création et utilisation du composable	3
Création et utilisation des routes	4
Gardes de navigation	2
Expérience utilisateur	
Esthétique du projet	5
Site responsive	3
TOTAL	100