

Configurateur de montres

Sur le modèle du site Deejo :

<https://my.deejo.fr/fr/37/titanium/olive/none/none/tree?currency=EUR>

Ce projet vise à développer un site web dédié à la vente en ligne de montres personnalisables. Votre site web doit permettre d'afficher une montre en 3D avec des options paramétrables permettant de modifier le prix de l'achat.

Partie Front:

Inventaire du fichier 3D "montre.dae"

- tous les modèles 3D composant toutes les versions possibles de montres :
 - Boitier (deux formes de cadran, pouvant recevoir des textures de fond)
 - boitier_carre
 - boitier_rond
 - bouton
 - Les aiguilles
 - aiguille_heures
 - aiguille_minutes
 - aiguille_secondes
 - pierres précieuses (pour décorer le tour du cadran), effets de transparence et de brillance
 - pierre
 - Une géométrie de bracelet pouvant recevoir plusieurs textures (métal, cuir, tissus)
 - bracelet
 - fermoir

Inventaire des textures

- Les textures : **background_XXX.png** sont des fonds de boitier utilisables sur les deux types de boîtier.
- Les textures : **texture-XXX.jpg** sont des images de matériaux pour le bracelet.

A faire :

- Chargement des modèles 3D, assemblage algorithmique en plaçant plusieurs pierres sur la montre.
- Création d'un décor (par exemple sol, avec ombre de l'objet...)
- Eclairage et brume
- Animation permanente :
 - rotation des aiguilles en fonction de l'heure (☆)
- Choix et changement de la texture de fond d'afficheur (au moins 3) (☆)
- Interactions :
 - La rotation de l'objet à la souris permet de mieux visualiser sa géométrie
 - L'heure affichée en permanence sur la montre est l'heure système du client

- Le clique sur le bouton de la montre permet de rendre les aiguilles self-illuminantes pendant 10 secondes
- Calcul automatique du prix et affichage du prix en fonction des choix d'options
- [optionnel] Saisie d'un texte pour créer une texture (gravure du nom du propriétaire sur le boîtier) (☆☆☆)

Partie vue :

- Intégration d'un design system
- Contenu du site :
 - Page de login (si non connecté, redirection sur la page de login)
 - Page liste avec plusieurs modèles prédéfinis + système de filtres (budget de xx€ à xx€, matière du bracelet, couleur) + pagination
 - Page détail avec un slug qui gère les modèles prédéfinis
 - Ajout au panier avec une vue qui affiche tous les modèles bien ajoutés précédemment
- Utilisation d'un GIT propre (avec des commits compréhensibles)

Partie Back:

Gestion des Utilisateurs et configuration d'authentification sur une page de login/sign up:

- Mettez en place un système d'authentification (login: username/password) pour permettre aux utilisateurs de créer des comptes, de se connecter et de gérer la configuration de leurs montres.

Gestion des Commandes et Panier d'achat :

- Implémentez un mécanisme permettant aux utilisateurs d'ajouter des montres configurées à leur panier.
- Un utilisateur peut ajouter plusieurs montres au panier. Le total des différentes montres doit s'afficher dans le panier.
- Un utilisateur peut supprimer une montre de son panier.
- Un utilisateur peut cliquer sur une montre pour revenir à la page de configuration de la montre en question et toute modification doit être reflétée dans le panier (style et prix) après enregistrement des modifications.
- **Processus de paiement : À ne pas développer!**

Pour accomplir ces tâches côté serveur, vous devez travailler sur la définition de votre base de données et la création du serveur avec les API nécessaires.

Configurez la base de données:

Configurez une base de données pour stocker les informations sur les utilisateurs, les paniers, les configurations de montres, et les informations sur les composants.

[A vous de définir les tables et les relations de votre base de données.](#)

Informations sur les différents composants :

- boîtier
- pierres: 1 rubis, 1 diamant, 1 émeraude
- bracelets (différents pour métal, cuir, tissus)
- le prix de chaque choix des composants

Information sur les utilisateurs:

- Info username/password pour vérifier la connexion
- Les différentes configurations de montres enregistrées par l'utilisateur
- Les différentes montres ajoutées au panier de l'utilisateur

Définissez les API nécessaires pour la logique.

Quelques exemples:

- Créer un nouveau compte avec un nom d'utilisateur et un mot de passe.
- Connecter les utilisateurs en vérifiant leur nom d'utilisateur et mot de passe fournis.
- Ajouter une nouvelle montre configurée dans la base de données, quand l'utilisateur décide de l'enregistrer.
- Retourner la liste des montres configurées par un utilisateur.
- Retourner la liste des montres ajoutées au panier.
- Ajouter une montre configurée au panier.
- Retourner une montre existante et re-configurable (changement de composant).
- Mettre à jour la configuration d'une montre déjà existante dans la base de données.
- Supprimer une montre de la base de données avec toutes les informations relatives.

N.B. Ne vous limitez pas aux APIs proposées, à vous de définir les APIs nécessaires pour que votre site fonctionne bien selon le design et la logique que vous proposez.