

Programmation Web

Projet

version du 5 avril
date limite: 15 mai

Le but de ce projet est de réaliser un site de friperie solidaire. Le site proposera à ses utilisateurs plusieurs sortes de produits bon marchés :

1. Des pantalons (jeans, chino, etc.) dont certains avec accessoire ou un choix d'accessoire (pantalon avec ceinture, pantalon avec une ceinture ou des bretelles, etc.) ;
2. Des chemises de différentes couleurs dont certaines avec un choix d'accessoires (cravate, nœud papillon, pochette de costume, etc.) ;
3. Des vestes de différents types (laine, cuir, jean, etc.) ;
4. Tout autre type de produit proposable par une friperie (pulls, shorts, chaussettes, etc.) ;
5. Des combinaisons à prix fixes de produits choisis dans une certaine catégorie par l'utilisateur. Par exemple :
 - (a) combinaison de base : un pantalon sans accessoire, une chemise, une veste ;
 - (b) combinaison business : un pantalon avec ceinture, une chemise avec cravate, une veste ;
 - (c) combinaison de luxe : un pantalon avec ceinture, une chemise avec cravate, une chemise avec nœud papillon ;
 - (d) etc.

Le site proposera deux types de connections : celles des clients, et celle du gérant.

1 Design et fonctionnalités du site

Le site doit être consultable aussi bien sur un téléphone portable ou sur une tablette que sur un écran large, avec le même confort d'usage : l'agencement des différents éléments d'une page doit donc s'ajuster en fonction de la taille de l'écran. Il doit être sur un nombre le plus réduit possible de pages – idéalement, une unique page – dont les contenus varieront dynamiquement. L'esthétique du site (couleurs, images, logos, décorations, animations, effets spéciaux) est totalement libre, mais elle doit être uniforme, et suffisamment riche visuellement.

1.1 Consultation du site

Lorsque le client consulte le site, il est initialement accueilli par une présentation de l'ensemble des vêtements. Le client doit pouvoir examiner l'ensemble des produits de chaque sorte, et passer librement d'une sorte à l'autre.

1.2 Ajout de produits au panier

Tout produit présenté au client sera accompagné de son nom, de son prix, d'une image de ce produit, et d'un bouton permettant au client d'ajouter ce produit à son panier, initialement vide :

1. La présentation d'un produit sera complétée par un menu déroulant permettant de choisir sa taille (XS, S, M, etc.). Pour un produit donné **toutes les tailles ne sont pas forcément disponibles** : le contenu de ce menu pourra varier dynamiquement en fonction des stocks disponibles.
2. La présentation d'un produit avec un choix multiple d'accessoires sera complétée par un menu déroulant permettant de choisir cet accessoire.
3. La présentation d'une combinaison prédéfinie sera complétée par un menu déroulant permettant également de choisir la taille de chaque composant de la combinaison.

Le montant total du panier du client sera visualisable en permanence, et mis à jour en continu au fil des ajouts.

1.3 Gestion du panier et finalisation de la commande

Le client doit pouvoir accéder au contenu courant de son panier, supprimer de ce panier les éléments de son choix, et revenir à l'une des présentations permettant d'ajouter des produits à son panier. Il peut aussi choisir de finaliser sa commande. Dans ce cas, le site lui demandera de remplir un formulaire précisant son nom, son prénom, son adresse, un numéro de portable et une adresse email, ainsi que l'heure de livraison souhaitée. Une fois ces données correctement entrées, la commande du client sera enregistrée, et le panier du client vidé.

1.4 Connection du gérant

Le gérant se connecte au site sur l'URI `/gerant`. On suppose que la base de données contenant les stocks est préalablement remplie. Le gérant peut à tout moment ajouter un ou plusieurs éléments au stock courant du site par exemple, ajouter dix éléments au stock de chemises rouges sans accessoires de taille XXL. Il peut aussi librement supprimer une commande enregistrée (car livrée).

1.5 Extensions possibles

La liste minimale de fonctionnalités décrites ci-dessus est imposée, mais n'est pas exhaustive - toute extension de cette liste est la bienvenue. Par exemple, la connection du gérant pourrait être authentifiée par mot de passe. Ou encore, quel que soit le visuel courant, un client pourrait à tout moment avoir la possibilité de s'authentifier par l'une ou l'autre des méthodes suivantes :

1. S'authentifier en tant que nouveau client. Le site demandera au client de remplir un formulaire contenant son nom, son prénom, une adresse, son ou ses codes d'entrée d'immeuble (optionnels), un numéro de portable, une adresse email et un mot de passe à confirmer. Le site enregistrera ces données dans sa base.

2. S'authentifier en tant qu'utilisateur déjà enregistré par la méthode (1). Le site lui demandera dans ce cas son adresse email et son mot de passe.

Un client authentifié n'aurait évidemment pas à remplir de formulaire lors de la finalisation de sa commande, les données nécessaires étant extraites de son profil. Il devrait aussi pouvoir : éditer son profil (ensemble des informations fournis dans le formulaire); se déconnecter.

2 Implémentation

Le serveur du site sera implémenté en faisant un usage libre mais approprié des *seuls* outils suivants : Bootstrap, jQuery, CSS, Node, les modules `express`, `ejs` et `pg` – PostgreSQL, pour la gestion d'une base de données de vêtements et de commandes de clients, et de toute autre information utile au fonctionnement du site¹ (aucune donnée ne doit apparaître en dur dans le code : ces informations seront toutes stockées dans la base de données).

Votre base de données devra être intégralement initialisée par un script `.sql` (et non par le code Javascript). Une fois initialisée (éventuellement avec plusieurs clients déjà enregistrés, *c.f.* les propositions d'extension ci-dessus), la base doit permettre une démonstration immédiate et convaincante.

Noter que la qualité d'écriture du code sera l'un des critères de l'évaluation : factorisation maximale (*aucun* copier-coller), généricité et efficacité, découpage en fonctions (ou encore mieux, en modules) mise en page (sur 80 colonnes, jamais plus), commentaires, etc.

3 Modalités

Composition des groupes. Le projet doit être réalisé en binôme, sauf cas d'impossibilité majeure (conditions d'études particulières, etc.) - nous vous invitons dans ce cas à en discuter le plus tôt possible avec votre enseignant. Un binôme peut être formé d'étudiants de deux groupes de cours/TD différentes. La répartition des tâches au sein d'un groupe doit être raisonnablement équilibrée. En cas de déséquilibre avéré, les notes finales pourront être individualisées.

Usage de Gitlab. Votre projet doit être réalisé à l'aide d'un dépôt sur le Gitlab de l'université². Ce dépôt doit être privé, créé par l'un des deux membres d'un binôme, l'autre membre du binôme étant invité comme développeur. Vous donnerez accès (au moins comme reporter) à ce dépôt à **chacun** des enseignants du cours (et bien sûr à aucun autre utilisateur). La régularité et l'équilibrage des commits seront pris en compte dans l'évaluation.

1. Ce projet n'est pas un projet de base de données. Une base de donnée basique mais fonctionnelle suffit.

2. https://gaufre.informatique.univ-paris-diderot.fr/users/sign_in. Utilisez le même identifiant et le même mot de passe que sur les machines de l'UFR.

Individualité des projets. De manière évidente, votre code doit être **strictement personnel**. Nous pouvons tolérer l'adaptation d'exemples repris sur la documentation des outils utilisés, mais en aucun cas le partage de code entre groupes – tous les rendus seront injectés dans un détecteur de plagiat insensible aux renommages ou aux changements de l'ordre des définitions³.

Rendus des projets. Le rendu sera une archive (*tar.gz* ou *zip*) sur moodle contenant l'intégralité des fichiers nécessaires au déploiement du site. En supposant `postgresql` installé et lancé, ce déploiement ne devra nécessiter que :

- la décompression de l'archive,
- l'exécution d'un script d'initialisation de la base de données nommé `init.sql`
- la commande `nodejs main.js`.

3. La soutenance de projet est un examen comme un autre. Le plagiat en projet constitue une fraude aux examens, passible au pire de lourdes sanctions disciplinaires – ce cas s'est produit plus d'une fois dans cette UFR.