

Projet de fin d'année

développeurs 1^{re} Année

SUJET

Vous réaliserez une boutique en ligne vendant les objets ou les services de votre choix. Ce site doit au moins comporter :

1. une page d'accueil présentant la boutique, ses catégories de produits et ses produits phare,
2. une page pour chaque catégorie,
3. une page détaillant chaque produit (avec des photos de ces produits)
4. un tunnel d'achat,
5. une page de compte utilisateur réservée au client,
6. un tableau de bord réservé aux administrateurs du site,
7. une page de contact, de mentions légales, de conditions générales de vente,
8. un menu de navigation efficace et fonctionnel.

Il doit y avoir au moins trois catégories de produits et chaque catégorie doit avoir au moins dix produits.

Vous n'implémenterez pas de véritable moyen de paiement, mais vous le simulerez.

Vous coderez ce CMS en PHP et il partagera sa base de données MySQL avec une IHM intégrée au navigateur, que vous coderez en Python, qui enverra des newsletters par e-mail.

Le rendu des sources du projet est exigible la veille de l'examen.

Repository GitHub OBLIGATOIRE pour les codes source.

VALIDATION DU THÈME DU SITE

Le thème de votre site doit d'abord être validé par Sonia.

Une fois que le thème de votre boutique aura été validé par l'équipe pédagogique, **vous devrez réaliser le dossier d'UX/UI avant de commencer à coder le front.**

Cahier des charges technique

UX/UI

1. Vous définirez le **contexte** de votre entreprise (ce que vend votre entreprise, ses valeurs, son positionnement sur le marché...),
2. Vous ferez un **benchmark** en faisant une analyse UI/UX (lois ergonomiques, gestalt...) de 4 sites concurrents (2 concurrents directs et 2 concurrents indirects, vous résumerez les points forts et les points faibles.
3. Benchmarking et fonctionnalité du projet

4. Vous définirez vos cibles (cœur de cible, cible principale, cible secondaire)
5. Vous proposerez **2 personas** avec **1 parcours utilisateurs**.
6. Vous réaliserez l'**arborescence** du site
7. Après avoir choisi vos **sources d'inspiration**, vous ferez ensuite le **wireframe d'un parcours utilisateurs** avec le logiciel de votre choix (Figma/XD) comprenant les pages demandées.
8. Vous réaliserez le **guide de style**
9. Vous réaliserez le **design** de ce même parcours

PHP

Technologies utilisées :

- PHP (MVC)
- MySQL

Le code créé doit répondre aux exigences de qualité qui ont été enseignées tout au long de l'année (indentation, nommage des variables / tables / fonctions, structure du code, sémantique, syntaxe, sécurité...).

Une attention particulière sera portée sur la qualité du code.

Les utilisateurs du site sont de 3 types :

- Guest (non connecté, aucun statut particulier)
- Un utilisateur inscrit (connecté, non-administrateurs)
- Admin (administrateurs du site)

Un utilisateur "guest" (simple visiteur non connecté) pourra naviguer sur le site, mais ne pourra pas passer de commande sans être connecté à un compte. Accès au back-office d'administration interdit.

Un utilisateur "customer" est un utilisateur connecté à un compte, qui peut passer des commandes et modifier ses infos (nom, prénom, email, etc...). Accès au back-office d'administration interdit)

Un utilisateur admin peut tout faire, y compris accéder au back-office d'administration.

BACK-OFFICE D'ADMINISTRATION :

L'utilisation de Bootstrap ou d'autres bibliothèques équivalentes est autorisée pour l'intégration du back-office d'administration.

Le site devra comporter un back-office d'administration (accessible uniquement par les

administrateurs du site) permettant de régir (ajout / modification / suppression) les éléments suivants :

1. Admin des utilisateurs :

Les utilisateurs du site

- page "liste des utilisateurs" : liste l'ensemble des utilisateurs du site
 - id
 - nom
 - prénom
 - email
 - is_admin (oui/non)
 - + boutons supprimer / modifier pour chaque utilisateur
- page de formulaire pour ajouter / modifier un utilisateur:
 - Champ "nom" (**champ obligatoire**) (type text)
 - Champ "prénom" (**champ obligatoire**) (type text)
 - Champ "email" (**champ obligatoire**) (type email)
 - Champ "password" (type password)
Notes pour le champ password :
 - **obligatoire à la création d'un nouvel utilisateur**
 - **modification d'un utilisateur : ne pas pré-remplir le champ. L'admin peut remplir ce champ pour re-définir le mot de passe de l'utilisateur qu'il est en train de modifier. S'il laisse ce champ vide, le mot de passe actuel de l'utilisateur ne sera pas modifié.**
 - Champ "is_admin" (**champ obligatoire**) (type select oui/non)
- possibilité de supprimer les utilisateurs. **Attention : la suppression d'un utilisateur ne doit pas entraîner la suppression des commandes qui lui sont liées afin de garder une trace comptable des commandes.**

Seuls les admins du site ont la possibilité de créer un nouvel Admin du site via le back-office. Les utilisateurs qui s'enregistrent via le formulaire d'inscription du "front" ne doivent pas être des admins.

2. Admin des catégories :

Les catégories des articles du site.

- page "liste des catégories" : liste l'ensemble des catégories du site
 - id

- nom de la catégorie
- + boutons supprimer / modifier
- page de formulaire pour ajouter / modifier une catégorie :
 - Champ "nom" (**champ obligatoire**) (type text)
 - Champ "description" (textarea)
 - Image (**champ obligatoire**) (type file) (**jpg, jpeg, png et gif, 2Mo maximum**)
- possibilité de suppression des catégories

La suppression d'une catégorie entraîne une suppression de la liaison entre cette catégorie et les produits qui lui sont liés, mais les produits de cette catégorie ne doivent pas être supprimés.

3. Admin des produits et de leurs images :

Les produits du site. Un produit peut appartenir à plusieurs catégories. Un produit peut avoir plusieurs images (dont l'image principale du produit).

Un produit non activé ne doit pas être visible dans ses catégories par les utilisateurs du site, mais doit rester visible et administrable dans le back-office d'administration.

De même, une image produit non activée ne doit pas être visible par les utilisateurs du site, mais doit rester visible et administrable dans le back-office d'administration

- page "liste des produits" : liste l'ensemble des produits du site
 - id
 - nom du produit
 - catégories
 - publié ou non ?
 - + boutons supprimer / modifier
- page de formulaire pour ajouter / modifier un article :
 - Champ "nom" (**champ obligatoire**) (type text)
 - Champ "catégories" (select list à choix multiples)
 - Champ "description" (**champ obligatoire**) (textarea)
 - Champ prix (**champ obligatoire**) (type text, ne doit contenir que des INT ou des FLOATS)
 - Champ quantité (**champ obligatoire**) (type number, ne doit contenir que des INT positifs)
 - Champ "activé" (type checkbox, coché par défaut)
- possibilité de suppression des produits
- page liste des images du produit ainsi qu'un formulaire d'ajout d'image liée au produit

- Champ "image" (**champ obligatoire**) (type file) (**jpg, jpeg, png et gif, 2Mo maximum**)
- Champ "image principale" (type checkbox, non-coché par défaut)
- Champ "activé" (select list oui/non)
- possibilité de suppression des images du produit

La suppression d'un produit entraîne la suppression des images liées à ce produit.

La suppression d'une image produit entraîne la suppression du fichier image lié sur le serveur afin de ne pas conserver de fichier fantômes.

Enregistrer en DB l'attribut alt des images stockées sur le serveur.

4. Admin des commandes

Une commande doit comporter les informations de ses produits à l'instant T de sa création.

Si par la suite les prix des produits sont changés par un admin, les prix de la commande eux NE DOIVENT PAS changer, et le total doit rester le même que la commande d'origine.

- Page liste des commandes passées par les utilisateurs
 - id
 - Nom / prénom du client
 - total de la commande
- Page détails d'une commande :
 - informations du customer qui a passé la commande (nom, prénom, adresse, email)
 - tableau récapitulatif de la commande (par ligne)
 - nom du produit
 - quantité
 - prix unitaire
 - total de la ligne
 - Total de la commande (total de toutes les lignes)

Aucune gestion de TVA ou de frais de port n'est demandée.

Pas de possibilité de supprimer ou modifier une commande.

FRONT-OFFICE :

Un utilisateur pourra :

1. S'inscrire (si pas connecté uniquement)

Tous les champs sont obligatoires :

- Nom
- prénom
- adresse
- email
- mot de passe

L'utilisateur pourra sélectionner les catégories auxquelles il veut s'inscrire pour recevoir les newsletters correspondantes (inscription non obligatoire, il peut ne sélectionner aucune catégorie s'il ne veut aucune newsletter).

(Pour faire la liaison avec l'interface de gestion de newsletters de la partie Python).

2. Se connecter (si pas connecté uniquement)

Tous les champs sont obligatoires :

- email
- mot de passe

3. Passer des commandes (si connecté uniquement)

Gestion du panier via SESSION PHP.

La page de panier doit comporter la liste des produits associés à leur quantité et leur prix unitaire, puis le total du panier.

4. Consulter son historique de commandes et en voir les détails via une page "détails de commande". (si connecté uniquement)

- Informations du customer qui a passé la commande
- tableau récapitulatif de la commande
 - nom du produit
 - quantité
 - prix unitaire
 - total de la ligne
- Total de la commande (total de toutes les lignes)

Une commande doit comporter les informations de ses produits à l'instant T de sa création.

Si par la suite les prix des produits sont changés par un admin, les prix de la commande eux NE DOIVENT PAS changer, et le total doit rester le même que la commande d'origine.

L'utilisateur pourra sélectionner ou désélectionner son inscription à la newsletter pour faire la liaison avec l'interface de gestion de la newsletter de la partie Python.

6. Se déconnecter (si connecté uniquement)

PYTHON

Technologies utilisées :

- Python avec utilisation
 - du Framework Flask pour la partie web
 - du module SQLAlchemy pour l'accès à la bdd
- Vos créations de template HTML devront être cohérent avec le code PHP afin que votre CSS continue de fonctionner

De même que pour les autres parties, votre code doit répondre aux exigences de qualité qui ont été enseignées tout au long de l'année (commentaires, nommage des variables et des fonctions, découpage du code, séparation de la logique...).

Une attention particulière sera portée sur la qualité du code.

Seul l'administrateur pourra accéder à cette fonctionnalité de votre site.

Vous devrez choisir et réserver l'URL d'accès afin que le serveur python et php puissent coexister sans conflit.

Le but de cette partie est de permettre à l'administrateur d'envoyer une newsletter.

ADMINISTRATION D'ENVOI DE LA NEWSLETTER :

L'utilisation de Bootstrap ou d'autres bibliothèques équivalentes est autorisée pour l'intégration du back-office d'administration.

Cette partie administrative ne sera accessible qu'aux administrateurs. L'utilisation de la même BDD que pour l'administration du site permettra de se baser sur les login / mdp déjà enregistrés.

- **Notes pour le champ password :**
 - **Ne peut pas être vide**

1. Création de la newsletter :

a. Rédaction d'une newsletter

Vous devrez permettre à l'administrateur de rédiger un texte qui pourra être formaté classiquement (titre, gras, italique, paragraphe...). La newsletter sera envoyée à tous les utilisateurs qui auront activé leur inscription à cette newsletter dans l'espace de gestion de leur compte de la boutique. Cette inscription sera stockée dans la base de données.

Ce texte pourra être injecté dans un template HTML, préparé en amont, afin d'habiller de votre identité graphique votre newsletter.

Sur la page de rédaction voici la liste des champs qui pourront être trouvés :

- Champ "Corps_Newsletter" (champ obligatoire) (type TextArea)

- Il doit proposer des options de formatage de texte classique (bold, italic ...)

Résumé des fonctionnalités :

- Rédiger une newsletter

2. Envoie des mails :

Une fois rédigée, la newsletter devra être envoyée à tous les destinataires, dont les adresses mail auront été récupérées dans la base de données.

Sur la page de rédaction à un endroit ergonomiquement intéressant que vous choisirez (à côté du textArea ou tout en bas de la page ou en bouton flottant...) il faut un bouton « Envoyer » qui, lorsqu'il sera cliqué, demandera confirmation à l'utilisateur avant envoi aux destinataires.

3. Historique des mails envoyés

Les newsletters envoyées devront être stockées dans la base de données et être accessibles pour relecture. Lorsque l'utilisateur choisira l'option historique, il pourra voir sous forme de liste ou de block toutes les newsletters déjà envoyées. Il pourra choisir de les trier par ordre chronologique ou chronologique inversé, et lorsqu'il sélectionnera l'une de ces newsletters, celle-ci s'affichera en popup sur la page.

Si l'utilisateur quitte le menu Création De Newsletter pour aller sur l'option Historique, tout le texte tapé dans le texte area, et les flux sélectionnés doivent être gardés pour que quand il retourne sur l'option de création, l'utilisateur n'ait pas perdu son travail.

4. Se déconnecter

Enfin lorsque l'utilisateur aura terminé ce qu'il avait à faire avec les newsletters, il devra pouvoir quitter l'application avec un bouton qui après lui avoir demandé confirmation, le déconnectera et terminera sa session.

Si l'utilisateur avait commencé à taper du texte et sélectionner des liens, son travail devra être sauvegardé à la déconnection et rechargé à sa prochaine connexion.

Les 3 options (Création, Historique et Déconnection) seront accessibles à tout moment à travers un menu toujours visible à gauche de l'espace de travail

Option Bonus : Mettre ces options dans un menu drawer escamotable

FRONT

Vous intégrerez les interfaces nécessaires aux applications développées en PHP et en Python. Le *frontoffice* du CMS site se doit

1. d'être *responsive*,
2. de répondre aux contraintes du SEO (du moins pour les pages aux contenus importants) :
 - a. les balises de chaque page du *frontoffice* meta:description et title sont correctement renseignées dans le head,
 - b. chaque page du *frontoffice* contient au moins un h1, un h2 et un h3,
 - c. chaque page contient un header, un main, un footer (ainsi que d'autres éléments HTML structurants à forte valeur sémantique quand cela est pertinent, par exemple : section, article, aside, etc.).

Votre intégration s'appuiera au maximum sur les maquettes préalablement conçues dans les cours d'UX/UI.

Le code créé doit répondre aux exigences de qualité qui ont été enseignées tout au long de l'année (indentation, nommage des variables / fonctions, structure du code, sémantique HTML5 et balises meta nécessaires à un bon référencement, syntaxe...). Une attention particulière sera portée sur la qualité du code.

BONUS (liste non exhaustive)

Ces features ne sont pas obligatoires

- Feature recherche de produits
 - un formulaire de recherche simple (champ texte pour les mots clés recherchés)
 - une page dédiée aux résultats de recherche qui liste les produits correspondants aux mots-clés précédemment entrés par l'utilisateur
- Feature notations des produits (reviews)
 - Un utilisateur **connecté** pourra laisser des avis sur les produits du shop via un simple formulaire en dessous de la fiche produit concerné (note du produit de 1 à 5 et un commentaire - les deux sont obligatoires pour laisser un avis)
 - un utilisateur ne peut laisser qu'un seul avis par produit
 - modération : seuls les administrateurs (connectés) du site peuvent supprimer un avis client
 - les avis postés par les utilisateurs seront listés en dessous de la fiche produit avec pour informations présentes :
 - date de l'avis
 - nom de l'utilisateur

- note /5
- commentaire

Bon dev à tous !