Projet final PHP Site E-Commerce

> Fini les OH SNAP et les BIM, on va s'attaquer au gros du gros, ce projet rentre dans le cadre de votre note finale du module PHP, et vous allez faire votre premier site web 100% en PHP.

👉 Le projet se fait en groupe de 2 ou 3 personnes **MAXIMUM**

Énoncé
Schéma de la base de données
Fonctionnalités supplémentaires
Guide pour vous lancer sur le projet
- XAMPP
- Votre environnement de travail
- PhpMyAdmin

Rendu

Énoncé

Vous allez devoir réaliser un site web sous forme d'un site de E-Commerce, il devra comporter plusieurs pages :

- Une page Connexion `/login` et Inscription `/register`

  → /register sera le formulaire de création de compte.

  → /login sera le formulaire de connexion. Une fois connecté, l'utilisateur est redirigé sur la page Home.

  → Un utilisateur doit avoir une adresse mail et un username unique 😉

  → Lorsque qu'un utilisateur crée son compte, il faudra faire en sorte qu'il soit automatiquement connecté.

- Une page Home `/`

  → Elle recensera tous les articles en vente avec les plus récents en premier.

- Une page Vente `/sell`

  → Elle permettra de créer un article pour mettre en vente.

  → Voir la table Article pour voir les champs à mettre sur le formulaire de création

  → Voir également la table Stock qui vous sera utile

- Une page Détail `/detail`

  → Affichera les détails de l'article passé en paramètre de votre requête GET.

  → L'utilisateur pourra ajouter l'article à son panier via cette page.

- Une page Panier `/cart`

  → Affichera tous les articles présents dans le panier de l'utilisateur actuellement connecté.

  → Possibilité de passer commande si le solde de l'utilisateur est suffisant.

  → Possibilité d'augmenter le nombre d'articles présent dans le panier (par exemple, je possède 1 gourde dans mon panier, je peux décider d'en ajouter pour en acheter 5).

  → Possibilité de supprimer des éléments du panier.

- Une page Confirmation `/cart/validate`

  → Validera le panier si le solde de l'utilisateur est suffisant.

  → Permet de renseigner les informations de facturation de l'utilisateur.

  → Permet de valider la commande, vider le panier, et générer une facture.

- Une page Modifier `/edit`

  → Accessible en méthode POST depuis la page d'un article.

  → Possibilité de modifier ou de supprimer l'article.

  → ⚠️ **SEUL** l'utilisateur qui a créé l'article est capable de le modifier *(à l'exception d'un administrateur).*

- Une page Compte `/account` pour afficher les informations du compte ainsi que tous les articles postés par ce compte, l'utilisateur doit voir les articles qu'il a acheté, et doit être capable de modifier ses propres informations comme son adresse mail ou son mot de passe

  → Accessible avec des paramètres de requête GET pour voir le compte d'une personne en particulier. Si aucun paramètre n'est mis, alors cette page sera pour visualiser votre compte

  → Si la page vise un utilisateur qui n'est pas le vôtre :

→ Affichera tous les articles postés par ce compte

→ Affichera les informations du compte

→ Si la page vise votre utilisateur :

→ Affichera tous les articles que vous avez publié

→ Affichera les factures.

→ Possibilité de modifier vos informations personnelles

→ Possibilité d'ajouter de l'argent à votre solde

- Une *partie* Tableau Administrateur `/admin`, accessible par les membres ayant le rôle administrateur où il peut modifier ou supprimer n'importe quel Article et/ou Utilisateur

  → Accessible uniquement pas les utilisateurs ayant le rôle administrateur

  → Répertorie tous les post, avec possibilité de les modifier / supprimer

  → Répertorie tous les utilisateurs, avec possibilité de les modifier / supprimer

  > 💡 J'ai bien fait en sorte d'écrire "une partie" et non une page. Libre à vous de faire plusieurs pages depuis le `/admin` afin d'avoir une belle arborescence 😃

> Un utilisateur non connecté peut accéder aux pages `Home` et `Detail`. Toutes les autres pages doivent le rediriger vers la page connexion.

> 🚨 Prenez conscience que principalement la partie **FONCTIONNELLE** de votre site web sera notée, vous n'avez pas besoin de vous entêter à faire un site qui claque, appuyez vous de la capsule de cours sur "comment faire un joli site web" que vous avez-eu l'année dernière

> ⛔ ATTENTION: l'utilisation de Framework et/ou toute technologie autre que PHP pour la réalisation du backend est formellement **INTERDITE**. C'est un projet PHP range ton Laravel frérot

Schéma de la base de données

Votre base de données devra comporter **AU MINIMUM** ces informations:

Une table User

→ Un ID qui s'auto incrémente

Une table Article

→ Un ID qui s'auto incrémente

→ Un username

→ Un mot de passe **chiffré en bcrypt**

→ Une adresse mail

→ Un solde

→ Une photo de profil

→ Un rôle

→ Un nom

→ Une description

→ Un prix

→ Une date de publication

→ L'ID de l'auteur

→ Lien de l'image

Une table Cart

→ Un ID qui s'auto incrémente

→ L'ID du User

→ L'ID de l'article

Une table Stock

→ Un ID qui s'auto incrémente

→ L'ID d'un article

→ Le nombre d'article en stock

Une table Invoice

→ Un ID qui s'auto incrémente

→ L'ID du User

→ Une date de transaction

→ Un montant

→ Une adresse de facturation

→ Une ville de facturation

→ Un code postal de facturation

Fonctionnalités supplémentaires

Voici quelques fonctionnalités bonus que vous pouvez rajouter à votre projet pour gratter quelques points 🙂

→ L'utilisateur peut garder des articles en favoris

→ Ajouter une barre de recherche pour trouver des articles en particulier

→ Ajouter la possibilité de trier les articles différemment dans Home

→ Ajouter la gestion de stock des articles (Si un article à un stock de 2, il n'est pas possible d'en commander 3. Ce bonus modifie la page Détail car il est maintenant possible d'ajouter plusieurs fois l'article à son panier en une seule fois)

→ Ajouter un système de catégories

→ Possibilité, si l'utilisateur a déjà acheté l'article, de le noter et de mettre un commentaire.

→ Système d'envoi de mail (mot de passe oublié, article bien commandé, etc…)

Guide pour vous lancer sur le projet

Vous pensiez réellement qu'on allait vous laisser galérer alors que vous venez de débuter en PHP ? (en vrai on aurait été capable mais on est sympa 😉)

XAMPP

Déjà vous allez installer un logiciel permettant de créer un serveur WEB en local, où vous pourrez dev en PHP dessus.
Le nom du logiciel dépend de votre OS mais en vrai je vous jure ce n'est pas compliqué !

→ **XAMPP** pour Windows (Version PHP 8)

→ **MAMP** pour MAC

→ **LAMP** pour Linux

> Ubuntu/Debian : `sudo apt-get install lamp-server^`
> Arch/Manjaro : `sudo pacman -S httpd php php-apache mysql phpmyadmin`
> (Toutes distrib) à la fin, faites :
> `sudo systemctl enable —now httpd`
> `sudo systemctl enable —now mysql`

J'appellerai ce logiciel XAMPP pour le reste du tuto, comme ça on ne se perd pas.

> ⚙️ Pensez bien à configurer votre XAMPP sur PHP 8 si jamais. MAMP par exemple utilise PHP 7.4 par défaut

Une fois votre XAMPP d'installé, lancez votre serveur Apache et SQL (sur MAMP il suffit d'appuyer sur start ça lance tout en même temps).

Votre environnement de travail

Bon, votre XAMPP a bien travaillé, il a même créé un dossier de travail normalement.

- Sur XAMPP, il suffit de cliquer sur le bouton `📂 Explorer`
- Sur MAMP, le dossier est localisé dans `/Applications/MAMP`
- Sur LAMP, le dossier est situé dans `/var/www`

Rendez-vous dans le dossier `htdocs`, c'est ici que XAMPP va aller chercher vos site web lorsque vous lui demanderez.

Dans ce dossier, créez un sous-dossier `php_exam` et mettez y un fichier que vous allez appeler `index.php`, écrivez le texte ci-dessous juste pour la démo.

```
<?php
  $hello = "World";
```

```
?>
<h1>Hello <?php echo $hello ?> !</h1>
```

> ⬆️ Je pense que vous avez compris comment on pouvait mélanger du PHP avec du HTML, je vous laisse vous renseigner sur Internet on peut faire des trucs encore plus stylé que ça

Pour découvrir votre superbe production, rien de plus simple : ouvrez un navigateur et rendez-vous sur `localhost/php_exam`

→ *Sur MAMP, le port utilisé par défaut est 8888 et non 80, vous devez donc vous rendre à l'adresse* `localhost:8888/php_exam`

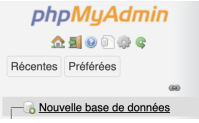
Il est beau votre site web vous trouvez pas ? (non)

> 👉 **Tout le reste de votre projet devra se faire dans ce dossier**

PhpMyAdmin

HOP HOP HOP vous allez où comme ça ? Il nous reste un dernier truc à voir avant de vous laisser développer 😁

→ Rendez-vous sur `localhost/phpmyadmin`, c'est ici qu'est stocké toutes vos bases de données, vous allez d'ailleurs en créer une nouvelle, juste ici :



Appelez la `php_exam_db`, à vous maintenant de créer les 5 tables qu'on vous demande

> 💡 Si on vous demande de vous connecter, l'user c'est *root* et le mot de passe *root*, ici on est sécure tavu 🤓

→ Pour vous connecter à une base de données en PHP, voici un bout de code pour vous éviter de galérer (On est VRAIMENT trop sympa 🤨)

```
<?php
$mysqli = new mysqli("localhost", "root", "root", "php_exam"); // Connexion à la db "php_exam"
// si vous avez une erreur ici, remplacez le deuxième "root" par une string vide

$result = $mysqli->query("SELECT * FROM users"); // On utilise l'instance créée pour faire une requête bidon
?>
```

→ À vous maintenant d'organiser correctement votre code pour éviter de répéter cette ligne de code dans chaque fichier PHP 😇

Rendu

On n'a pas parlé du rendu, et ça serait dommage parce que c'est un peu ça qu'on va noter 💁🏾‍♂️

Le rendu est simple, on vous demande de créer un **repo git** qui sera votre dossier `php_exam` que vous avez mis dans le fameux dossier `htdocs`

> N'oubliez pas de nous fournir un beau README clair et explicatif de votre projet, avec les procédures d'installations et les membres du groupe 🙂

⚠️ Attention si vous utilisez **Gitea** : les fichiers de plus de 1 Mo ne sont pas acceptés ! Soyez vigilants lors d'ajouts de photos de profils ou autres fichiers volumineux 🙂

🚨 **ATTENTION**: Au sein de votre repo, ajoutez-y le fichier SQL qui permet de générer votre base de données. Dans la page phpMyAdmin, rendez-vous ici `php_exam_db ▸ Exporter ▸ Executer`, cela doit vous générer le fichier `php_exam_db.sql` et c'est ça qui nous intéresse

Un malus de **5 points** sera attribué à votre note final si ce fichier n'est pas présent dans votre répo git !

> Les soutenances auront lieux le **MERCREDI 4 JUIN**

> Le projet est à rendre avant le **DIMANCHE 8 JUIN** à **23h59**, seul le dernier commit avant cette date sera considéré 🙂

📎 Le lien de votre repo devra nous être transmis par Mail ou Teams.

Votre repo doit etre public ou vous devez rajouter lomathieu sur gitea ou louvelmathieu