

Énoncé du Travail Pratique 2

18 août 2023

Préparé par
Benjamin Lemelin et Karine Filiatreault

1 Résumé

Créer un site web dynamique en PHP pour une entreprise spécialisée dans la vente de vêtements. Le site doit montrer un catalogue de produits, le détail de chaque produit et permettre des achats en ligne. Pour simplifier, il n'y aura pas de gestion de compte utilisateur ni de transactions par carte de crédit.

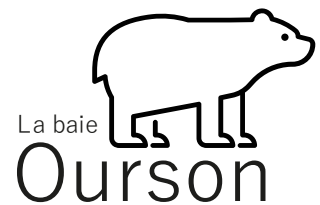
2 Conditions de réalisation

Valeur de la note finale	Type	Durée	Nombre de remises
25 %	En équipe	3 semaines	2

3 Mise en contexte

La compagnie de *La Baie d'Ourson* se spécialise dans la vente de vêtements pour enfants de 0 à 16 ans. Leur clientèle est surtout composée de parents ayant à cœur l'achat local, les vêtements étant fabriqués au Québec.

Durant ses premières années, *La Baie d'Ourson* ne prenait que des commandes par catalogue, façon *Sears*. Depuis, l'entreprise a énormément grossi et désire offrir un site de ventes en lignes. Ce site doit permettre à un client consulter des produits, de créer un compte, un panier et d'effectuer des commandes.



4 Spécifications

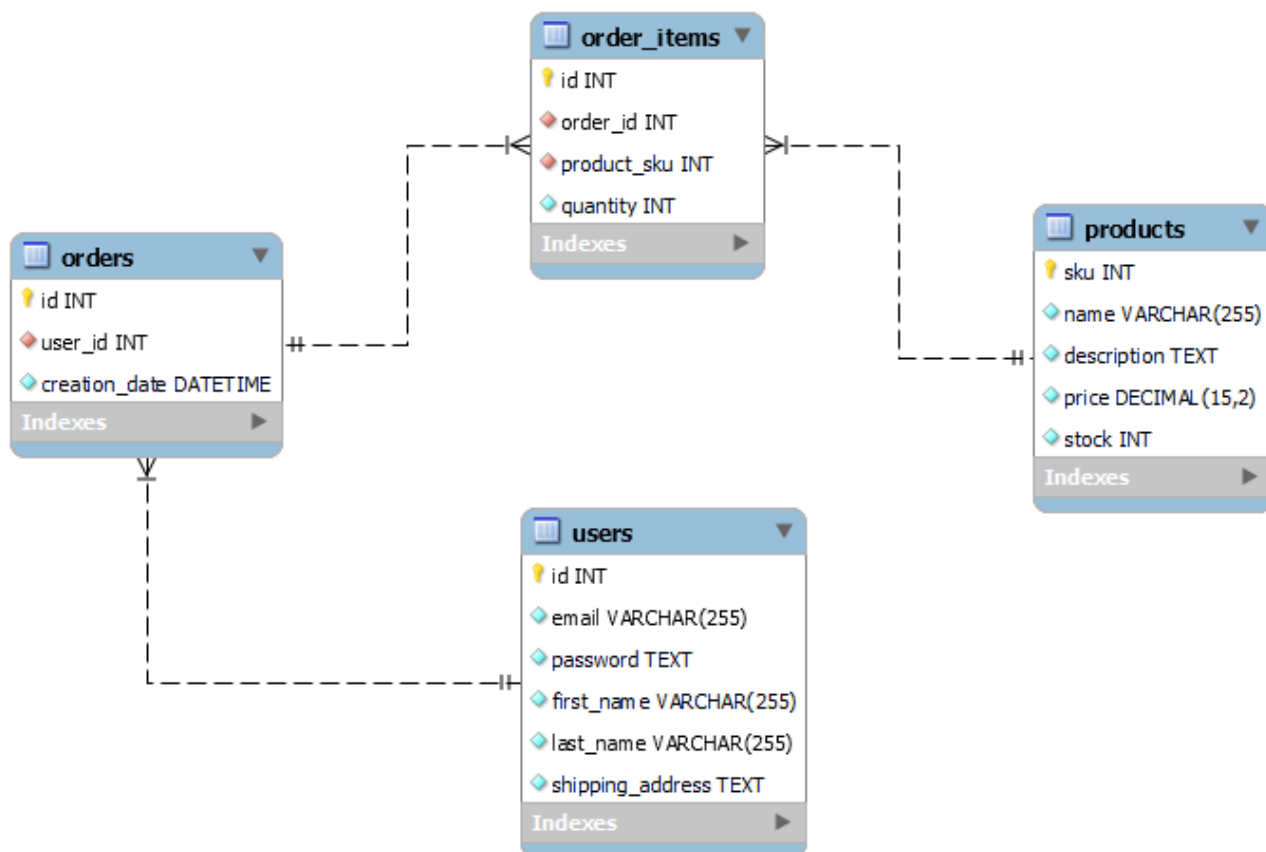
Cette section vise à vous guider durant la réalisation de ce travail. Réalisez chaque étape dans l'ordre et passez à l'étape suivante uniquement lorsque vous avez terminé l'étape actuelle.

4.1 Base de données

Le script de la base de données est fourni avec le TP et ne doit pas être modifié sans consulter le professeur. Nommez votre base de données **TP2_Nom1_Nom2** avec l'interclassement « utf8_general_ci » puis importez le script.

Pour vous connecter à la base de données dans votre code, vous devez utiliser l'utilisateur **webphp** et le mot de passe **qwerty123**.

Voici le schéma de base de données pour le site.



Notez que ce même fichier insère les produits suivants dans la base de données. Vous pouvez les changer et vendre ce que vous voulez 😊 Je vous suggère même d'en ajouter quelques-uns pour que ce soit plus réaliste.

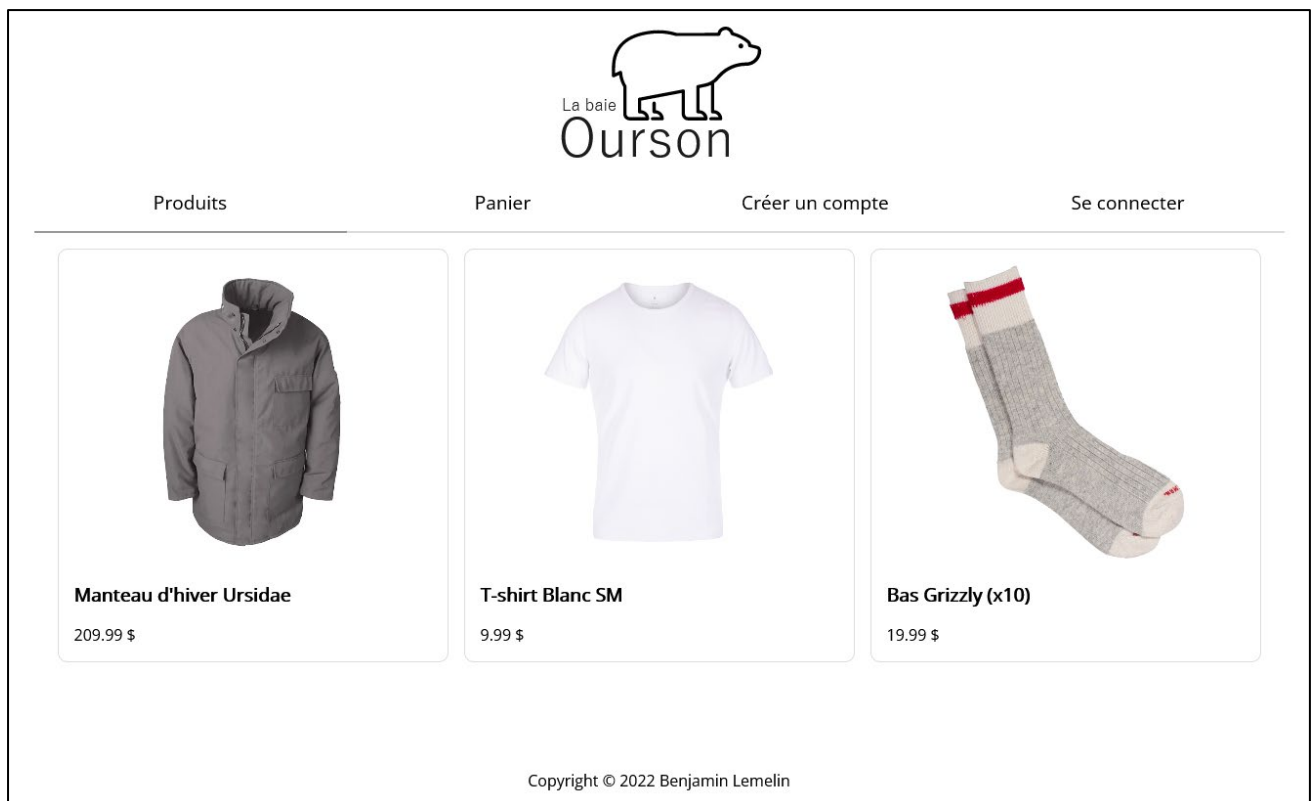
Skus	Nom	Description	Prix
612458	T-shirt Blanc SM	100% polyester. Facile à laver. Format Small.	9.99
154789	Manteau d'hiver Ursidae	Manteau pour l'hiver très chaud avec doublure.	209.99
887414	Bas Grizzly (x10)	Dix bas de laine gris. Très résistants.	19.99

4.2 Page d'accueil

Créez la page d'accueil respectant la maquette ci-dessous. Les produits sur cette page doivent être obtenus à partir de la base de données. Cliquer sur un produit mène à [une page ayant plus de détails](#) sur ce dernier.

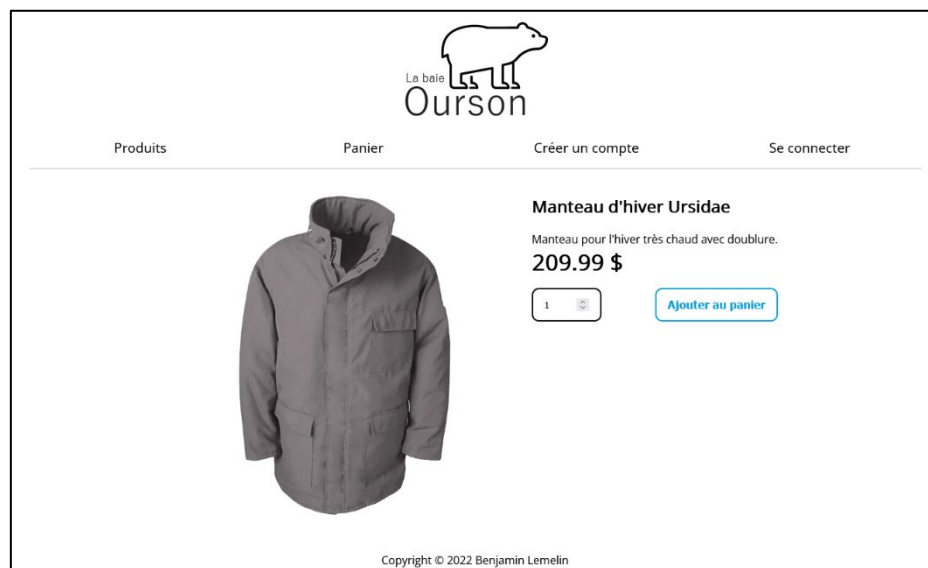
i Information

Vous n'avez pas à respecter parfaitement la maquette. Seule la structure est importante. Par exemple, si vous désirez changer les couleurs ou utiliser *Bootstrap*, gêtez-vous!



4.3 Page de consultation de produit

Créez la page de consultation de produit (**product.php**) respectant la maquette ci-dessous. Les informations sur le produit doivent être obtenues à partir de la base de données. Le bouton « *Ajouter au panier* » permet d'ajouter le produit au panier, avec la quantité indiquée.



Astuce

Le **sku** du produit à consulter devrait être à même l'url de cette page. C'est similaire à l'envoi d'un formulaire avec la méthode **GET**, mais sans utiliser de formulaire.


4.4 Barre de navigation

La barre de navigation en haut de chaque page change si l'utilisateur est connecté à un compte. La [création d'un compte](#) et la [connexion à ce dernier](#) se font sur d'autres pages. Seul le email du compte actuel doit être conservé dans la session car il est suffisant pour pouvoir identifier un utilisateur de façon unique.

Produits	Panier	Créer un compte	Se connecter
----------	--------	-----------------	--------------

Produits	Panier	Se déconnecter
----------	--------	----------------

4.5 Page de création de compte



Produits

Panier

Créer un compte

Se connecter

Créer un compte

Information sur le compte

Email :

Mot de passe :

Confirmation du mot de passe :

Information sur la livraison

Prénom :

Nom :

Adresse de livraison :

[Créer le compte](#)

Copyright © 2022 Benjamin Lemelin

Cette page ([signup.php](#)) doit permettre à l'utilisateur de se créer un compte. N'oubliez pas de faire des validations et d'afficher un message d'erreur. Le compte doit être conservé dans la base de données. Le mot de passe doit être minimalement chiffré avec [password hash](#). Lorsque la création du compte est réussie, redirigez l'utilisateur vers [la page de connexion](#).

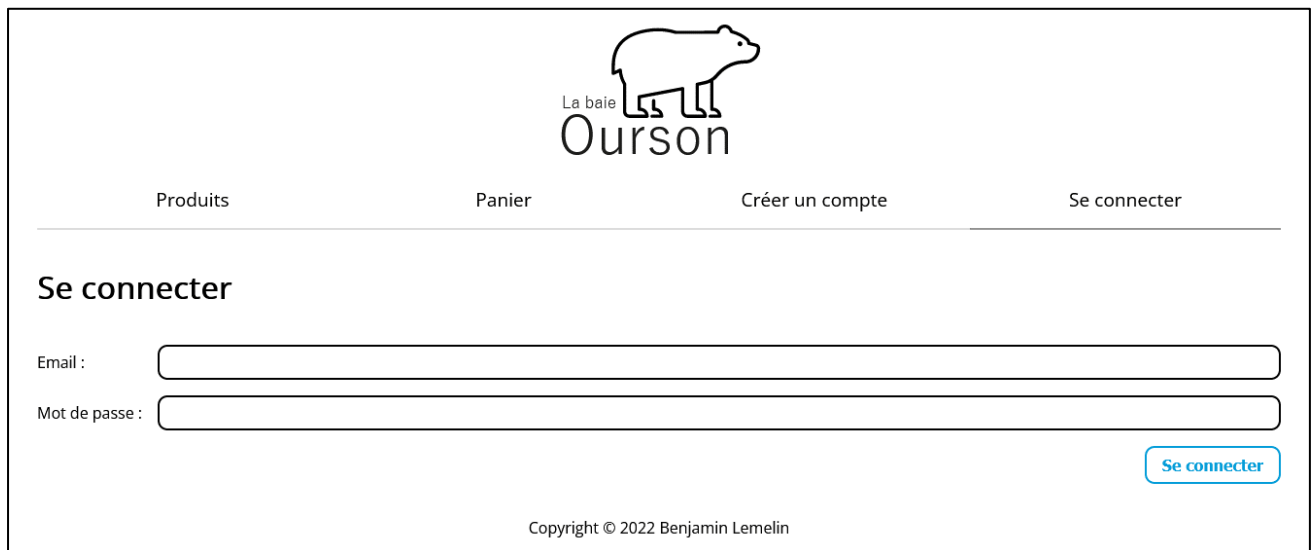
Champ	Validations
Email	<ul style="list-style-type: none">Obligatoire et non vide (excluant les espaces).Le format doit être valide (voir filter var).
Mot de passe	<ul style="list-style-type: none">Obligatoire.

Confirmation du mot de passe	<ul style="list-style-type: none"> Obligatoire. Doit être identique au mot de passe.
Prénom	<ul style="list-style-type: none"> Obligatoire et non vide (excluant les espaces).
Nom	<ul style="list-style-type: none"> Obligatoire et non vide (excluant les espaces).
Adresse de livraison	<ul style="list-style-type: none"> Obligatoire et non vide (excluant les espaces).

Information

Dans un contexte réel, les validations seraient beaucoup plus poussées. Par exemple, l'adresse de livraison serait validée par un fournisseur de services de livraison, tel que [Poste Canada](#).

4.6 Page de connexion à un compte



La page de connexion est présentée dans une maquette. En haut, le logo 'La baie Ourson' (un ours) est centré. En dessous, une barre de navigation contient les liens 'Produits', 'Panier', 'Créer un compte' et 'Se connecter'. Le titre 'Se connecter' est affiché à gauche. Deux champs de saisie sont prévus : 'Email :' et 'Mot de passe :'. Un bouton 'Se connecter' est situé à droite des champs. Le copyright 'Copyright © 2022 Benjamin Lemelin' est en bas.

La page de connexion (**signin.php**) doit permettre à l'utilisateur de se connecter à son compte. N'oubliez pas de faire des validations et d'afficher un message d'erreur. Lorsque la connexion est réussie, redirigez l'utilisateur vers la page d'accueil.

Champ	Validations
Email	<ul style="list-style-type: none"> Obligatoire et non vide (excluant les espaces). Le format doit être valide (voir filter_var).
Mot de passe	<ul style="list-style-type: none"> Obligatoire.



Pour se déconnecter, créez une page **signout.php** qui, lorsque accédée, déconnecte l'utilisateur et le redirige vers la page d'accueil. C'est de loin la manière la plus élégante de faire cette opération.

4.7 Page du panier

Créez la page de consultation du panier (**cart.php**) respectant la maquette ci-dessous.

[Produit](#)[Panier \(2\)](#)[Créer un compte](#)[Se connecter](#)

Votre panier

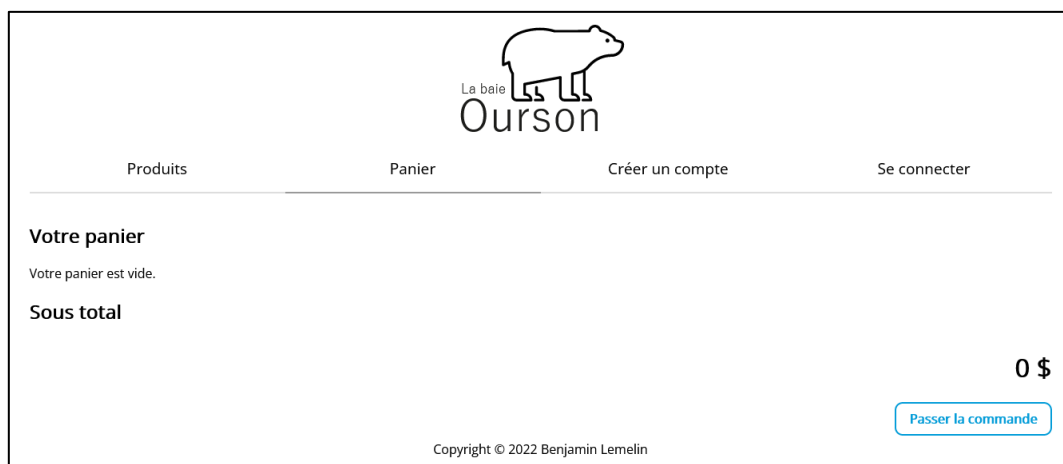
	T-shirt Blanc SM x1 <div>- 1 +</div>	9.99 \$ <div>Supprimer</div>
	Manteau d'hiver Ursidae x2 <div>- 2 +</div>	209.99 \$ <div>Supprimer</div>

Sous-total

429.97 \$

Préparer la facture

Chaque produit du panier doit être listé sur cette page, avec le sous-total (excluant les taxes) en bas de la page. Si le panier est vide, affichez un message indiquant que le panier est vide. N'oubliez pas de prendre en compte la quantité de chaque produit dans le calcul du sous-total. Enfin, notez que le panier doit être conservé dans la session de l'utilisateur et non pas sur la base de données.



⚠ Attention

Les données du panier doivent être représentées via une classe nommée **Cart**. Cette classe doit aussi être testée unitairement.



Pour utiliser des objets dans la session, il faut faire de la sérialisation :
<https://www.tutorialspoint.com/storing-objects-in-php-session>

Il doit être possible de supprimer les éléments du panier individuellement. Pour faire cela, il vous est conseillé de générer un formulaire avec chaque produit n'ayant qu'un seul bouton et où tous [les champs sont cachés](#).

	Manteau d'hiver Ursidae x2	209.99 \$
		Supprimer

4.8 Page de confirmation de commande

Il est possible de passer une commande seulement lorsqu'on s'est connecté préalablement. Lorsque l'utilisateur connecté clique sur le bouton « Passer la commande » dans le panier, sa commande doit être affichée avec le **total** et l'**adresse** de livraison. Ensuite, s'il confirme l'achat, la commande sera enregistrée (**checkout.php**) dans la base de données et une page de confirmation (**confirm.php**) doit être affichée. Le panier doit ensuite être vidé et les quantités doivent être mises à jour dans l'inventaire.

Facture		
Vos items:		
	T-shirt Blanc SM x1	9.99 \$
	Manteau d'hiver Ursidae x2	419.98 \$
Vos informations:		
Prénom:	Karine	
Nom:	Filiatreault	
Adresse:	123 des Sapins	
Total: 429.97 \$		
Passer la commande		



Information

Si l'utilisateur clique sur le bouton « Passer la commande », mais qu'il n'est pas connecté, il devrait être redirigé vers [la page de connexion](#).

4.9 Patrons obligatoires

Vous devez obligatoirement utiliser ces patrons dans le projet :

- Le patron `Post/Redirect/Get` pour chaque soumission de formulaire. Seule exception à cette règle : si l'utilisateur a des erreurs de validation, alors vous n'avez pas à le faire.
- Le patron `Data Access Object` pour vos accès à la base de données. Vous n'avez pas à tester ces classes non plus, car vous ne savez pas encore faire de *mocks*.

4.10 Informations supplémentaires

Voici quelques informations supplémentaires sur ce projet :

- Organisez vos fichiers correctement en séparant les fichiers qui ne contiennent que du php de ceux qui contiennent à la fois du HTML et du PHP.
- Chaque produit possède une image. L'image correspondante a le même nom que le `sku` du produit, suivi de `.png`. Par exemple, `154789.png` ou `887414.png`. Ces images sont déjà fournies.
- Certaines pages ne devraient être accessibles que si l'utilisateur est connecté à son compte. Empêchez tout accès à ces pages et redirigez l'utilisateur vers la page d'accueil.
- Vous devez prévoir une page d'erreur 404 conforme.
- Les validations sont seulement demandées en PHP.

4.11 Tests unitaires

Vous devez tester unitairement la classe `Cart`.



Prudence

Les tests unitaires sont importants dans la mesure où ils sont utiles. Par exemple, il est inutile de tester unitairement vos classes de formulaires, car il n'y a pour ainsi dire aucune logique qui mérite d'être testée. Le potentiel à l'erreur est pratiquement inexistant.

À l'inverse, le potentiel à l'erreur de la classe **Cart** est beaucoup plus élevé. Par exemple, avez-vous pensé au cas où un utilisateur ajouterait un item déjà présent dans le panier ? Dans cette situation, il faudrait simplement additionner les quantités au lieu de l'ajouter une seconde fois. C'est le genre de situation complexe qui mérite d'être bien testée.

Notez qu'il est possible de tester l'interface de votre site via des outils comme [Selenium](#). C'est aussi ce que l'on appelle des tests d'interface, et ils sont très utiles pour détecter les régressions. Cependant, vous ne savez pas encore comment en faire.

5 Collaboration et plagiat

Un travail pratique ne peut **en aucun temps** être le produit d'une collaboration ou de partage de code entre les équipes. Toute collaboration, partage de code¹, ou autre interaction sera sanctionnée selon la clause 6.1.12 du plan de cours :

Tout acte de plagiat, de tricherie et de fraude sera sanctionné. Constitue notamment un plagiat, une tricherie ou une fraude tout acte de copier, de fournir ou de recevoir volontairement de l'information lors d'un examen, de reproduire en tout ou en partie le travail d'une autre personne, qu'il s'agisse d'un document imprimé, multimédia ou électronique, sans y faire expressément référence, de remplacer un étudiant ou de se faire remplacer lors d'un examen, de remettre un travail réalisé par une autre personne, d'obtenir, posséder ou utiliser frauduleusement des questions ou réponses d'examen, d'utiliser du matériel, des applications, des sites Web ou des ressources non autorisés, de se faire aider par une autre personne lorsqu'il est demandé de réaliser l'évaluation seul, de falsifier les résultats de travaux ou d'examens.

Source : Politique d'évaluation des apprentissages du collège

Si de tels comportements sont observés en classe, le professeur se réserve le droit de mettre fin à l'évaluation des étudiants fautifs en cours de réalisation et d'attribuer la note 0.

6 Modalités de remise

6.1 Remise 1 – 27 août avant minuit

Remettez sur LÉA une archive ZIP contenant l'entièreté de votre projet (**incluant le script de votre BD**) pour :

- Page d'accueil
- Page de consultation des produits provenant de la BD ainsi qu'une page de détails pour chaque produit
- Authentification des clients (sécurité)
- Ajout du client (avec validation de l'information)
 - Toujours le mot de passe en double et les caractères cachés

¹ Du code identique avec des variables renommées et/ou des espaces introduits est considéré comme du plagiat.

- Mot de passe chiffré avec *password_hash*
 - Le même login ne peut pas être utilisé plusieurs fois
- Présentation uniforme
- Fichier texte avec vos noms.

N'incluez pas l'énoncé ni la grille de correction avec la remise : une pénalité de 10% sera appliqué si ces fichiers sont remis. Une pénalité de 15 % sera aussi appliqué en cas de retard de moins d'une journée. Au-delà de ce délai, le travail est refusé et la note de 0 % est automatiquement attribuée.

6.2 Remise 2 – 10 septembre avant minuit

Remettez sur LÉA un fichier ZIP contenant l'entièreté de votre projet et toutes les fonctionnalités du projet. Ajoutez-y aussi un fichier texte avec vos noms. Les nouvelles fonctionnalités sont :

- Achat en ligne seulement si l'inventaire le permet
- Panier d'achat modifiable (suppression et modification des quantités)
 - Le panier est modifiable en tout temps (client connecté ou non)
- Facture complète (incluant l'adresse du client)
 - La commande doit être ajoutée dans la base de données lorsque le client confirme l'achat (client connecté seulement)
 - L'inventaire doit être géré
- Tests unitaires de la classe **Cart**

N'incluez pas l'énoncé ni la grille de correction avec la remise : une pénalité de 10% sera appliqué si ces fichiers sont remis. Une pénalité de 15 % sera aussi appliqué en cas de retard de moins d'une journée. Au-delà de ce délai, le travail est refusé et la note de 0 % est automatiquement attribuée.

7 Évaluation

Fonctionnalités – Le Quoi
Page d'accueil : <ul style="list-style-type: none">• Respect de la maquette (personnalisation du style permise).• Grille de produits (nom, image et prix) provenant de la base de données.
Page de consultation de produit : <ul style="list-style-type: none">• Respect de la maquette (personnalisation du style permise).• Numéro du produit consulté obtenu à partir de l'url.• Détails sur le produit (nom, image, description et prix) provenant de la base de données.• Possibilité d'ajouter le produit au panier (quantité incluse).
Page du panier : <ul style="list-style-type: none">• Respect de la maquette (personnalisation du style permise).• Liste des produits dans le panier (nom et quantité) provenant de la session courante.• Détails sur les produits (nom et prix) provenant de la base de données.• Calcul du sous-total impliquant le prix de chaque produit et leur quantité.• Possibilité de retirer un produit du panier.• Affichage d'un message informatif en cas de panier vide.
Page de création de compte : <ul style="list-style-type: none">• Respect de la maquette (personnalisation du style permise).• Champ de texte pour l'adresse courriel (obligatoire, non vide et du bon format).• Champ pour le mot de passe (obligatoire et non vide).• Champ pour la confirmation du mot de passe (obligatoire et égal au mot de passe).• Champ pour le prénom (obligatoire et non vide).• Champ pour le nom (obligatoire et non vide).• Champ pour l'adresse de livraison (obligatoire et non vide).• Création du compte dans la base de données.• Chiffrage du mot de passe.• Redirection vers la page de connexion après création du compte.
Page de connexion à un compte : <ul style="list-style-type: none">• Respect de la maquette (personnalisation du style permise).• Champ de texte pour l'adresse courriel (obligatoire, non vide et du bon format).• Champ pour le mot de passe (obligatoire et non vide).• Vérification du compte à partir de la base de données.• Redirection vers la page d'accueil après connexion au compte.• Connexion au compte ajoute le compte à la session.• Déconnexion du compte retire le compte de la session.
Page de confirmation de commande : <ul style="list-style-type: none">• Respect de la maquette (personnalisation du style permise).• L'utilisateur ne peut passer une commande s'il n'est pas connecté.• Possibilité de passer la commande (enregistré dans la base de données, vide le panier, gérer inventaire).
Autres : <ul style="list-style-type: none">• Barre de navigation s'adaptant à la connexion et la déconnexion de l'utilisateur (toutes pages).• Liste de messages d'erreur de validation en haut des pages.• Redirection vers la page d'accueil ou une page 404 en cas de ressource non trouvée.

Compétences – Le Comment
<p>Préparer la base de données (00SU-3) (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Rédaction de requêtes SQL (syntaxe générale). • Utilisation de PDO (syntaxe générale, connexion et gestion des erreurs). • Utilisation du patron Data Access Object. • Utilisation de transaction lors des requêtes en écriture (si nécessaire). • Utilisation de requêtes préparées.
<p>Programmer la logique applicative coté serveur (00SU-5) (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Rédaction de code en PHP (syntaxe générale). • Séparation suffisante de la logique application et du visuel. • Utilisation du bon verbe Http (Get, Post) pour l'envoi et la réception de formulaires. • Utilisation du patron Post/Redirect/Get pour les soumissions de formulaires. • Utilisation du patron Form Object pour les données des formulaires et leur validation. • Usage de sessions.
<p>Programmer la logique applicative coté client (00SU-6) (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Rédaction de code en HTML (syntaxe générale). • Rédaction de code en Css (syntaxe générale).
<p>Contrôler la qualité de l'application (00SU-7) (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Rédaction de tests unitaires avec PHPUnit. • Tous les tests unitaires passent avec succès.
<p>Rédiger la documentation (00SU-9) (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Emplacement et nature des commentaires. • Réusinage du code.

Pénalités – La rigueur
<p>Qualité générale du code (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Erreur à l'interprétation du code. • Avertissement de l'interpréteur ou de l'IDE ignoré sans raison valable. • Duplication de code ou découpage en fonctions déficient. • Code malpropre, illisible ou comportant du code en commentaire. • Code « patché » visant à corriger une erreur se produisant ailleurs dans le code. • Classement malpropre des fichiers et des dossiers. • Formatage incorrect ou déficient. • Nommage ambiguë ou paresseux. • <i>Autres pénalités.</i>
<p>Remise du travail (incluant, mais sans s'y limiter) :</p> <ul style="list-style-type: none"> • Non-respect des consignes de remise. • Usage du mauvais type d'archive pour la remise du projet. • Fichier ou dossier inutile remis avec le projet. • <i>Autres pénalités.</i>

La qualité de la langue française fait partie de l'évaluation. Chaque faute de français retire 0,5 % à la note finale jusqu'à concurrence de 20 %.