

CÉGÉP de Chicoutimi

TP-01

Programmation Web Transactionnelle | 420-D51-CH

Henri Pitre (2432978)
2026-01-30

Rapport Écrit

1. Informations et Remise

De nombreux clients veulent pouvoir commander des produits à n'importe quelle heure du jour ou de la nuit. Le problème a donc trait à la façon de traiter ces commandes 24 heures par jour sans augmenter les frais de vente. P.4

2. Quelles sont les principales étapes à suivre pour résoudre un problème?

Liens pour accéder au site

- https://web.decinfo-cchic.ca/dev-2432978/S4/2432978-HPitre_Pharmacie/index.php (ACCUEIL)
- https://web.decinfo-cchic.ca/dev-2432978/S4/2432978HPitre_Pharmacie/login.php (CONNEXION)

Informations supplémentaires

- Le document CSS a été généré par IAg. Celui-ci a transféré le code CSS imbriqué un peu partout dans le HTML pour le restructurer dans un document à part entière pour aider à la clarté.
- La technologie utilisée dans vos exemples pour établir la connexion avec la base de données SQL était la PDO. En revanche, nous avons utilisé le MySQLi la session passée. Pour cette raison, j'ai peut-être utilisé quelques fonctions de PDO imparfaitement, et il est possible que je n'aie pas eu les connaissances de certaines fonctions qui auraient pu garantir une meilleure sécurité avec cette technologie.
- Une démonstration visuelle est demandée en annexe au projet afin de vous montrer le fonctionnement de notre site, cependant je crois que vous donner le lien du site fonctionnel en revient au même. De plus, je n'ai pas le temps de la réaliser; alors les commentaires dans le code ainsi que ce rapport écrit devrait être amplement suffisants pour votre compréhension.

- La base de données contient trois tables : Une pour la liste des produits existants, une pour les utilisateurs, ainsi qu'une pour les commandes passées et en cours(isActive).

2. Code en détail

Functions.php

Fichier de code php qui regroupe des fonctions générales utiles un partout pour le projet;

Les plus importantes :

- RenderTemplate : Fonction montrée en PrWeb1 : En recevant le html d'une page par le paramètre \$content, elle assemble la page en entier avec le header,menu,body et footer
- GetMySQLInstance : Une fonction démarrant la connexion avec la base de données SQL. C'est simplement le fichier cnx.php fourni en exemple, mais converti en une fonction pour simplifier le tout 😊

index.php (Accueil et liste des produits)

Page d'accueil du site qui salue le visiteur et liste les produits (Depuis la base de données), puis donne la possibilité de faire une commande d'un produit (si l'utilisateur est connecté)

S'il n'est pas connecté, le visiteur peut tout de même voir les produits, mais le bouton commander est remplacé par le message « Connectez-vous pour commander »

Commander.php

Un « script » php qui valide la création d'une nouvelle commande. Celui-ci est enclenché lorsque l'utilisateur entre une quantité et appuie sur le bouton commander sur la page index.php.

Cette page gère les erreurs de commandes possibles (article indisponible/erreurs...). Si la commande passe avec succès, elle redirige l'utilisateur vers l'onglets commandes (orders.php) pour voir les commandes actuelles.

orders.php

Page qui demande à la base de données pour toutes les commandes en cours et passées, les traite et les affiche en rendu formaté.

Une commande est toujours constituée d'un seul article, mais d'une quantité donnée (La gestion d'un panier de commande aurait été plus complexe).

La page se sépare en deux parties verticales : Une pour les commandes qui ont le paramètre isActive à 1 (true), qui indique qu'elles sont en cours :

```
try {
    $pdo = GetMySQLInstance();

    // get commandes en cours (isActive = 1)
    $sqlActive = "SELECT o.id, o.userId, o.orderDate, o.price, o.isActive, o.productId
                  d.name AS drugName, d.link AS drugLink
                  FROM pwt_tp1_orders o
                  JOIN pwt_tp1_drugs d ON d.id = o.productId
                  WHERE o.userId = ? AND o.isActive = 1
                  ORDER BY o.orderDate DESC";

    $stmtActive = $pdo->prepare($sqlActive);
    $stmtActive->execute([$SESSION['userId']]);
    $activeOrders = $stmtActive->fetchAll();

    <?php foreach ($activeOrders as $order) : //Show le meme template de "carte" pour chaque medicament?>
        <tr>
            <td><?=(int)$order['id']; ?></td>
            <td><?= htmlspecialchars($order['drugName']); ?></td>
            <td>
                <?php if (!empty($order['drugLink']) : ?>
                    "
                <?php else : ?>
                    <span>Aucune image</span>
                <?php endif; ?>
            </td>
            <td><?= number_format($order['price'], 2, ',', ' '); ?> $</td>
            <td><?= htmlspecialchars($order['orderDate']); ?></td>
        </tr>
    <?php endforeach; ?>
```

Et la seconde partie avec celles passées :

```
// get commandes passées (isActive = 0)
$sqlPast = "SELECT o.id, o.userId, o.orderDate, o.price, o.isActive, o.productId
            d.name AS drugName, d.link AS drugLink
            FROM pwt_tp1_orders o
            JOIN pwt_tp1_drugs d ON d.id = o.productId
            WHERE o.userId = ? AND o.isActive = 0
            ORDER BY o.orderDate DESC";

$stmtPast = $pdo->prepare($sqlPast);
$stmtPast->execute([$SESSION['userId']]);
$pastOrders = $stmtPast->fetchAll();
```

```

<?php foreach ($pastOrders as $order) : //Show le meme template de "carte" pour chaque medicament?>
    <tr>
        <td><?=(int)$order['id']; ?></td>
        <td><?= htmlspecialchars($order['drugName']); ?></td>
        <td>
            <?php if (!empty($order['drugLink'])) : ?>
                " />
            <?php else : ?>
                <span>Aucune image</span>
            <?php endif; ?>
        </td>
        <td><?= number_format($order['price'], 2, ',', ' '); ?> $</td>
        <td><?= htmlspecialchars($order['orderDate']); ?></td>
    </tr>
<?php endforeach; ?>

```

Login.php

Formulaire de connexion simple, avec un bouton qui redirige à la page-script « login-validation.php ».

Se bloque après 3 tentatives erronées :

```

<?php
if ($exceedsTries) { //si bloqué car trop d'essais
    echo '<p> Accès bloqué : trop de tentatives infructueuses.</p>';
    echo '
        <form method="post" action="">
            <button type="submit" name="restart">Recommencer!</button>
        </form>';
}

```

Login-validation.php

Reçoit les informations de connexion en POST de login.php, et les confirme. Si les données concordent avec celles d'un utilisateur dans la base de données, il enregistre les données utilisateur en variable \$_SESSION et redirige l'utilisateur vers l'Accueil (index.php).

Dans le cas contraire, il renvoie l'utilisateur à login.php en spécifiant une tentative erronée.

Bonne correction !

Henri Pitre- 2432978