

Document d'exploitation

- Le manuel d'utilisation

1. Inscription

Fonctionnalités :

- Permet aux utilisateurs de créer un compte en fournissant les informations suivantes :
 - **Pseudo**
 - **E-mail**
 - **Mot de passe**
 - **Vérification du mot de passe**
- Les données saisies sont validées avant d'être insérées dans la base de données.

Interface utilisateur

Resto Web Accueil Panier Connexion Inscription Non connecté

Inscription

Pseudo

Mail

Mot de passe

Vérification du mot de passe

S'inscrire

Code :

Traitement PHP (dans `inscription.php`)

```
if ($submit){  
    //Rajouter des isset avec les erreurs liées  
    $loginUtil = $_POST['loginUtil'];  
    $mail = $_POST['mail'];  
    $pwd = $_POST['pwd'];  
    $pwd_check = $_POST['pwd_check'];  
  
    //rejouer execute si tous les isset sont ok  
    $error_message=add_user_db();  
}
```

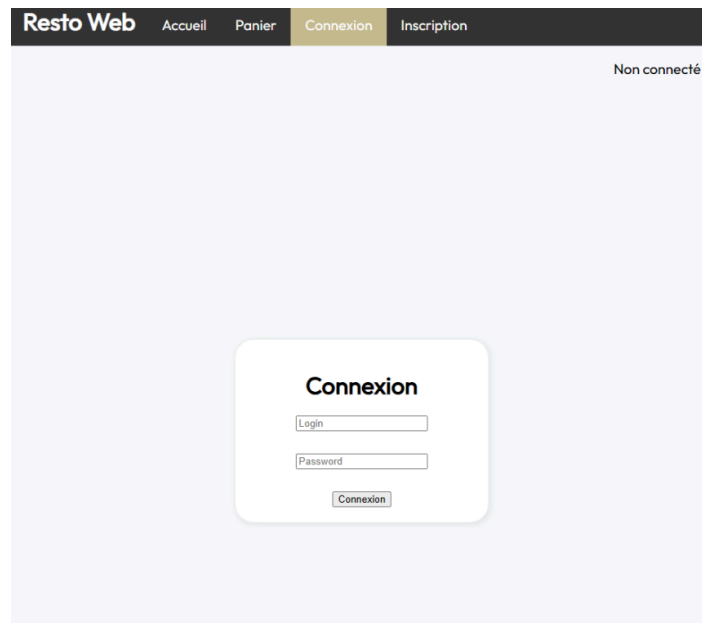
On utilise des champs html, pour pouvoir récupérer les informations de l'utilisateur et les attribuer à des variables. Ces informations sont ensuite passées en session pour pouvoir les réutiliser sans passer par des requêtes SQL.

2. Connexion

Fonctionnalités :

- Permet aux utilisateurs de se connecter à leur compte avec leur **login** et **mot de passe** .
- Redirige l'utilisateur vers la page d'accueil en cas de succès.

Interface utilisateur



Code :

Algorithme PHP (dans `connection.php`)

```
session_start();

// Initialisation de variables pour les messages d'erreur
$error_message = null;
$submit = isset($_POST['submit']);

if ($submit) {
    $login = isset($_POST['login']) ? $_POST['login'] : "";
    $password = isset($_POST['password']) ? $_POST['password'] : "";

    // Vérification que les champs ne sont pas vides
    if (!empty($login) && !empty($password)) {
        // Appeler la fonction pour vérifier l'utilisateur dans la base de données
        if (login_user_db($login, $password)) {
            // Si l'utilisateur est trouvé et le mot de passe est correct
            header("Location: ../../index.php"); // Redirection vers la page d'accueil
            exit();
        } else {
            // Message d'erreur si le login ou mot de passe est incorrect
            $error_message = "Login ou mot de passe incorrect.";
        }
    } else {
        // Message d'erreur si les champs sont vides
        $error_message = "Veuillez remplir tous les champs.";
    }
}

?>
```

Ce code en php indique que si le bouton connexion est cliqué, on récupère les login et password qui ont été mis en session.

On vérifie que les champs ne soient pas vide, si c'est le cas on vérifie à l'aide de la fonction login_user_db() que les login et password sont corrects et si c'est le cas on redirige vers la page d'accueil.

Dans le cas contraire un message d'erreur s'affiche en disant que le login ou le mot de passe est incorrect.

3. Commande

Fonctionnalités :

- Affichez les produits dans le panier avec leurs informations (quantité, prix unitaire, total).
- Calculez le total HT pour le panier.
- Validez le panier en enregistrant la commande et ses lignes dans la base de données.

Interface utilisateur

Resto Web Accueil **Panier** Deconnexion

Connecté : nouhbg

Total

Produit(s)	Quantité(s)	Prix	Total
Burger Classic	1	7.20	7.2
Pizza Margherita	1	8.50	8.5
Spaghetti Bolognese	1	9.30	9.3
Tarte aux Pommes	1	4.50	4.5

Total HT: 29.5 € ☐ à emporter ?

Valider

Code :

Affichage des produits dans le panier (dans panier.php)

```
echo "<table>
<tr>
<th>Produit(s)</th>
<th>Quantité(s)</th>
<th>Prix</th>
<th>Total</th>
</tr>";
$jj=0;
foreach($scart as $products){

    foreach($products as $product){

        $id=$product['productId'];
        $qt=$product['quantity'];
        $product=get_product($id);
        $lib=$product['libProduit'];
        $prix=$product['prixProHT'];
        $total=$prix*$qt;
        $totalHT+=$total;
        echo "<tr>
<td><input type='hidden' name='idPro' . $j . '' value='' . $id . ''> . $lib . "</td>
<td><input type='hidden' name='qtLigne' . $j . '' value='' . $qt . ''> . $qt . "</td>
<td><input type='hidden' name='prixProHT' . $j . '' value='' . $prix . ''> . $prix.</td>
<td><input type='hidden' name='totalLigneHT' . $j . '' value='' . $total . ''> . $total.</td>
</tr>";

        $jj++;
    }
}

echo "<input type='hidden' name='totalHT' value='' . $totalHT . ''>";
echo "</table>";
```

La page panier utilise la session pour récupérer les différents produits et leurs quantités pour pouvoir les afficher à l'aide d'une boucle foreach. Mais pour l'affichage sous forme de quadrillage, il est nécessaire d'utiliser un autre foreach.

Insertion de la commande (dans `insert_commande.php`)

```
1 <?php
2 function insert_commande($totalComTTC,$typeCom,$idUtil){
3     $dbh=connect_db();
4     $sql="INSERT INTO commande (etatCom, totalComTTC, typeCom, dateCom, heureCom, idUtil) VALUES (:etatCom, :totalComTTC, :typeCom, :dateCom, :heureCom, :idUtil)";
5     try{
6         $sth=$dbh->prepare($sql);
7         $sth->execute(array(':etatCom'=>0,':totalComTTC'=>$totalComTTC,':typeCom'=>$typeCom,':dateCom'=>date('Y-m-d'),':heureCom'=>date('H:i:s'),':idUtil'=>$idUtil));
8         $insertState=true;//Insertion réussie
9     }catch(PDOException $e){
10         die("Erreur dans la fonction insertCommande : ".$e->getMessage());
11         $insertState=false;//Insertion échouée
12     }
13     //On enregistre que l'on vient de créer dans la session pour s'en servir dans la page payer.php
14     $_SESSION['idCom']=$dbh->lastInsertId();
15
16     return $insertState; //Retour du statut de l'insertion
17 }
18
19
20
21 >>
```

Lorsque le bouton commander est cliqué, une requête SQL s'exécute pour pouvoir ajouter dans la table commande les différents champs nécessaires.

La requête est préparée puis exécutée pour pouvoir éviter les injections sql. Si la requête génère une erreur, on utilise une variable qui est renseigné par PDOException pour renvoyer le message d'erreur de MySQL.

4. Paiement

Fonctionnalités :

- Permet à l'utilisateur d'entrer ses informations bancaires pour payer sa commande.
- Met à jour l'état de la commande dans la base de données après validation.

Interface utilisateur

Resto Web Accueil Panier Deconnexion

Connecté : nouhbg

Paiement de votre commande

Total de la commande : 32.45 €

N° de CB

Date expiration

CCV

Payer

Code :

Formulaire HTML (dans `payer.php`)

```
<form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
  <input type="text" name="cb_number" placeholder="N° de CB" required="required">
  <input type="text" name="exp_date" placeholder="Date expiration" required="required">
  <input type="text" name="ccv" placeholder="CCV" required="required">
  <input type="submit" name="submit" value="Payer">
</form>
<?php
if($paymentState==true){
    echo "<p class='form_success_message'>Paielement effectué avec succès</p>";
    echo "<p class='form_success_message'>Vous allez être redirigé vers la page de confirmation de commande</p>";
}
?>
```

Pour le paiement, le client renseigne les champs de la carte bleue en vérifiant que ces champs sont bien renseignés.

Mise à jour de la commande (dans `payer.php`)

```
try {
    $etatCom = "Calculée";
    $sql = "UPDATE commande SET etatCom = :etatCom WHERE idCom = :idCom;";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([
        ':etatCom' => $etatCom,
        ':idCom' => $idCom
    ]);
}
```

Une fois le bouton payer cliquer, on fait une requête UPDATE pour la base de donnée pour pouvoir mettre à jour l'état de la commande.