

## Groupe G2A-9

---

# Récapitulatif PHP SAÉ S3

---

Thomas Demeyere

Anton Xu

Louis Yvelin

Oryann Prochaska

# Sommaire

---

<b>Sommaire</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
<b>1- Arborescence du site</b>	<b>4</b>
<b>2 - Connexion à la base</b>	<b>6</b>
<b>3 - Page index</b>	<b>7</b>
<b>4 - La page connexion</b>	<b>9</b>
<b>5 - La page produit</b>	<b>10</b>
<b>6 - La page compte</b>	<b>12</b>
<b>7 - La page inscription</b>	<b>15</b>
<b>8 - La page panier</b>	<b>17</b>
<b>9 - La page Rechercher</b>	<b>18</b>

# Introduction

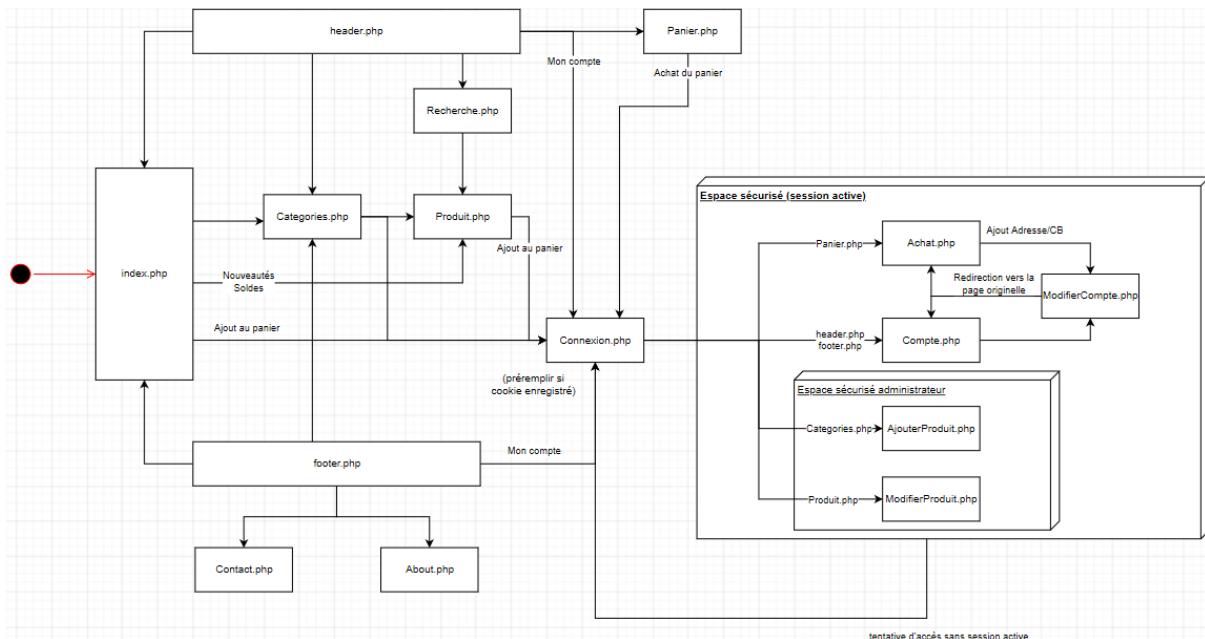
---

Ainsi s'achève une SAÉ de S3 longue de six semaines, dans celle-ci, nous avons dû coder un site web pour l'entreprise nommée "*Commerce de la Rue*". Un site statique purement informatif ne contient généralement que du HTML pour la forme/structure et du CSS pour l'aspect esthétique.

Dans notre cas, nous devions le rendre dynamique, mais aussi le lier à une base de données contenant toutes les données clients. Afin de rendre le site plus dynamique sans avoir besoin de recharger la page, nous avons utilisé un peu de JavaScript. Pour établir une liaison à la base et formuler les requêtes SQL ou appeler les procédures, nous avons dû utiliser le langage PHP.

La liaison à la base de données, la page produit et l'affichage des images a été faite par Mlle Prochaska, la partie connexion utilisateur, ainsi qu'administrateur par M. Xu. La partie inscription et modification information utilisateur et catégories par M. Yvelin. La partie recherche produit avec application de filtres ainsi que l'inclusion du JavaScript dans les pages par M. Demeyere.

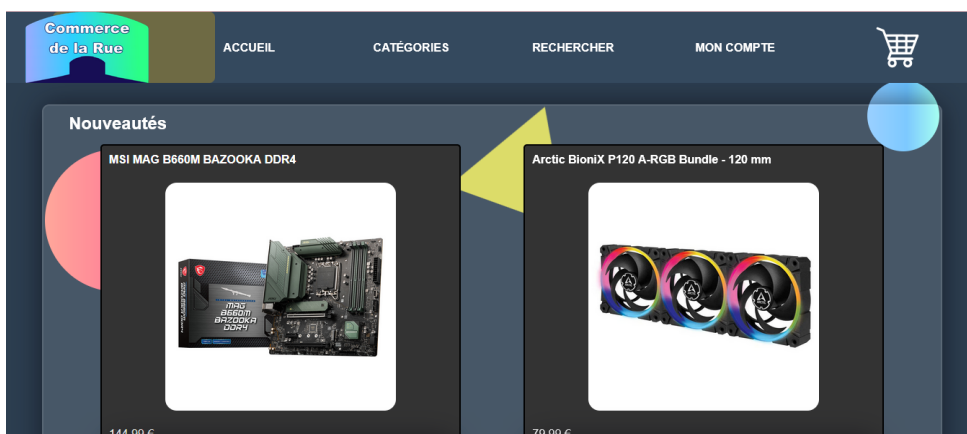
# 1- Arborescence du site



Nous avons réalisé l'arborescence du site de manière à ce que certaines pages soient accessibles seulement par les utilisateurs et/ou administrateur connectés que nous appelons dans le schéma ci-dessus "espace sécurisé".

La première page visible du site par défaut est celle portant le nom de "index.php", sur celle-ci, nous affichons les nouveautés ainsi que les promotions sur demande de l'entreprise commanditaire. Depuis cette page, toutes les autres pages du site (excepté l'espace sécurisé) y sont accessibles.

Dans le menu présent sur le haut de la page seuls les pages Accueil, Catégories, Recherche, Mon compte et Panier. Quant aux pages Contact, Notre entreprise, nous les avons placées dans le pied de page sur indication de la maquette donnée.





## 2 - Connexion à la base

```
<?php
$db = "(DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = oracle.iut-blagnac.fr)(PORT = 1521))
        (CONNECT_DATA =
            (SERVER = DEDICATED)
            (SID = db11g)
        )
    )" ;
$conn = oci_connect("SAEBD09", "M0ntBlanc1UT", $db);

// si la connexion a échoué, on affiche le message d'erreur
if (!$conn) {
    $e = oci_error();
    trigger_error(htmlentities($e['message'], ENT_QUOTES), E_USER_ERROR);
    echo $e;
}

?>
```

Pour se connecter à notre base de données, il nous faut le document connect.inc.php qui va nous permettre de rentrer le nom de la base, le port, le SID ainsi que le mot de passe donnant accès aux données. Nous prenons soin de gérer les potentielles erreurs de connexion en reportant un message d'erreur visible sur le site.

Ce fichier sera obligatoirement inclus dans toutes les autres pages du site afin de garantir le bon fonctionnement.

## 3 - Page index

```
//nouveauautés
$query2 = "SELECT NOMPRODUIT, IDPRODUIT, PRIXPRODUIT FROM produit WHERE dateproduit BETWEEN ADD_MONTHS(sysdate, -6) and sysdate and rownum <= 4";
$stmt2 = oci_parse($conn, $query2);
oci_execute($stmt2);

while ($row2 = oci_fetch_array($stmt2, OCI_ASSOC)) {
    $res2[] = ['nom' => $row2['NOMPRODUIT'], 'id'=> $row2['IDPRODUIT'], 'prix' => $row2['PRIXPRODUIT']];
}
```

Dans notre page index.php, nous voulons afficher les quatre dernières nouveautés toutes catégories confondues. Après avoir fait un include du connect.inc.php et commencé une session à l'aide de session\_start(), nous constituons une requête SQL pour trouver les produits ayant une date d'ajout récente. Ajout situé entre la date système (soit la date locale sur le serveur) et la date du système six mois avant. La variable stmt2 permet de se connecter à la base et de rentrer cette requête. Le ou les résultats obtenus seront parcourus à l'aide de oci\_fetch\_array(), ceux-ci seront mis dans un certain ordre pour composer par la suite l'affichage.

```
// produits en réductions
$query3 = "SELECT NOMPRODUIT, IDPRODUIT, PRIXPRODUIT, (PRIXPRODUIT - REDUCTION) as REDUC FROM produit WHERE reduction > 0 and rownum <= 4";
$stmt3 = oci_parse($conn, $query3);
oci_execute($stmt3);

while ($row3 = oci_fetch_array($stmt3, OCI_ASSOC)) {
    $res3[] = ['nom' => $row3['NOMPRODUIT'], 'id'=> $row3['IDPRODUIT'], 'prix' => $row3['PRIXPRODUIT'], 'reduc' => $row3['REDUC']];
}
```

Nous faisons de même pour les produits en réduction.

```
oci_free_statement($stmt);
oci_free_statement($stmt2);
oci_free_statement($stmt3);
oci_close($conn);

?>
```

Puis, nous libérons l'espace sans pour autant fermer la connexion si jamais nous avons de nouveau besoin de faire des requêtes dans cette page.

```
<?php
foreach($res2 as $produit) {
    echo " <div class=\"produit\">
        <div><a><strong>\".$produit['nom'].\"</strong></a></div>
        <div class=\"image-produit-content\"><img class=\"image-produit\" src=\"./img/produits/\".$produit['id'].\"_1.jpg\" alt=\"Image
        <div><a>\".$produit['prix'].\" €</a></div>
        <form name= \"aie\" action=\"produit.php\" method= \"get\">
        <div><button name= \"identifiantP\" type= \"submit\" value= \".$produit['id'].\">Acheter</button></div>
        </form>
    </div>";
}


?>
```

Nous allons à présent procéder à l'affichage des produits provenant des requêtes effectuées en mettant en forme grâce à du HTML / CSS. Pour cela, nous faisons une boucle foreach parcourant l'ensemble des produits renvoyés par la base. Nous affichons dans un "echo" tout d'abord le nom du produit qui sera affiché en haut, puis l'image correspondante. Les images produits ne sont pas stockées dans une base de données, mais dans un dossier.

Nous donnons donc le chemin relatif ainsi que le nom de l'image et son numéro (sachant qu'il y a à chaque fois deux images par produit).

Nous affichons ensuite son prix, puis nous insérons le bouton "Acheter" qui va rediriger l'utilisateur vers la page détaillée du produit en question grâce à une variable de session.

**Arctic BioniX P120 A-RGB Bundle - 120 mm**



79.99 €

**Acheter**



## 4 - La page connexion

---

```
<?php error_reporting(E_ERROR | E_PARSE); ?>
<?php
session_start();
extract($_POST);
if (isset($valider)) {
    include("include/connect_inc.php");
```

error\_reporting() nous permet de spécifier les erreurs reportées, dans notre cas, on prend seulement les erreurs du runtime.

```
//EMAIL
$query = "SELECT * FROM UTILISATEUR WHERE EMAILUSER LIKE :email";
$stmt = oci_parse($conn, $query);

oci_bind_by_name($stmt, ":email", $email);

oci_execute($stmt);

if ($row = oci_fetch_array($stmt, OCI_ASSOC)){
    $_SESSION["email"] = $row['EMAILUSER'];
    $res["emailUser"] = true;
    $res["mdpUser"] = $row['MDPUSER'];
}
else $res["emailUser"] = false;
```

Nous récupérons dans le HTML le champ de texte avec l'id "email" que nous insérons dans notre requête afin de tester si l'utilisateur existe bel et bien. La commande OCI\_ASSOC indique que oci\_fetch\_array() devra nous retourner un tableau associatif. Nous déclarons une variable de session email qui est associée à l'email utilisateur trouvé dans la base, oci\_bind\_by\_name() permet d'éviter les injections SQL malveillantes ainsi que de lier une variable PHP à un attribut de la base.

## 5 - La page produit

---

```
6
7     $id = htmlspecialchars($_GET['identifiantP']);
8
9     if (isset($_POST['ajout'])) {
10         $query = "begin AjouterPanier(:user, :idproduit, 1); end;";
11         $stid = oci_parse($conn, $query);
12
13         oci_bind_by_name($stid, ":user", $_SESSION['email']);
14         oci_bind_by_name($stid, ":idproduit", $_POST['ajout']);
15
16         oci_execute($stid);
17         oci_free_statement($stid);
18         echo "<script>alert('Article ajouté au panier');</script>";
19     }
20
```

Depuis cette page, nous allons tout d'abord récupérer l'identifiant produit contenu dans le formulaire de la page précédente. Cela fera que lorsqu'on clique sur un produit, ce soient les informations du produit en question qui sont affichées.

Nous appelons la procédure stockée AjouterPanier dans une requête afin d'ajouter le produit au panier si l'utilisateur clique sur un certain bouton.

```
//prix + description
$query = "SELECT NOMPRODUIT, PRIXPRODUIT, DESCPRODUIT, IDPRODUIT, IDSOUSCAT FROM produit WHERE idproduit = :id";
$stid = oci_parse($conn, $query);

oci_bind_by_name($stid, ":id", $id);

oci_execute($stid);

while ($row = oci_fetch_array($stid, OCI_ASSOC)) {
    $res[] = ['nom' => $row['NOMPRODUIT'], 'id' => $row['IDPRODUIT'], 'prix' => $row['PRIXPRODUIT'], 'desc' => $row['DESCPRODUIT'], 'souscat' => $row['IDSOUSCAT']]
}

oci_free_statement($stid);
```

Nous affichons le prix et la description du produit

```

// si produit non existant
if (empty($res)) {
    header("Location: index.php");
}

//produits similaires
$query = "SELECT NOMPRODUIT, IDPRODUIT, PRIXPRODUIT, IDSOUSCAT FROM produit WHERE IDSOUSCAT LIKE :souscat";
$stmt2 = oci_parse($conn, $query);

oci_bind_by_name($stmt2, ":souscat", $res[0]['souscat']);
oci_execute($stmt2);

while ($row2 = oci_fetch_array($stmt2, OCI_ASSOC)) {
    $res2[] = ['nom' => $row2['NOMPRODUIT'], 'id'=> $row2['IDPRODUIT'], 'prix' => $row2['PRIXPRODUIT'], 'idsouscat2' => $row2['IDSOUSCAT']];
}

oci_free_statement($stmt2);
oci_close($conn);
?>

```

Dans le cas où le produit n'existerait pas, l'utilisateur sera redirigé vers la page index.php. Une sélection de produits similaires sous critère qu'ils soient de la même sous-catégorie est effectuée pour faire un affichage en bas de la page.

## 6 - La page compte

```
//Utilisateur
$query = "SELECT * FROM UTILISATEUR WHERE EMAILUSER LIKE :email";
$stmtid = oci_parse($conn, $query);

oci_bind_by_name($stmtid, ":email", $_SESSION['email']);

$res = oci_execute($stmtid);

while($row = oci_fetch_array($stmtid, OCI_ASSOC)){
    $email = $row['EMAILUSER'];
    $nom = $row['NOMUSER'];
    $prenom = $row['PRENOMUSER'];
    $tel = $row['TELUSER'];
}
```

La variable de session utilisée ici est l'email de l'utilisateur validé et redirigé par la page connexion.

```
//Ajout CB
$numcb = str_replace(' ', '', $numcb);

if(isset($addCB)){
    if (!preg_match("/[0-9]{16}/", $numcb)) $erreur = "Numéro de carte bancaire invalide.";
    else if (!preg_match("/.{1,64}/", $nomcb)) $erreur = "Nom invalide.";
    else if (!preg_match("/[0-9]{3,4}/", $cryptocb)) $erreur = "Cryptogramme invalide.";
    else{
        $query = "INSERT INTO CARTEBANCAIRE (idCb, numeroCb, nomCb, dateCb, cryptoCb, emailuser)
VALUES(CB_SEQ.NEXTVAL, :numeroCb, :nomCb, TO_DATE(:dateCb,'YYYY-MM-DD'), :cryptoCb, :emailuser)";
        $stmtid = oci_parse($conn, $query);

        oci_bind_by_name($stmtid, ":numeroCb", $numcb);
        oci_bind_by_name($stmtid, ":nomCb", $nomcb);
        oci_bind_by_name($stmtid, ":dateCb", $datecb);
        oci_bind_by_name($stmtid, ":cryptoCb", $cryptocb);
        oci_bind_by_name($stmtid, ":emailUser", $_SESSION['email']);

        $res = oci_execute($stmtid);

        header("Location: Compte.php");
    }
}
```

Nous avons la possibilité d'ajouter une carte bancaire pour de futurs paiements de commandes. Cela nécessite de réaliser une insertion dans la base de données.

```

//Ajout Adresse
if(isset($addAdresse)){
    if (!preg_match("/.{0,64}/", $alias)) $erreur = "Alias invalide.";
    else if (!preg_match("/.{1,45}/", $ville)) $erreur = "Nom de ville invalide.";
    else if (!preg_match("/.{1,128}/", $adresse)) $erreur = "Adresse invalide.";
    else if (!preg_match("/[0-9]{5}/", $code_postal)) $erreur = "Code-postal invalide.";
    else if (!preg_match("/.{0,64}/", $complement)) $erreur = "Complément d'adresse invalide.";
    else{
        $query = "INSERT INTO ADRESSE (idAdresse, surnomadresse, ville, adresse, codepostal, complement, emailuser)
VALUES(ADRESSE_SEQ.NEXTVAL, :alias, :ville, :adresse, :code_postal, :complement, :emailuser)";
        $stid = oci_parse($conn, $query);

        oci_bind_by_name($stid, ":alias", $alias);
        oci_bind_by_name($stid, ":ville", $ville);
        oci_bind_by_name($stid, ":adresse", $adresse);
        oci_bind_by_name($stid, ":code_postal", $code_postal);
        oci_bind_by_name($stid, ":complement", $complement);
        oci_bind_by_name($stid, ":emailUser", $_SESSION['email']);

        $res = oci_execute($stid);

        header("Location: Compte.php");
    }
}

```

L'adresse doit respecter un certain motif que nous testons à l'aide d'expressions régulières, le caractère "." désigne un caractère quelconque et les intervalles tels que {1, 128} indiquent la longueur minimale et maximale de la chaîne de caractères. Pour le code postal, il est indiqué que si c'est différent de cinq chiffres entre 0 et 9 alors ce n'est pas valide. Nous insérons ensuite la nouvelle adresse souhaitée par l'utilisateur.

```

//Supprimer CB
if(isset($delCB)){
    $query = "DELETE FROM CARTEBANCAIRE WHERE IDCْب LIKE :idCB AND EMAILUSER LIKE :email";
    $stid = oci_parse($conn, $query);

    oci_bind_by_name($stid, ":idCB", $idCB);
    oci_bind_by_name($stid, ":email", $_SESSION['email']);

    $res = oci_execute($stid, OCI_COMMIT_ON_SUCCESS);

    header("Location: Compte.php");
}

//Supprimer Adresse
if(isset($delAdresse)){
    $query = "DELETE FROM ADRESSE WHERE IDADRESSE LIKE :idAdresse AND EMAILUSER LIKE :email";
    $stid = oci_parse($conn, $query);

    oci_bind_by_name($stid, ":idAdresse", $idAdresse);
    oci_bind_by_name($stid, ":email", $_SESSION['email']);

    $res = oci_execute($stid, OCI_COMMIT_ON_SUCCESS);

    header("Location: Compte.php");
}

```

Si l'utilisateur souhaite supprimer une carte bancaire ou une adresse enregistrée, il ne lui sera pas demandé de confirmation et cette action est irréversible, car directement supprimée de la base de données avec la requête commençant par DELETE.

```
//Affichage Adresse
$query = "SELECT * FROM ADRESSE WHERE EMAILUSER LIKE :email";
$listeadresses = oci_parse($conn, $query);

oci_bind_by_name($listeadresses, ":email", $_SESSION['email']);

$res = oci_execute($listeadresses);

?>
```

## 7 - La page inscription

---

```
if (isset($valider)) {  
    //EMAIL  
    $query = "SELECT * FROM UTILISATEUR WHERE EMAILUSER LIKE '$email'";  
    $stid = oci_parse($conn, $query);  
    oci_execute($stid);  
  
    //Pour vérifier le mot de passe plus bas  
    $uppercase = preg_match('@[A-Z]@', $password);  
    $lowercase = preg_match('@[a-z]@', $password);  
    $number     = preg_match('@[0-9]@', $password);  
    $specialChars = preg_match('@[^\w]@', $password);
```

Nous commençons par faire une requête récupérant les mails identiques à celui renseigné. Puis, nous initialisons des variables booléennes validant la conformité du mot de passe. Celui-ci doit contenir au moins : une majuscule, une minuscule, un chiffre, et un caractère spécial.

```
if ($row = oci_fetch_array($stid, OCI_ASSOC)) $erreur = "Email déjà utilisé, connectez-vous ou réinitialisez votre mot de passe.";  
if ($row === false){  
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
        $erreur = "Le format de l'email est incorrect.";  
    }  
    else if (!$uppercase || !$lowercase || !$number || !$specialChars || strlen($password) < 8) {  
        $erreur = "Le mot de passe doit contenir huit caractères dont une majuscule, une minuscule, un chiffre et un caractère spécial.";  
    }  
    else if (!preg_match("/^[a-zA-Z-' ]*$/", $nom)) {  
        $erreur = "Le prénom peut seulement contenir des lettres, des tirets et des espaces.";  
    }  
    else if (!preg_match("/^[a-zA-Z-' ]*$/", $prenom)) {  
        $erreur = "Le nom peut seulement contenir des lettres, des tirets et des espaces.";  
    }  
    else if (preg_match('/^[0-9]{8, 10}$/', $numtel)) {  
        $erreur = "Le numéro de téléphone doit être entre 8 et 10 chiffres.";  
    }  
}
```

Pour inscrire un nouveau client, nous devons impérativement enregistrer son adresse email, car celle-ci lui servira de nom d'utilisateur lors de ses connexions. Nous utilisons la requête au SGBD afin de vérifier que cette adresse n'existe pas déjà en s'assurant qu'elle ne renvoie rien. Nous vérifions à l'aide d'expressions régulières la composition du mot de passe afin que celui-ci soit sécurisé. Il faut que celui-ci possède une minuscule, une majuscule, un chiffre, un caractère spécial et qu'il fasse une taille minimale de huit caractères.

```

else{
    $query = "INSERT INTO utilisateur (emailUser,mdpUser,adminUser,nomUser,prenomUser,telUser,compteEntreprise)
VALUES (:email, :password, 0, :nom, :prenom, :numtel, :entreprise)";
    $stid = oci_parse($conn, $query);

    $password_hash = password_hash($password, PASSWORD_DEFAULT);
    $isset_entreprise = isset($entreprise) ? 1 : 0;

    oci_bind_by_name($stid, ":email", $email);
    oci_bind_by_name($stid, ":password", $password_hash);
    oci_bind_by_name($stid, ":nom", $nom);
    oci_bind_by_name($stid, ":prenom", $prenom);
    oci_bind_by_name($stid, ":numtel", $numtel);
    oci_bind_by_name($stid, ":entreprise", $isset_entreprise);

    $res = oci_execute($stid);
    if (!$res) {
        $e = oci_error($stid); // on récupère l'exception liée au pb d'execution de la requete
        $error_res = htmlentities($e['message'], ENT_QUOTES, 'UTF-8');
    }

    header("Location: Connexion.php");
}
oci_free_statement($stid);

```

S'il n'y a aucune des erreurs mentionnées, alors nous pouvons procéder à l'insertion des données de l'utilisateur dans la base de données. Nous envisageons la possibilité qu'il y ait un problème d'exécution et faisons afficher un message en conséquence si cela devait se produire.



## 8 - La page panier

```
if (isset($delproduit)) {  
    $query = "DELETE FROM Panier  
              WHERE Panier.emailUser = :email  
              AND idProduit = :idProduit";  
  
    $stid = oci_parse($conn, $query);  
  
    oci_bind_by_name($stid, ":email", $_SESSION['email']);  
    oci_bind_by_name($stid, ":idProduit", $delproduit);  
  
    oci_execute($stid);  
  
    oci_free_statement($stid);  
}
```

Cette instruction conditionnelle permet de pouvoir supprimer un article du panier de l'utilisateur en envoyant la requête à la base de données.

```
$query = "SELECT Pr.idProduit, nomProduit, prixProduit, reduction, quantite  
          FROM Produit Pr, Panier Pa  
          WHERE Pr.idProduit = Pa.idProduit  
          AND Pa.emailUser = :email";  
  
$stid = oci_parse($conn, $query);  
  
oci_bind_by_name($stid, ":email", $_SESSION['email']);  
  
oci_execute($stid);  
  
$res = [];  
$sum = 0;  
oci_fetch_all($stid, $res, null, null, OCI_FETCHSTATEMENT_BY_ROW|OCI_ASSOC);  
  
oci_free_statement($stid);  
?>
```

Cette partie du code permet de sélectionner les produits ajoutés dans le panier de l'utilisateur en envoyant une requête à la base de données.

## 9 - La page Rechercher

---

```
if (isset($_GET['recherche'])){
    $recherche = htmlspecialchars($_GET['recherche']);
} else {
    $recherche = '';
}
if (isset($_GET['categorieRecherchee'])){
    $categorieRecherchee = htmlspecialchars($_GET['categorieRecherchee']);
} else {
    $categorieRecherchee = '';
}
if (isset($_GET['sousCategorieRecherchee'])){
    $sousCategorieRecherchee = htmlspecialchars($_GET['sousCategorieRecherchee']);
} else {
    $sousCategorieRecherchee = '';
}
if (isset($_GET['prixMin'])){
    $prixMin = htmlspecialchars($_GET['prixMin']);
} else {
    $prixMin = '';
}
if (isset($_GET['prixMax'])){
    $prixMax = htmlspecialchars($_GET['prixMax']);
} else {
    $prixMax = '';
}
if (isset($_GET['tri'])){
    $optionTri = htmlspecialchars($_GET['tri']);
} else {
    $optionTri = '';
}
```

Cette partie permet de mettre dans des variables les différentes valeurs de recherches saisies par l'utilisateur lors de l'envoi du formulaire expliqué plus tard. Si l'utilisateur n'a pas rempli les champs, on laisse les variables vides. Dans le cas contraire, on ajoute la fonction `htmlspecialchars()` qui permet d'éviter les injections SQL.

```
$query = "SELECT DISTINCT P.NOMPRODUIT, P.IDPRODUIT, P.PRIXPRODUIT, P.DATEPRODUIT,
(P.PRIXPRODUIT - P.REDUCTION) as REDUC FROM produit P, categorie C, SOUSCATEGORIE S";
```

Cette variable contient la requête de base qui nous permet de récupérer tous les produits.

```
if ($recherche == ''){
    $titreRecherche = "Tous les produits :";
} else {
    $query .= " WHERE NOMPRODUIT = '$recherche.'";
    $titreRecherche = "Résultats de la recherche :";
}
```

Cette instruction conditionnelle permet d'ajouter à la requête un nom exact de produit à la requête si l'utilisateur a rempli le champ correspondant.

```
if ($categorieRecherchee != ''){
    if ($recherche != ""){
        $query .= " AND";
    }
    $query .= " WHERE C.IDCAT = '$categorieRecherchee.' AND P.IDSOUSCAT = S.IDSOUSCAT AND S.IDCAT = C.IDCAT";
    $titreRecherche = "Résultats de la recherche :";
}
```

Cette instruction conditionnelle permet d'ajouter à la requête une catégorie de produit à la requête si l'utilisateur a rempli le champ correspondant.

```
if ($sousCategorieRecherchee != ''){
    if ($recherche != "" or $categorieRecherchee != ""){
        $query .= " AND";
    } else {
        $query .= " WHERE";
    }
    $query .= " P.IDSOUSCAT = '$substr($categorieRecherchee, -2).$sousCategorieRecherchee.'";
    $titreRecherche = "Résultats de la recherche :";
}
```

Cette instruction conditionnelle permet d'ajouter à la requête une sous-catégorie de produit à la requête si l'utilisateur a rempli le champ correspondant.

```
if ($prixMin != ""){
    if ($recherche != "" or $categorieRecherchee != ""){
        $query .= " AND";
    } else {
        $query .= " WHERE";
    }
    $query .= " (P.PRIXPRODUIT - P.REDUCTION) > '$prixMin.'";
    $titreRecherche = "Résultats de la recherche :";
}
```

Cette instruction conditionnelle permet d'ajouter à la requête un prix minimum à ne pas dépasser à la requête si l'utilisateur a rempli le champ correspondant.

```

if ($prixMax != ''){
    if ($recherche != "" or $categorieRecherchee != "" or $prixMin != ""){
        $query .= " AND";
    } else {
        $query .= " WHERE";
    }
    $query .= " (P.PRIXPRODUIT - P.REDUCTION) <'".$prixMax."'";
    $titreRecherche = "Résultats de la recherche :";
}

```

Cette instruction conditionnelle permet d'ajouter à la requête un prix maximum à ne pas dépasser à la requête si l'utilisateur a rempli le champ correspondant.

```

if ($optionTri == 0){
    $query .= " ORDER BY (P.PRIXPRODUIT - P.REDUCTION) ASC";
} elseif ($optionTri == 1){
    $query .= " ORDER BY (P.PRIXPRODUIT - P.REDUCTION) DESC";
} elseif ($optionTri == 2){
    $query .= " ORDER BY P.NOMPRODUIT ASC";
} elseif ($optionTri == 3){
    $query .= " ORDER BY P.NOMPRODUIT DESC";
} elseif ($optionTri == 4){
    $query .= " ORDER BY P.DATEPRODUIT DESC";
}

```

Ces instructions conditionnelles permettent d'appliquer une fonction de tri des produits en fonction du choix de l'utilisateur. Les valeurs entre 0 et 4 comprises dans les instructions correspondent aux options de la liste déroulante. En fonction du choix, un ajout de code est réalisé à la requête afin que celle-ci retourne la liste des produits directement triés.

```

$stmtid = oci_parse($conn, $query);
oci_execute($stmtid);
while ($row = oci_fetch_array($stmtid, OCI_ASSOC)) {
    $res[] = ['nom' => $row['NOMPRODUIT'], 'id'=> $row['IDPRODUIT'], 'prix' => $row['PRIXPRODUIT'], 'reduc' => $row['REDUC']];
}

```

Cette partie du code exécute la requête dans la base de données et retourne les résultats sous la forme d'un tableau associatif, ce qui permet d'accéder aux valeurs plus facilement par la suite.

```
//categories
$query2 = "SELECT DISTINCT nomcat, IDCAT FROM categorie ORDER BY IDCAT";
$stmt2 = oci_parse($conn, $query2);
oci_execute($stmt2);

while ($row2 = oci_fetch_array($stmt2, OCI_ASSOC)) {
    $res2[] = $row2['NOMCAT'];
}

```

Le code ci-dessus récupère les catégories de produits directement dans la base de données puis les stocke dans un tableau.

```
// Sous catégories
$query3 = "SELECT DISTINCT IDSOUSCAT, NOMSOUSCAT, IDCAT FROM SOUSCATEGORIE";
$stmt3 = oci_parse($conn, $query3);
oci_execute($stmt3);

while ($row3 = oci_fetch_array($stmt3, OCI_ASSOC)){
    $res3[] = ['id' => $row3['IDSOUSCAT'], 'nom' => $row3['NOMSOUSCAT'], 'cat' => $row3['IDCAT']];
}

```

Le code ci-dessus récupère les sous-catégories de produits directement dans la base de données et les stocke dans un tableau.

```
<form action="Recherche.php" method="get" name="Recherche">
    <?php
        if($recherche==''){
            echo"<input type='text' name='recherche' placeholder='Nom du produit'>";
        } else {
            echo"<input type='text' name='recherche' value='".$recherche."' placeholder='Nom du produit'>";
        }
    </?php

```

Cette partie du code marque le début du formulaire de recherche. Pour ce champ, il permet de rentrer un nom de produit pour le rechercher dans la base de données. Si on a déjà recherché et qu'on envoie le formulaire, le nom recherché restera dans le champ. Dans le cas contraire, le champ est vide.

```

echo"<select name=\"categorieRecherchee\" class=\"select-style\" id=\"cat\">";
if($categorieRecherchee == ''){
    echo"<option value=\"\"selected>Catégorie</option>";
} else {
    echo"<option value=\"\">Catégorie</option>";
}
$i = 1;
foreach ($res2 as $categorie) {
    if($categorieRecherchee == $i){
        echo"<option value=\"0000\".strval($i).\"selected>\".$categorie.\"</option>";
    }
    echo"<option value=\"0000\".strval($i).\"\">\".$categorie.\"</option>";
    $i += 1;
}

```

Cette partie du code concerne le choix par catégorie. Comme vu précédemment, si une catégorie est recherchée, elle restera dans le formulaire. Les catégories sont récupérées dans la base de données puis affichées grâce à la boucle foreach.

```

if($sousCategorieRecherchee == ''){
    echo"<select name=\"sousCategorieRecherchee\" class=\"select-style\" id=\"sousCat\">
    <option value=\"\"selected>Sous Catégorie</option>
    <option value=\"00A\">SousCat1</option>
    <option value=\"00B\">SousCat2</option>
    </select>";
} elseif($sousCategorieRecherchee == '00A'){
    echo"<select name=\"sousCategorieRecherchee\" class=\"select-style\" id=\"sousCat\">
    <option value=\"\">Sous Catégorie</option>
    <option value=\"00A\"selected>SousCat1</option>
    <option value=\"00B\">SousCat2</option>
    </select>";
} elseif($sousCategorieRecherchee == '00B'){
    echo"<select name=\"sousCategorieRecherchee\" class=\"select-style\" id=\"sousCat\">
    <option value=\"\">Sous Catégorie</option>
    <option value=\"00A\">SousCat1</option>
    <option value=\"00B\"selected>SousCat2</option>
    </select>";
}

```

Le code ci-dessus permet de créer la liste déroulante pour les sous-catégories. Les noms "SousCat1" et "SousCat2" sont temporaires et seront remplacés en fonction de la catégorie sélectionnée en temps réel grâce à un script JavaScript.

```

if($prixMin == ''){
    echo"<input type=\"number\" name=\"prixMin\" placeholder=\"Prix minimum\">";
} else {
    echo"<input type=\"number\" name=\"prixMin\" placeholder=\"Prix minimum\" value=\".$prixMin.\">";
}
if($prixMax == ''){
    echo"<input type=\"number\" name=\"prixMax\" placeholder=\"Prix maximum\">";
} else {
    echo"<input type=\"number\" name=\"prixMax\" placeholder=\"Prix minimum\" value=\".$prixMax.\">";
}

```

Ces instructions conditionnelles gèrent les prix minimum et maximum. Comme pour les autres champs, si des valeurs ont déjà été saisies au préalable, elles resteront dans les champs.

```

if($optionTri == ''){
    echo"<select name=\"tri\" class=\"select-style\">
    <option value=\"\"selected>Options de tri</option>
    <option value=\"0\">Prix 🟡</option>
    <option value=\"1\">Prix dé 🟡</option>
    <option value=\"2\">Nom A-Z</option>
    <option value=\"3\">Nom Z-A</option>
    <option value=\"4\">Nouveautés</option>
    </select>";
}

```

Cette partie correspond à la liste déroulante des choix de tri.

```

echo"<script>
    document.getElementById(\"cat\").addEventListener(\"change\", function(){
        var nbCategories = ".$res3[0].count($res3)."/2;
        var catSelected = document.getElementById(\"cat\").selectedIndex;
        if(catSelected == 1 ){
            document.getElementById(\"sousCat\").options[1].innerHTML = \"\".$res3[0]['nom'].\"\";
            document.getElementById(\"sousCat\").options[2].innerHTML = \"\".$res3[1]['nom'].\"\";
        } else if(catSelected == 2 ){
            document.getElementById(\"sousCat\").options[1].innerHTML = \"\".$res3[2]['nom'].\"\";
            document.getElementById(\"sousCat\").options[2].innerHTML = \"\".$res3[3]['nom'].\"\";
        } else if(catSelected == 3 ){
            document.getElementById(\"sousCat\").options[1].innerHTML = \"\".$res3[4]['nom'].\"\";
            document.getElementById(\"sousCat\").options[2].innerHTML = \"\".$res3[5]['nom'].\"\";
        } else if(catSelected == 4 ){
            document.getElementById(\"sousCat\").options[1].innerHTML = \"\".$res3[6]['nom'].\"\";
            document.getElementById(\"sousCat\").options[2].innerHTML = \"\".$res3[7]['nom'].\"\";
        } else if(catSelected == 5 ){
            document.getElementById(\"sousCat\").options[1].innerHTML = \"\".$res3[8]['nom'].\"\";
            document.getElementById(\"sousCat\").options[2].innerHTML = \"\".$res3[9]['nom'].\"\";
        }
    })
</script>";

```

Ce script JavaScript permet d'afficher les sous-catégories récupérées dans la base de données et correspondantes à la catégorie sélectionnée en temps réel.

```
if (!is_null($res) or isset($res)) {
    echo"<div class='main-card'>
    <h2>".$titreRecherche."</h2>
    <div class='main-card-content'>";
    foreach($res as $produit) {
        echo" <div class='produit'>
        <div><a><strong>".$produit['nom']."</strong></a></div>
        <div class='image-produit-content'><img class='image-produit' src='./img/produits/".$produit['id']."_1.jpg\" alt='Image du produit'></div>";
        if($produit['reduc'] != $produit['prix']){
            echo"<div><a class='reduc'>".$produit['prix']." €</a></div>
            <div><a>".$produit['reduc']." €</a></div>";
        } else{
            echo"<div><a>".$produit['prix']." €</a></div>";
        }
        echo"<form name= 'aie' action='produit.php' method= 'get'>
        <div><button name= 'identifiantP' type= 'submit' value= ' ".$produit['id']."'>Acheter</button></div>
        </form>";
    }
    echo"</div>";
} else {
    echo"<div class='main-card'>
    <h2>Désolé nous n'avons pas de '$_GET['recherche']' en stock :(</h2>";
}
```

Cette partie du code gère l'affichage des produits sélectionnés dans la requête (image, nom, prix). Dans le cas où la recherche ne mène à aucun résultat, on affiche une boîte dans laquelle un message prévient l'utilisateur que le produit recherché n'est pas en stock.