

---

# COMPTE RENDU WEB SAÉ DEVAPP

---

Groupe 2B11

## Table des matières

1 – Page de connexion .....	2
2 – Page d'inscription .....	5
3 – Page des produits .....	9

## 1 – Page de connexion

Il est possible pour les utilisateurs de notre site de se connecter à un compte, ce qui leur permettra de gérer un panier ainsi que des commandes :



**Connexion**

**Nom d'utilisateur**

Entrez votre email

**Mot de passe**

Entrez le mot de passe

☐

Se souvenir de moi

LOGIN

Créer votre compte Royal bio

Avant tout chose, nous devons créer une session en utilisant la méthode **session\_start()**

```
<?php session_start(); ?>
<!DOCTYPE html>
<html lang="en">
<head>
```

Ensuite, nous vérifions si les cookies existent et sont définies. Si c'est le cas, nous déclarons deux variables `$login` et `$pwd` qui prennent comme valeurs, celles contenues dans le cookie.

```
<?php
    if(isset($_COOKIE['cookieIdent']) && isset($_COOKIE['cookieMdp'])){
        $login = $_COOKIE['cookieIdent'];
        $pwd = $_COOKIE['cookieMdp'];
    }else{
        $login = "";
        $pwd = "";
    }
?>
```

## Semaine 50

Pour permettre la connexion d'un utilisateur à un compte, nous mettons en place un formulaire contenant un champ dans lequel l'utilisateur entrera le nom du compte auquel il veut se connecter et un autre champ dans lequel il entrera le mot de passe de ce compte, à noter que ses champs seront remplis automatiquement si un cookie est présent :

```
<form action="TraitConnexion.php" method="POST">
  <h1>Connexion</h1>

  <label><b>Nom d'utilisateur</b></label>
  <input type="email" value="<?php echo($login); ?>" placeholder="Entrer votre email" name="email" required>

  <label><b>Mot de passe</b></label>
  <input type="password" value="<?php echo($pwd); ?>" placeholder="Entrer le mot de passe" name="password" required>
  <input type="checkbox" name="memo"/> Se souvenir de moi<br></br>
  <input type="submit" id='submit' value='LOGIN' >
  <br>

  <a href="/~SAESYS11/pages/inscription.php"><input type="button" class='creation' value='Cr  er votre compte Royal bio' ></a>
</form>
```

Ce formulaire sera trait   dans le fichier TraitConnexion.php.

Nous v  rifions dans un premier temps que les champs ont bien   t   remplis.

Par la suite nous incluons le fichier nous permettant de nous connecter    la base de donn  es :

```
if(isset($_POST['email']) && isset($_POST['password'])){
    // connexion    la base de donn  es
    require_once("../include/connect.inc.php");
    error_reporting(0);
```

Nous cr  ons par la suite deux variables prenant comme valeurs celle de l'email et du mot de passe en utilisant les m  thodes htmlspecialchars() ainsi que password\_hash() qui permettent respectivement de s  curiser notre site contre les attaques XSS ainsi que de hasher le mot de passe pour ne pas le stocker en clair. :

```
$username = htmlspecialchars($_POST['email']);
$password=password_hash(htmlspecialchars($_POST['password']), PASSWORD_DEFAULT);
```

Par la suite, nous pr  parons et ex  cutons une requ  te SQL qui recherche l'adresse mail et le mot de passe taper par l'utilisateur pr  c  demment dans la base de donn  e. Cela permet d'une part de v  rifier que l'utilisateur poss  de bien un compte    cette adresse mail et d'une autre part que le mot de passe correspond bien    cette adresse mail :

```

if($username != "" && $password != ""){

    $req2 = "SELECT MOTPASSECLIENT FROM clients WHERE MAILCLIENT = :MAILCLIENT";

    $lmdps = oci_parse($connect, $req2);

    oci_bind_by_name($lmdps, ":MAILCLIENT", $username);
    $result = oci_execute($lmdps);
    while($row=oci_fetch_array($lmdps)){

        $mdp = $row[0];
    }
}

```

Pour vérifier la validité du mot de passe renseigné par l'utilisateur, nous vérifions dans un premier temps que celui-ci n'est pas vide. Dans un second temps, nous utilisons la fonction `password_verify()` qui devrait renvoyer 1 si ils sont identiques.

Si la vérification est un succès, dans le cas où l'utilisateur aurait sélectionner le bouton « se souvenir de moi » nous créons 2 cookies qui seront utilisés respectivement pour le mot de passe et pour le mail :

```

// nom d'utilisateur et mot de passe correctes
if($mdp!=''){

    if (password_verify (($_POST['password']),$mdp)!=1){
        echo 'les mots de passe ne sont pas identiques';
    }else{
        echo 'les mots de passe sont identique';
        if(isset($_POST['memo'])){
            setcookie('cookieIdent', $_POST['email'], time()+60 );
            setcookie('cookieMdp', $_POST['password'], time()+60 );
        }
        session_start();
    }
}

```

Par la suite nous préparons et exécutons une requête permettant d'obtenir toutes les informations du compte que nous allons ranger dans des SUPER VARIABLES : `$_SESSION`. Nous les utiliserons plus tard dans future page pour le compte du client :

```
$req2 = "SELECT * FROM clients WHERE MAILCLIENT = :MAILCLIENT";

$lemdps = oci_parse($connect, $req2);
oci_bind_by_name($lemdps, ":MAILCLIENT", $username);
$result = oci_execute($lemdps);
while($row=oci_fetch_array($lemdps)){

    $_SESSION['emailC'] = $row['MAILCLIENT'];
    $_SESSION['passwordC'] = $row['MOTPASSECLIENT'];
    $_SESSION['nomC'] = $row['NOMCLIENT'];
    $_SESSION['prenomC'] = $row['PRENOMCLIENT'];
    $_SESSION['adressC'] = $row['ADRESSECLIENT'];
    $_SESSION['telC'] = $row['TELEPHONECLIENT'];

}
header ('location: ../index.php');
```

## 2 – Page d’inscription

Naturellement, si il est possible de se connecter à un compte sur le site, il est possible aussi pour les utilisateurs de créer leur compte si ce n’est pas déjà fait :

# Inscription

**Nom**

**Prénom**

**Adresse mail**

**Mot de passe**

Au moins 8 caractères, dont 1 majuscule, 1 minuscule et 1 chiffre. Parce qu'on sait jamais.

Cette action aura des conséquences

---

Dans ce formulaire de connexion plusieurs champs de saisies sont présent permettant respectivement à l'utilisateur d'enregistrer un nom, un prénom, une adresse mail ainsi qu'un mot de passe pour son compte :

## Semaine 50

```
<div class="connect">
  <form action="inscription.php" method="POST">
    <h1>Inscription</h1>

    <label><b>Nom</b></label>
    <input type="text" placeholder="Entrer le nom" name="nomClient" required>

    <label><b>Prénom</b></label>
    <input type="text" placeholder="Entrer le prénom" name="prenomClient" required>

    <label><b>Adresse mail</b></label>
    <input type="text" placeholder="Entrer l'adresse mail" name="mailClient" required>

    <label><b>Mot de passe</b></label>
    <input type="password" placeholder="Entrer le mot de passe" name="motPasseClient" required>

    <div id="mdp"><p>Au moins 8 caractères, dont 1 majuscule, 1 minuscule et 1 chiffre. Parce qu'on sait jamais.</p></div>

    <input type="submit" id="submit" value='Cette action aura des conséquences' >
    <br>
  </form>
</div>
```

Pour traiter ce formulaire d'inscription, dans un premier temps nous incluons le fichier permettant de se connecter à la base de donnée :

```
require_once("../include/connect.inc.php");
error_reporting(0);
```

Ensuite, nous vérifions que les différents champs ont été saisis. Si c'est le cas, nous préparons et exécutons une requête permettant de vérifier si un compte a déjà été créé avec ses informations :

```
if((isset($_POST['nomClient'])) && (isset($_POST['prenomClient'])) && (isset($_POST['mailClient'])) && (isset($_POST['motPasseClient']))) {
    $isValid=true;

    $req1 = "select MAILCLIENT from clients where MAILCLIENT=:MAILCLIENT ";

    $lemdps = oci_parse($connect, $req1);
    $MAILCLIENTT=$_POST['mailClient'];
    oci_bind_by_name($lemdps, ":MAILCLIENT", $MAILCLIENTT);

    $result = oci_execute($lemdps);

    while($row=oci_fetch_array($lemdps)){
        $mailverif = $row[0];
    }
}
```

Par la suite, nous mettons en place une REGEX permettant de vérifier que les saisies sont correctes. Ainsi, le nom et le prénom ne doivent contenir que des lettres (les accent et le tiret sont acceptés), l'adresse mail doit contenir un @ et le mot de passe doit au minimum contenir 8 caractères en sachant qu'il doit contenir au minimum : 1 majuscule, 1 minuscule ainsi qu'un caractère spécial pour plus de sécurité :

```

$regex = '/^[A-Z][\p{L}-]*$/';

if ((!preg_match($regex,$_POST['nomClient'])) || (!preg_match($regex,$_POST['prenomClient']))) {
    echo "<br>Nom ou prénom invalide";
    $isValid=false;
}

$pattern = '/^(?=.*[!@#%&*~])(?=.*[0-9])(?=.*[A-Z]).{8,20}$/';
if (!preg_match($pattern,$_POST['motPasseClient'])) {
    echo "<br>Mot de passe trop faible";
    $isValid=false;
}

if ((!preg_match('#[@]#i',$_POST['mailClient'])) || ($_POST['mailClient']==$mailverif)){
    echo "<br>Adresse mail incorrecte";
    $isValid=false;
}

```

Lorsque les vérifications ont été effectuées, nous préparons et exécutons une requête permettant de créer ce nouveau compte dans la base de données.

Dans la première partie de cette requête, nous écrivons la requête SQL. Nous devons donner des noms pour chaque valeur à insérer dans la requête. Nous avons donc créé le nombre de variables nécessaire pour insérer un client dans notre cas 7. Puis ensuite nous allons associer chaque valeur à insérer avec une variable. Puis nous exécutons la requête, faisons une redirection si la requête fonctionne. Nous faisons un commit pour enregistrer le nouveau compte de notre client :

```

if($isValid==true){
    $req2 = "SELECT IDCLIENT FROM clients order by idclient asc";

    $lemmps = oci_parse($connect, $req2);

    $result = oci_execute($lemmps);

    while($row=oci_fetch_array($lemmps)){
        $idClient = $row[0];
    }
    $idClient= $idClient+1;
    $req3=('INSERT INTO clients VALUES (:idClient, :motPasseClient, :nomClient, :prenomClient, :adresseClient, :telephoneClient, :mailClient)');

    $insertClient = oci_parse($connect, $req3);

    $motPasseClient = password_hash(htmlspecialchars($_POST['motPasseClient']),
    PASSWORD_DEFAULT);
    $nomClient = htmlspecialchars($_POST['nomClient']);
    $prenomClient = htmlspecialchars($_POST['prenomClient']);
    $adresseClient = "vide";
    $telephoneClient = "06";
    $mailClient = htmlspecialchars($_POST['mailClient']);
}

```



```

oci_bind_by_name($insertClient, ":idClient", $idClient);

oci_bind_by_name($insertClient, ":motPasseClient", $motPasseClient);
oci_bind_by_name($insertClient, ":nomClient", $nomClient);
oci_bind_by_name($insertClient, ":prenomClient", $prenomClient);
oci_bind_by_name($insertClient, ":adresseClient", $adresseClient);
oci_bind_by_name($insertClient, ":telephoneClient", $telephoneClient);
oci_bind_by_name($insertClient, ":mailClient", $mailClient);

$result = oci_execute($insertClient);

if (!$result) {
    $e = oci_error($insertClient); // on récupère l'exception liée au pb d'exécution de la requête (violation PK par exemple)
    print htmlentities($e['message'], ENT_QUOTES, 'UTF-8');
} else {
    header('location: ../index.php');
}

// vérifiez l'insertion dans SQL Developer avec cette requête : select * from joueur where nj > 299;
// on commit les modifs faites
oci_commit($connect);

// Libère toutes les ressources réservées par un résultat Oracle
oci_free_statement($insertJoueur);

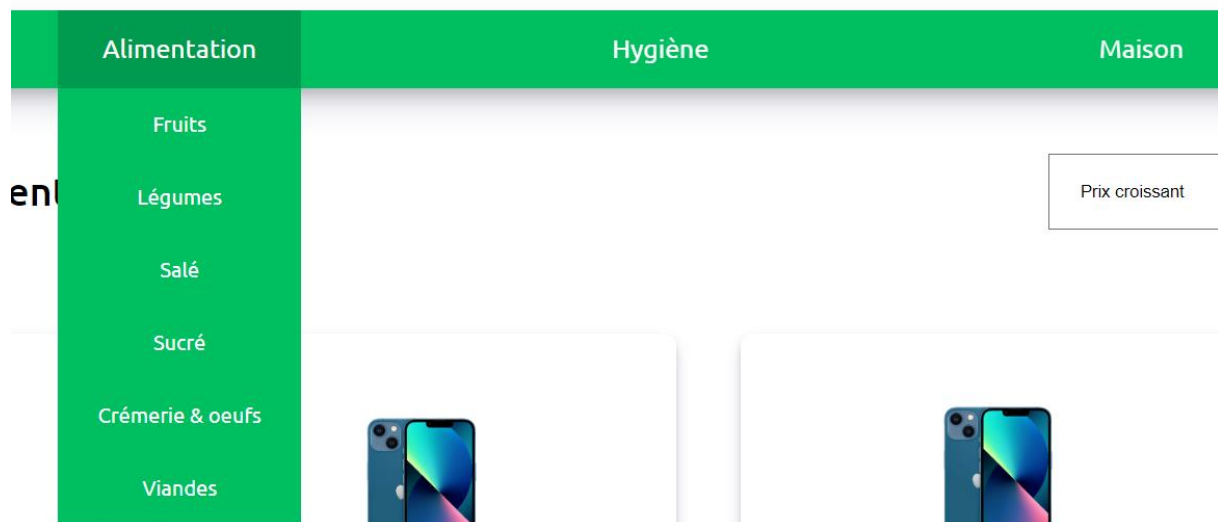
}

```

### 3 – Page des produits

Plusieurs catégories et sous catégories de produits sont disponibles sur le site que nous développons :

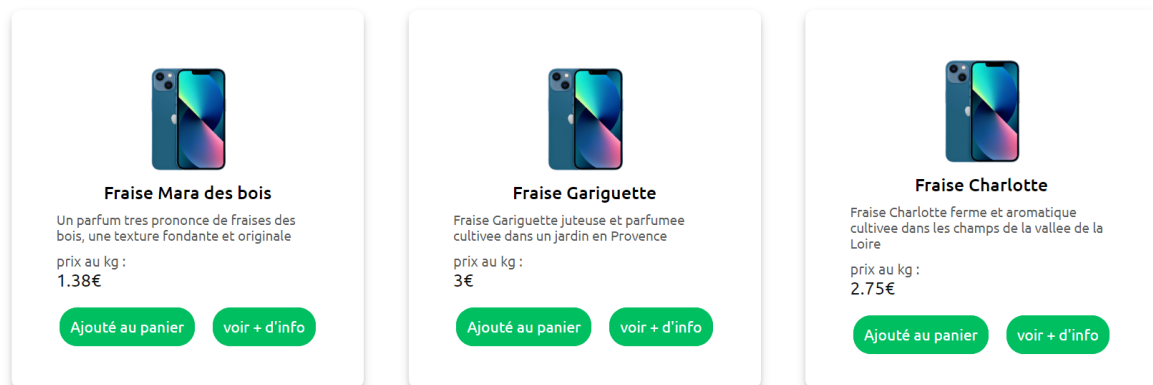
Figure 1 : affichage des catégories mères et certaines catégories filles



L'affichage des produits sera identique tandis que les informations seront naturellement différentes :

*Dans cette exemple nous affichons les produits contenues dans la catégorie des fruits (vous remarquerez que les images ne sont pas encore correctes : en cours de développement)*

Figure 2 : affichages des produits et leurs informations



Pour permettre l'affichage des produits ainsi que leurs informations, nous travaillons avec une base de donnée. Nous mettons donc en place des requêtes préparées.

Dans un premier temps, nous incluons le fichier permettant la connexion à la base de donnée pour faire des requêtes dans celle-ci :

```
<div class="product_container">
  <?php
    require_once("../include/connect.inc.php");
    error_reporting(0);
```

Ensuite, selon le lien que l'utilisateur aura sélectionné, nous envoyons des paramètres dans l'URL :

Figure 3 : code source du header.php

```
<div class="subheader">
  <ul class="menu">
    <li><a href="/~SAESYS11/">Accueil</a></li>
    <li><a href="/~SAESYS11/pages/produit.php?recherche=Aliments">Alimentation</a>
      <ul class="submenu">
        <li><a href="/~SAESYS11/pages/produit.php?recherche=Fruits">Fruits</a></li>
        <li><a href="/~SAESYS11/pages/produit.php?recherche=Légumes">Légumes</a></li>
        <li><a href="/~SAESYS11/pages/produit.php?recherche=Salé">Salé</a></li>
        <li><a href="/~SAESYS11/pages/produit.php?recherche=Sucré">Sucré</a></li>
        <li><a href="/~SAESYS11/pages/produit.php?recherche=Crèmerie">Crèmerie & oeufs</a></li>
        <li><a href="/~SAESYS11/pages/produit.php?recherche=Viandes">Viandes</a></li>
      </ul>
    </li>
    <li><a href="#">Hygiène</a>
      <ul class="submenu">
        <li><a href="/~SAESYS11/pages/produit.php">Savons</a></li>
        <li><a href="/~SAESYS11/pages/produit.php">Shampooings</a></li>
        <li><a href="/~SAESYS11/pages/produit.php">Soins & Visages</a></li>
      </ul>
    </li>
    <li><a href="/~SAESYS11/pages/produit.php">Maison</a>
      <ul class="submenu">
        <li><a href="/~SAESYS11/pages/produit.php">Soin du linge</a></li>
        <li><a href="/~SAESYS11/pages/produit.php">Vaisselle & Ménage</a></li>
        <li><a href="/~SAESYS11/pages/produit.php">Décoration</a></li>
      </ul>
    </li>
  </ul>
</div>
```

Cela permet, dans la page produit.php de vérifier quels liens l'utilisateur a sélectionné et donc mettre en place une requête différente selon son choix :

Figure 4 : 2 requêtes différentes en fonction du liens cliquer ou de la recherche effectuée

```
if($_GET['recherche'] == 'Aliments'){  
    $query = "  
    select *  
    from produit  
    where idcategorie = '01'  
        or idcategorie = '01'  
        or idcategorie = '02'  
        or idcategorie = '03'  
        or idcategorie = '04'  
        or idcategorie = '05'  
        or idcategorie = '06'  
        or idcategorie = '001'  
        or idcategorie = '002'  
        or idcategorie = '003'  
        or idcategorie = '004'  
        or idcategorie = '005'  
        or idcategorie = '006'";  
}  
else if($_GET['recherche'] == 'Fruits') {  
    $query = "  
        select *  
        from produit  
        where idcategorie = '01'  
            or idcategorie = '001'  
    ";  
}
```

Par la suite, nous préparons puis exécutons la requête :

```
$lesProduits = oci_parse($connect, $query);  
  
$result = oci_execute($lesProduits);
```

Et nous parcourons le curseur de la requête pour afficher les différents produits :

```
while (($leProduit = oci_fetch_assoc($lesProduits)) != false) {
    // echo 'Nom : ' . $leProduit['NOMPRODUIT'] . ' ; ' . ' ID : ' . $leProduit['IDPRODUIT'];
    echo '
        <div class="product_card">
            <div class="product_img">
                
            </div>
            <div class="product_title">'. $leProduit['NOMPRODUIT'] .'
```

*Il est à noter que l'affichage des produits par la recherche grâce à la barre de recherche présente dans le header.php est en cours de développement*