

Date : 18/12/2022

# Rapport - Web Sessions & cookies

---

Par FERNANDEZ Mickael, DOURLENT Maxime, STRAPUTICARI Luca et  
VIGNAL Alexandre



## Sommaire :

<b>Sommaire :</b>	<b>2</b>
Améliorations de la charte graphique	3
Connexion d'un utilisateur - aspect visuel	4
1. Formulaire de connexion	5
2. Fiche d'informations de l'utilisateur	6
Connexion d'un utilisateur - aspect fonctionnel	6
1. connexion.php	6
2. traitConnexion.php	9

## Améliorations de la charte graphique

Jusque là, aucun contenu présent dans le corps du site web n'était présent. C'est pourquoi faire en sorte que le bas de page soit fixe était une bonne idée.

Néanmoins, après clarification de notre client, il s'avère que leur demande ne correspondait pas exactement à ce que nous avons réalisé pour le bas de page. En effet, au lieu d'avoir une présence constante de celui-ci, il était préférable dans leur demande à ce que le bas de page soit visible uniquement lorsqu'on défilait le contenu d'une page.

Pour ce faire, des modifications ont été apportées au niveau du `html`, du `body` ainsi que du `footer` que voici :

```
html {  
  width:100%;  
  height:100%;  
  scroll-behavior:smooth;  
}  
  
body {  
  display:flex;  
  flex-direction: column;  
  
  margin:0;  
  
  min-height:100%;  
  width:100%;  
  font-family: 'Roboto', sans-serif;  
  background-image: url('./images/background-image.jpg');  
  background-repeat: no-repeat;  
  background-attachment: fixed;  
  background-size: cover;  
}
```

```

footer {
  display: flex;
  margin-top: auto;
  right: 0;
  bottom: 0;
  left: 0;
  justify-content: space-around;
  width: 100%;
  height: 120px;
  background-color: #162133;
}

```

Concrètement, le `html` prendra désormais 100% de la taille de la page en question, le `body` est retiré de sa propriété `"absolute"` pour être remplacé par des propriétés le rendant flexible.

Enfin, un `"margin-top: auto"` est ajouté à notre `footer`.

Pour finir, le logo d'Art'i'Chaude est désormais cliquable. Celui-ci permet de rediriger l'internaute sur la page d'accueil, présent sur l'ensemble des pages web du site.

## Connexion d'un utilisateur - aspect visuel

Sur le site web d'Art'i'Chaude, un simple internaute ne pourra jamais bénéficier de l'ensemble des fonctionnalités proposées par celui-ci. C'est pourquoi il lui est nécessaire de se connecter ou de s'inscrire, s'il ne possède pas de compte. Cette semaine, nous nous sommes focalisés sur l'identification d'un utilisateur, comme mentionné par notre prof de développement web.

Ainsi, dès lors qu'il clique sur l'image d'un client (voir ci-dessous), il tombe automatiquement sur un formulaire de connexion.

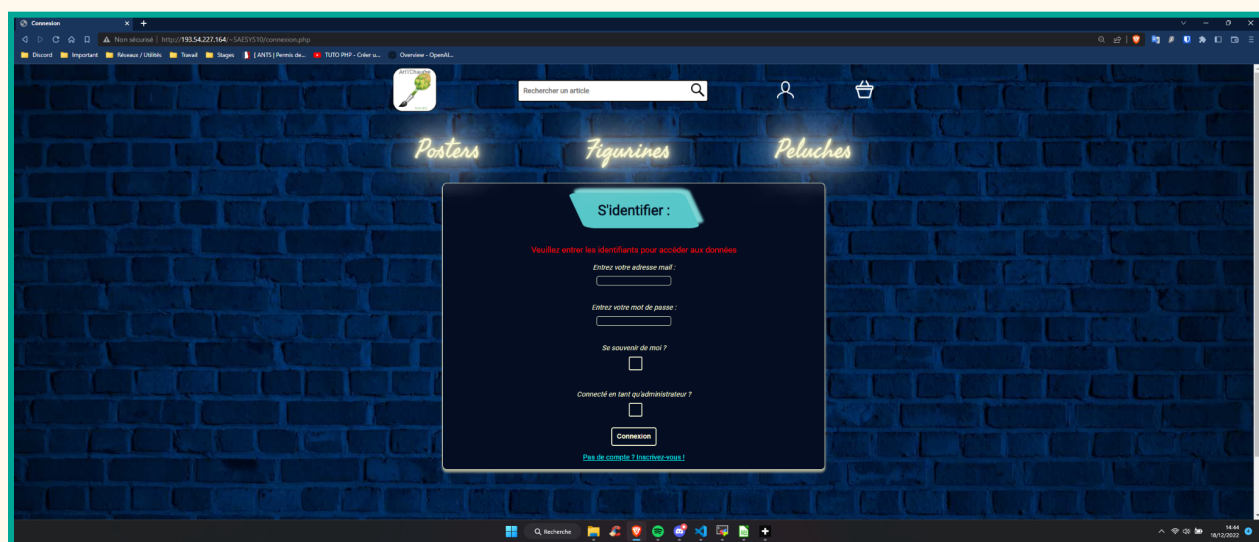


## 1. Formulaire de connexion

Cette page de formulaire de connexion sera, en définitive, la même page que la fiche d'informations de l'utilisateur. Tout dépendra si celui-ci sera connecté sur le site web ou non. Ce formulaire de connexion est composé de deux champs de texte : l'un, nécessitant une adresse mail, l'autre pour le mot de passe.

S'il le souhaite, il peut choisir qu'on se souvienne de lui : c'est un cookie. Dès lors qu'il décide de se connecter au site web, le premier champ de texte correspondant à son adresse mail sera automatiquement rempli.

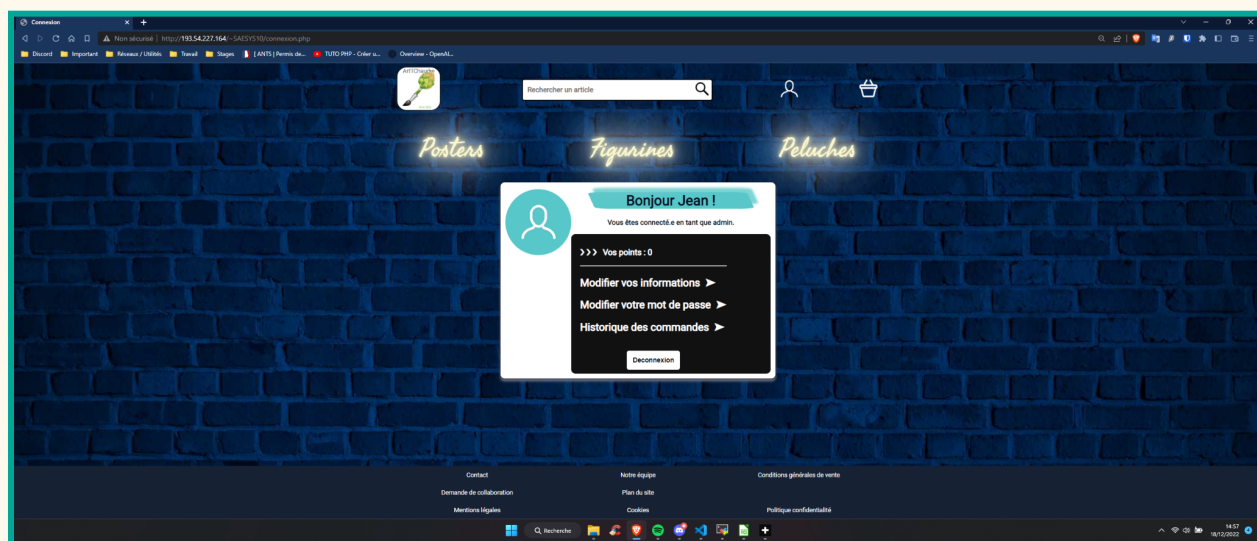
Deuxième possibilité, si l'utilisateur s'avère être un administrateur, il n'aura d'autres choix que de cliquer sur la *checkbox* en question. S'il s'avère qu'il ne soit pas un administrateur mais que pour des raisons malicieuses, il décide de cliquer sur le bouton, son accès sera tout de même rejeté. En réalité, une multitude de possibilités de contournement de connexion à ce formulaire sont possibles : non-remplissage des champs de texte (ou seulement l'un d'entre-eux), accès direct à la page de traitement de connexion etc. Tous ces cas sont traités et formulés par un message d'erreur en rouge.



## 2. Fiche d'informations de l'utilisateur

Une fois l'utilisateur connecté, il retombera automatiquement sur la page principale. S'il décide de cliquer à nouveau sur le client pour voir si tout se passe correctement, une fiche d'informations de celui-ci lui sera à sa disposition. Il a ainsi un ensemble de possibilités diverses mais également le fait de pouvoir se déconnecter.

S'il décide de se déconnecter, il retombera sur le formulaire de connexion :



## Connexion d'un utilisateur - aspect fonctionnel

Tout ce mécanisme de connexion d'un utilisateur n'est pas visible à l'œil nu : c'est un mécanisme qui se déroule en arrière-plan, selon ce que l'utilisateur réalise. Nous allons enfin évoquer l'utilisation du langage PHP pour permettre tout ce processus.

### 1. connexion.php

Le fichier `connexion.php` correspond en grande partie à l'aspect visible de la page web. Celui-ci va vérifier s'il existe une session (autrement dit, si l'utilisateur est connecté) et en fonction de son existence, l'affichage en sera également différent. Si l'utilisateur est alors connecté, c'est la page de connexion qui sera affichée et que voici :

```

if(isset($_SESSION['utilisateur'])) {
    echo('<div id="userConnected">');
    echo('<div id="gridContainer">');
        echo('<div>');
            echo('<div id="circle">');
                echo('');
            echo('</div>');
        echo('</div>');
    echo('<div id="userTitleFrame"><h1 id="userTitle">Bonjour ' . $_SESSION['prenom'] . ' !</h1></div>');
    echo('<p id="statusUser">Vous êtes connecté.e en tant que ' . $_SESSION['utilisateur'] . '</p>');

    echo('<div id="infosUser">');
        echo('<div id="paddingTexts">');
            echo('<h3>&#10095; &#10095; &#10095; &#8205; &#8205; Vos points : <strong>0</strong> </h3>');
            echo('<hr>');
            echo('<h2>Modifier vos informations &#8205; &#x27A4;</h2>');
            echo('<h2>Modifier votre mot de passe &#8205; &#x27A4;</h2>');
            echo('<h2>Historique des commandes &#8205; &#x27A4;</h2>');
            echo('<br/>');

            echo('<form method="POST" action="deconnexion.php">');
                echo('<input id="deconnexion" type="submit" name="Deconnexion" value="Deconnexion"/>');
            echo('</form>');

        echo('</div>');
    echo('</div>');
    echo('</div>');
    echo('</div>');
    echo('</div>');
}

```

Sinon, à contrario, le formulaire de connexion sera affiché et que voici :

```

} else {
    echo('<form method="post" id="formConnexion" action="traitConnexion.php">');

    echo('<div id="titleFrame"><h1 id="formTitle">S'identifier :</h1></div>');

    if(isset($_GET['msgErreur'])) {
        echo('<h3 class="errorDatabase">' . $_GET['msgErreur'] . '</h3>');
    } else {
        echo('<h3 class="errorDatabase">Veuillez entrer les identifiants pour accéder aux données</h3>');
    }

    echo('<label class="formLabels">Entrez votre adresse mail :</label>');
    echo('<input type="mail" class="champsForm" name="adresseMail" value="$login" required/>');
    echo('<br/><br/>');

    echo('<label class="formLabels">Entrez votre mot de passe :</label>');
    echo('<input type="password" class="champsForm" name="motDePasse" required/>');
    echo('<br/><br/>');

    echo('<label class="formLabels">Se souvenir de moi ?</label>');
    echo('<label id="checkboxLabel">');
        echo('<input type="checkbox" id="checkboxInput" name="memo"/>');
        echo('<svg id="checkboxCheck">');
            echo('<polyline points="20 6 9 17 4 12"></polyline>');
        echo('</svg>');
    echo('</label>');
    echo('<br/><br/>');

    echo('<label class="formLabels">Connecté en tant qu'administrateur ?</label>');
    echo('<label id="checkboxLabel">');
        echo('<input type="checkbox" id="checkboxInput" name="memoAdmin"/>');
        echo('<svg id="checkboxCheck">');
            echo('<polyline points="20 6 9 17 4 12"></polyline>');
        echo('</svg>');
    echo('</label>');
    echo('<br/>');

    echo('<input type="submit" name="Connexion" value="Connexion" id="btnConnexion"/>');

    echo('<a id="linkToInscription" href="#"><p>Pas de compte ? Inscrivez-vous !</p></a>');

    echo('</form>');
}

```

La présence de tous ces `echos` constituent le fond des pages webs.

Par ailleurs, au début de ce même fichier, la vérification de l'existence d'un cookie est à déclarer. Si c'est le cas, la variable `$login` en prendra son résultat et sera affiché dans le champ de texte de l'adresse mail.

```
session_start();

if(isset($_COOKIE['cookIdentifiant'])) {
    $login = $_COOKIE['cookIdentifiant'];
} else {
    $login = "";
}
```

## 2. `traitConnexion.php`

Ce fichier constitue ainsi tout le mécanisme de connexion, alliant propriétés PHP et interaction avec une base de données Oracle.

En réalité, le script de connexion pour un administrateur ou un client/artiste est très similaire. La différenciation se réalisera à partir du moment où l'internaute aura cliqué ou non sur la `checkbox` correspondant à son statut d'administrateur ou non.

Pour procéder à cette connexion, il est nécessaire, dans un premier temps, de joindre un fichier nommé `connect.inc.php` qui permet une connexion à notre base de données Oracle. Nous avons préféré privilégier dans un premier temps, de créer la base de données sur la session de l'un de nos membres.

Ainsi, à la différence d'utiliser une base de données via MySQL (constamment utilisée lors de nos TP en développement web), nous privilégierons cette fois-ci Oracle. Ainsi, les propriétés PHP pour interagir avec notre base de données seront différentes, bien que le fonctionnement s'y ressemble grandement.

En voici le fichier `connect.inc.php` :



```

1  <?php
2      $db = "(DESCRIPTION =
3          (ADDRESS = (PROTOCOL = TCP)(HOST = oracle.iut-blagnac.fr)(PORT = 1521))
4          (CONNECT_DATA =
5              (SERVER = DEDICATED)
6              (SID = db11g)
7          )
8      )";
9      $connect = oci_connect("IUTB2052", "videofest", $db);
10
11     // si la connexion a échoué, on affiche le message d'erreur
12     if (!$connect) {
13         $e = oci_error();
14         trigger_error(htmlentities($e['message'], ENT_QUOTES), E_USER_ERROR);
15     }
16     ?>

```

En vue de l'utilisation d'une base de données temporaire, nous laissons notre nom de session et mot de passe clair. Si la connexion échoue, l'erreur sera alors interceptée et affichée.

Pour revenir sur notre fichier traitant le formulaire, nous allons vérifier dans un premier temps, si les champs de texte ainsi que notre bouton ont été interagi.

De là, comme énoncé plus haut, nous allons vérifier si l'utilisateur a coché "Connecté en tant qu'administrateur" car deux procédures, similaires dans leur structure mais différente dans leur fonctionnement, se réaliseront.

Si l'utilisateur s'avère être en réalité un administrateur, nous allons réaliser une requête préparée avec notre base de données. Nous allons sélectionner l'ensemble des attributs nécessaires de notre table "Administrateur" où le mail correspond à un paramètre nommé. Ce paramètre nommé sera associé à la saisie de l'utilisateur dans le champ de texte de l'adresse mail.

Ainsi, une fois la requête SQL réalisée, nous allons la préparer à l'aide de notre base de données précédemment configurée pour cela.

A l'exécution de la requête, nous allons vérifier si une erreur existe. Si c'est le cas, l'exception sera captée et affichée. A contrario, nous allons renvoyer la ligne suivante à

partir des données de résultat et la stocker dans une variable nommée `$adminDatabase`.

S'il s'avère que cette variable est vide, nous allons rediriger l'utilisateur vers la même page avec l'affichage de l'erreur "Compte inexistant", stockée dans la variable `'msgErreur'` et récupérée dans l'URL par le biais de `$_GET['msgErreur']`.

Si ce n'est pas le cas, nous allons vérifier si le mot de passe clair, hashé par la suite et renseigné par l'utilisateur, correspond au même mot de passe hashé stocké dans la base de données. A partir de ce moment, il s'avère que notre internaute est un administrateur, et on viendra créer une session, récupérant l'ensemble des informations nécessaires à celui-ci comme son statut, son prénom ou encore son nom. Nous verrons plus tard si d'autres informations doivent également être récupérées.

Enfin, nous terminerons par vérifier si l'utilisateur souhaite à ce que l'on se souvienne de lui. Si c'est le cas, un cookie sera créé et autrement, il sera automatiquement détruit.

A la fin, nous libérons la ressource réservée pour la requête Oracle et nous renvoyons notre utilisateur sur la page d'accueil. Lorsqu'il reviendra sur la même page, ce traitement ne subsistera pas.

En voici le code PHP pour l'ensemble de ce traitement :

```

traitConnexion.php • deconnexion.php connexion.php # styles.css
C: > Users > micka > AppData > Roaming > MobaXterm > slash > RemoteFiles > 198244_2_1 > traitConnexion.php
1 <?php
2 // on inclut le fichier de connexion à la base Oracle
3 require_once("connect.inc.php");
4
5 session_start();
6
7 if(isset($_POST['Connexion'])) {
8
9     if(isset($_POST['adresseMail']) && isset($_POST['motDePasse'])) {
10
11         if(isset($_POST['memoAdmin'])) {
12
13             $req = "SELECT * FROM ADMINISTRATEUR WHERE MAILADMIN = :pMailAdmin";
14
15             //On prépare la requête
16             $adminRecup = oci_parse($connect, $req);
17
18             //On récupère les valeurs renseignées par l'utilisateur dans les champs de texte
19             $mailAdmin = $_POST['adresseMail'];
20             $mdpAdmin = $_POST['motDePasse'];
21
22             //On lie les valeurs aux paramètres de la requête
23             oci_bind_by_name($adminRecup, ":pMailAdmin", $mailAdmin);
24
25             //On exécute la requête
26             $result = oci_execute($adminRecup);
27
28             // si erreur de requete alors affichage...
29             if (!$result) {
30
31                 $e = oci_error($adminRecup); // on récupère l'exception liée au pb d'exécution de la requete (violation PK par ex
32                 print htmlentities($e['message'].' pour cette requete : '.$e['sqltext']);
33
34             } else {
35
36                 $adminDatabase = oci_fetch_assoc($adminRecup);
37
38                 if(empty($adminDatabase)){
39
40                     header('Location: ../connexion.php?msgErreur=Compte inexistant');
41                     exit;
42                 }

```

```

        if(password_verify($mdpAdmin, $adminDatabase['MDPADMIN'])) {

            $_SESSION['utilisateur'] = "admin";
            $_SESSION['prenom'] = $adminDatabase['PRENOMADMIN'];
            $_SESSION['nom'] = $adminDatabase['NOMADMIN'];

            if(isset($_POST['memo'])) {
                setcookie('cookIdentifiant', $_POST['adresseMail'], time()+60);
            } else {
                setcookie('cookIdentifiant', false, -1);
            }

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

            header('Location: ../index.php');
            exit;

        } else {

            header('Location: ../connexion.php?msgErreur=Mot de passe invalide');
            exit;

        }

    }
}

```


Ce procédé pour vérifier si l'internaute est un utilisateur/artiste est strictement similaire. La différence sera à notifier au niveau de la requête SQL ainsi qu'au nommage des variables, pour plus de cohérence :

```
//On crée une requete paramétrée  
$req = "SELECT * FROM ACTEUR WHERE MAILACTEUR = :pMailActeur";
```

```
$clientDatabase = oci_fetch_assoc($clientRecup);
```

### 3. deconnexion.php

Enfin, lorsque notre utilisateur souhaite se déconnecter, sa session sera automatiquement détruite au moment où il aura cliqué sur le bouton associé. De plus, il sera redirigé vers la page du formulaire de connexion.



```
traitConnexion.php ● deconnexion.php × connexion.php # styles.css  
C: > Users > micka > AppData > Roaming > MobaXterm > slash > RemoteFiles > 198244_6_18 > deconnexion.php  
1 <?php  
2  
3 session_start();  
4  
5 if(isset($_POST['Deconnexion'])) {  
6  
7 session_destroy();  
8 header('Location: ./connexion.php');  
9  
10 }  
11  
12 ?>
```