

AP2.2 : Projet Gestion des arbitres : Mission 7

Présentation de l'équipe et de la mission	
Présentation	Nous sommes deux élèves de BTS SIO SLAM, Christian et Mélanie, et, dans le cadre de notre deuxième AP, nous avons décidé de prendre la mission 7 qui concerne principalement les équipes de la ligue de basketball.
Tâches	Réalisation des tâches suivantes : Menu général, T3.1 et T2.3. Comme demandé dans le DokuWiki, nous avons développé ces tâches sous la forme d'architecture MVC (Modèle Vue Contrôleur).
Outils	Afin de réaliser ces tâches, nous avons utilisé le logiciel Visual Studio Code dans lequel nous avons utilisé les langages PHP, HTML et CSS.

T3.1 Gestion des équipes

Rappel des objectifs

Cette tâche doit permettre la création, la modification et la suppression d'une équipe.

Gestion des équipes

Nous commençons par créer une nouvelle **action** dans le **Contrôleur principal**.

```
$lesActions["equipes"] = "goequipes.php";
```

Comme nous le voyons ci-dessus, nous avons appelé cette **action** **equipes**.

Affichage du tableau

Avant la création d'une **équipe**, nous avons commencé par l'affichage du tableau **Gestion des Equipes** comprenant comme nous le voyons sur le **PDF**, le **numéro** et le **nom de l'équipe**, le **nom du club** et le **nom du championnat**.

Nous avons développé tout cela sous la forme **MVC**. De ce fait, nous avons récupéré les données correspondant dans la base de données via le **Modèle**. Nous pouvons rappeler que ce dernier est le noyau de l'application qui gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu'elles puissent ensuite être traitées par le **Contrôleur**.

```
function getEquipe()  
  
    {  
  
        try {  
  
            $cnx = connexionPDO();  
  
            $req = $cnx->prepare("SELECT num_equipe, nom_equipe, nom_club,  
nom_championnat from equipe
```

```

        inner join championnat on equipe.num_championnat =
championnat.num_championnat

        inner join club on equipe.num_club = club.num_club");

$req->execute();

$resultat = $req->fetchAll(PDO::FETCH_OBJ);

} catch (PDOException $e) {

    print "Erreur !: " . $e->getMessage();

    die();

}

return $resultat;

}

```

Pour cela nous avons fait une requête classique qui nous permet de récupérer les champs demandés à l'aide de deux jointures (équipe – club et équipe – championnat).

Puis nous avons créé le fichier `goEquipes` dans le `Contrôleur`. Rappelons que ce dernier correspond à la principale partie de l'architecture, car c'est lui qui va faire le lien entre l'utilisateur, le `Modèle` et la `Vue`.

```
$Equipe = getEquipe();
```

Avec cette commande que nous voyons ci-dessus, nous récupérerons dans la variable `$Equipe` le résultat de la requête `getEquipe`.

```

<table>

    <tr>

        <th colspan="5">GESTION DES EQUIPES</th>

    </tr>

    <tr>

        <td><B>Sélectionner</B></td>

        <td><B>Numéro de l'équipe</B></td>

        <td><B>Nom de l'équipe</B></td>

```

```

        <td><B>Nom du club</B></td>

        <td><B>Nom du Championnat</B></td>

    </tr>

    <form action="./?action=equipes" method="POST">

        <?php foreach ($Equipe as $UneEquipe) { ?>

            <tr>

                <td></td>

                <td><?php echo $UneEquipe->num_equipe; ?> </td>

                <td><?php echo $UneEquipe->nom_equipe; ?></td>

                <td><?php echo $UneEquipe->nom_club; ?></td>

                <td><?php echo $UneEquipe->nom_championnat; ?></td>

            </tr>

        <?php }

    ?>

</table>

```

La variable, dont nous avons parlé précédemment, va nous être utile dans la **Vue** pour le fichier **VueEquipe**. Il semble important de rappeler que la **Vue** correspond au composant graphique de l'interface qui permet de présenter les données du **Modèle** à l'utilisateur.

Puis, nous avons ajouté du **PHP** dans notre tableau **HTML** et grâce à la boucle **foreach** nous allons exécuter la fonction donnée sur chaque élément du tableau.

Une fois allié au **CSS**, le tableau se trouvant dans la **VueEquipe** est comme ci-dessous.

GESTION DES EQUIPES				
Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée
<input type="radio"/>	35	Confinement	Immobilité sportive	Génie au travail

Création d'une équipe

Pour la création d'une **équipe**, nous allons commencer, comme tout à l'heure, par la description du fichier du **Modèle**, **bd.equipes**, car nous y avons mis les fonctions utiles.

Nous souhaitons que l'utilisateur ajoute un **nom d'équipe**, choisisse son **club** et son **championnat**. Il nous faut donc trois requêtes : les deux premières vont permettre de récupérer les **noms des clubs** et les **noms des championnats** dans leurs tables respectives, et la dernière va concerner l'insertion.

```
function getClub()

{

    try {

        $cnx = connexionPDO();

        $req = $cnx->prepare("SELECT * from club");

        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;

}

function getChampionnat()

{

    try {

        $cnx = connexionPDO();

        $req = $cnx->prepare("SELECT * from championnat");

        $req->execute();
```

```

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;

}

```

Comme nous le voyons ci-dessus, ces deux requêtes récupèrent tous les éléments des tables **club** et **championnat**. Grâce à cela nous allons créer deux listes déroulantes où figurera leur nom respectif.

```

function insertEquipe($nomEquipe, $numClub, $numChamp)

{

    try {

        $cnx = connexionPDO();

        $req = $cnx->prepare("INSERT INTO equipe VALUES (NULL, :nomEquipe, :numClub, :numChampionnat)");

        $req->execute(array(

            'nomEquipe' => $nomEquipe,

            'numClub' => $numClub,

            'numChampionnat' => $numChamp

        ));

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}

```

Enfin, nous avons la requête `insertEquipe`. Nous remarquons que le **numéro de l'équipe** correspond à `NULL`. En effet, comme demandé dans le PDF, il est en `auto-incrémentation`. Cela veut dire que nous ne le prendrons pas en charge et qu'il se définira automatiquement.

Nous pouvons passer ensuite au `Contrôleur`.

```
$Club = getClub();

$Championnat = getChampionnat();

if (isset($_POST['valider'])) {

    insertEquipe($_POST['txtNomEquipe'], $_POST['listeclub'], $_POST['listechamp']);

}
```

Comme tout à l'heure, nous récupérerons le résultat des deux requêtes dans deux variables. Puis nous testons l'existence du `input valider`. S'il existe, nous appelons la requête dont nous avons parlé précédemment, `insertEquipe` et nous passons en paramètre le **nom de l'équipe**, le **numéro de club** et le **numéro de championnat** qui correspondent aux variables `$nomEquipe`, `$numClub` et `$numChamp` que nous avons passés en paramètre dans le fichier `bd.equipes` du `Modèle`.

Le script que nous avons au-dessous est celui de la `VueEquipes`.

```
<input type="submit" name="valider" value="AJOUTER"/></br>

<input type="hidden" name="numEquipe">

<label name="nomequipe">{Nom d'Equipe :</label>

<input type="text" name="txtNomEquipe"></br>

<label name="nomclub">{Nom du Club :</label>

<select id="liste" name="listeclub">

    <?php foreach ($Club as $UnClub) { ?>

        <option value="<?php echo $UnClub->num_club; ?>"><?php echo $UnClub->nom_club; ?></option>

    <?php } ?>

</select>

<label name="nomchampionnat" name="nomchampionnat">{Nom du Championnat :</label>
```

```
<select id="listechamp" name="listechamp">
```

```
<?php foreach ($Championnat as $UnChampionnat) { ?>
```

```
    <option value="<?php echo $UnChampionnat->num_championnat; ?>"><?php echo
    $UnChampionnat->nom_championnat; ?></option>
```

```
<?php } ?>
```

```
</select>
```

Après avoir créé un `input type submit` afin d'ajouter la nouvelle **équipe**, nous avons caché l'`input` de `numEquipe` grâce au `type hidden` car, puisqu'il est en `auto-incrément`, l'utilisateur ne l'ajoutera pas.

Ensuite, nous avons créé un `input type text` afin de rentrer le nom de la nouvelle **équipe**, puis deux listes déroulantes qui vont contenir les **noms des clubs** et des **championnats**. Comme précédemment, nous créons un `foreach` pour chaque `select`, afin que tous les noms s'affichent.

Enfin, dans nos balises `option`, nous rajoutons une `value` car c'est cette dernière qui est transmis dans le formulaire.

Après avoir y ajouter du `CSS`, nous pouvons tester l'insertion.

Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes Incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée

Nom d'Equipe :

Nom du Club :

Nom du Championnat :

AJOUTER
 MODIFIER
 SUPPRIMER

Comme le montre la capture ci-dessus, nous souhaitons ajouter l'**équipe Distanciel** avec pour **club**, **Immobilité sportive** et pour **championnat**, **La teinture foirée**.

GESTION DES EQUIPES				
Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes Incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée
<input type="radio"/>	37	Distanciel	Immobilité sportive	La teinture foirée

Cela a fonctionné 😊

Modification d'une équipe

La **modification** se fera sur la même page que l'ajout. De ce fait, nous n'allons pas créer d'autre **Contrôleur**, **Modèle**, **Vue** ni **Action** dans le **Contrôleur principal**.

Nous souhaitons que l'utilisateur sélectionne l'**équipe** qu'il veut modifier à l'aide d'un **input type radio** que nous avons ajouté dans le tableau.

```
<td><input value="<?php echo $UneEquipe->num_equipe; ?>" name="equipe" type="radio"></td>
```

Comme nous pouvons le voir ci-dessus, nous avons mis dans la **value** un **echo** du **foreach** avec pour paramètre le **numéro de l'équipe** car c'est ce dernier qui sera transmis lorsque l'utilisateur fera son choix.

Puis, lorsqu'il aura choisi l'**équipe** qu'il veut modifier, il devra valider son choix en cliquant sur le **submit** qui permettra d'afficher les données dans la zone de texte et les deux listes déroulantes. Pour cela, nous avons besoin de deux requêtes : une pour modifier et une autre pour afficher.

Nous allons commencer par l'affichage.

```
function getEquipeByNum($numEquipe)

{

    try {

        $cnx = connexionPDO();

        $req = $cnx->prepare("SELECT num_equipe, nom_equipe, equipe.num_club,
nom_club, equipe.num_championnat, nom_championnat from equipe

            inner join championnat on equipe.num_championnat =
championnat.num_championnat

            inner join club on equipe.num_club = club.num_club WHERE num_equipe =
:numEquipe");

        $req->bindValue(':numEquipe', $numEquipe, PDO::PARAM_INT);

        $req->execute();

        $resultat = $req->fetch(PDO::FETCH_ASSOC);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}
```



```

        return $resultat;

    }

```

Cette requête est partiellement identique à celle concernant l'affichage du tableau `getEquipe`. En effet, nous avons rajouté un `Where` car nous souhaitons afficher l'**équipe**, le **club** et le **championnat** en fonction de la clé primaire `numEquipe`. Comme elle contient un paramètre, nous rajoutons la variable `$numEquipe`, à côté du nom de la fonction que nous déclarons.

Nous devons maintenant gérer cette requête dans le `Contrôleur`.

```

if (isset($_POST['afficher'])) {

    $numEquipe = $_POST['equipe'];

    $afficher = getEquipeByNum($numEquipe);

}

```

Nous faisons de nouveau un `if isset` qui nous permet de tester l'existence du `submit afficher`. A l'intérieur, nous déclarons la variable `$numEquipe` et nous passons en paramètre l'`input type radio`, dont nous avons parlé précédemment, et, qui se situe dans la `Vue`. Ensuite, nous redéclarons une autre variable `$afficher`, afin de récupérer le résultat de la requête.

Nous pouvons passer à la `Vue`.

```

<input type="hidden" name="numEquipe">

    <label name="nomequipe">{Nom d'Equipe :</label>

    <input type="text" name="txtNomEquipe" value="<?php if (isset($afficher)) { echo
    $afficher['nom_equipe']; } else { ?> <?php } ?>"></br>

    <label name="nomclub">{Nom du Club :</label>

    <select id="liste" name="listeclub">

        <option value="<?php if (isset($afficher)) { echo $afficher['num_club'];
        ?>"><?php echo $afficher['nom_club']; } else { ?><?php } ?></option>

        <?php foreach ($Club as $UnClub) { ?>

            <option value="<?php echo $UnClub->num_club; ?>"><?php echo $UnClub-
            >nom_club; ?></option>

            <?php } ?>

    </select>

```

```

<label name="nomchampionnat" name="nomchampionnat">{Nom du Championnat :</label>

<select id="listechamp" name="listechamp">

    <option value="<?php if (isset($afficher)) { echo $afficher['num_championnat'];
?>"><?php echo $afficher['nom_championnat']; } else { ?><?php } ?></option>

    <?php foreach ($Championnat as $UnChampionnat) { ?>

        <option value="<?php echo $UnChampionnat->num_championnat; ?>"><?php echo
$UnChampionnat->nom_championnat; ?></option>

    <?php } ?>

</select>

```

Comme nous faisons la **modification** dans la même page, nous avons décidé de faire l'affichage dans ce que nous avons créé pour l'**ajout**. Dans la zone de texte, nous avons rajouté une **value** avec à l'intérieur un **if** qui nous permet de faire : s'il y a quelque chose dans la variable **\$afficher**, on affiche à l'aide du **echo** son contenu.

Nous avons également fait cela dans les deux listes déroulantes.

Nous pouvons à présent passer à la modification.

```

function updateEquipe($numEquipe, $nomEquipe, $numClub, $numChampionnat)

{

    try {

        $cnx = connexionPDO();

        $req = $cnx->prepare("UPDATE equipe

                                SET nom_equipe = ?, num_club = ?, num_championnat = ?

                                WHERE num_equipe = ?");

        $req->bindValue(1, $nomEquipe);

        $req->bindValue(2, $numClub);

        $req->bindValue(3, $numChampionnat);

        $req->bindValue(4, $numEquipe);

        $req->execute();
    }
}

```

```

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}

```

Ci-dessus, se trouve la requête `updateEquipe` qui possède quatre paramètres. Grâce à elle nous allons modifier le **nom de l'équipe** et/ou le **numéro du club** et/ou du **championnat**.

```

if (isset($_POST['modifier'])) {

    $numEquipe = $_POST['numEquipe'];

    $nomEquipe = $_POST['txtNomEquipe'];

    $numClub = $_POST['listeclub'];

    $numChampionnat = $_POST['listechamp'];

    updateEquipe($numEquipe, $nomEquipe, $numClub, $numChampionnat);

    header('Location: ?action=equipes');

}

```

Comme pour l'affichage, nous testons l'existence du `input type submit` et, nous passons en paramètres les variables qui nous avons créé dans la fonction `updateEquipe`. Enfin, l'`option header` va nous permettre de recharger la page après avoir cliqué sur ce `submit` et de voir automatiquement le résultat de la **modification**.

```

<input type="submit" name="modifier" value="MODIFIER" /></br>

```

Pour la `Vue`, il nous faut seulement rajouter l'`input` pour modifier.

Après y avoir ajouter du `CSS`, nous pouvons tester la **modification** et l'**affichage**.

GESTION DES EQUIPES				
Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée
<input type="radio"/>	39	Distanciel	Immobilité sportive	La teinture foirée

Nom d'Equipe :

Nom du Club :

Nom du Championnat :

Afficher la sélection

AJOUTER
MODIFIER
SUPPRIMER

Comme le montre la capture ci-dessus, nous souhaitons modifier l'équipe **Distanciel**.

Nom d'Equipe :

Nom du Club :

Nom du Championnat :

En effet, nous voulons modifier son **nom d'équipe** afin qu'elle s'appelle **Présentiel**. Nous voulons aussi modifier le **championnat** (**La teinture foirée** remplacé par **Génie au travail**).

GESTION DES EQUIPES				
Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée
<input type="radio"/>	37	Présentiel	Immobilité sportive	Génie au travail

Comme le montre la capture du tableau ci-dessus, la **modification** a fonctionné 😊

Suppression d'une équipe

Pour la **suppression**, nous avons réutilisé l'affichage que nous avons fait précédemment, afin que lorsque l'utilisateur souhaite supprimer une **équipe**, il le peut, mais après contrôle des données.

Pour supprimer nous avons besoin d'une requête, **Delete** et cette dernière se place dans le **Modèle**.

```
function deleteEquipe($numEquipe)
{
    try {
```

```

        $cnx = connexionPDO();

        $req = $cnx->prepare("DELETE FROM equipe

                                WHERE num_equipe = ?");

        $req->bindValue(1, $numEquipe);

        $req->execute();

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}

```

Comme la **suppression** d'une équipe comprend tous les champs de la table **Équipe** nous mettons dans le **Where** seulement, la clé primaire **num_equipe**.

```

if(isset($_POST['supprimer'])){

    $numEquipe = $_POST['numEquipe'];

    deleteEquipe($numEquipe);

    header('Location: ?action=equipes');

}

```

Puis, comme pour l'**ajout** et la **modification**, nous testons l'existence du **input supprimer**, nous passons en paramètre la variable **\$numEquipe** et nous appelons le **header Location**.

```

<input type="submit" name="supprimer" value="SUPPRIMER" />

```

Dans la **Vue**, nous rajoutons l'**input type submit** et nous le nommons **Supprimer**.

Nous pouvons enfin tester la **suppression**.

GESTION DES EQUIPES

Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée
<input checked="" type="radio"/>	37	Présentiel	Immobilité sportive	Génie au travail

Nom d'Equipe :

Présentiel

Nom du Club :

Immobilité sportive

Nom du Championnat :

Génie au travail

Afficher la sélection

AJOUTER

MODIFIER

SUPPRIMER

Nous voulons supprimer l'équipe **Présentiel**.

Sélectionner	Numéro de l'équipe	Nom de l'équipe	Nom du club	Nom du Championnat
<input type="radio"/>	1	Les reines de Valadon	Elan Sportif	Génie au travail
<input type="radio"/>	3	Les artistes incompris	Les arts quoi	Très bonne question
<input type="radio"/>	4	Les Rookies	Roux Sot	La teinture foirée

Nous pouvons voir grâce à la capture du tableau que la **suppression** a fonctionné 😊

T2.3 Édition de la liste des arbitres

Rappel des objectifs

Le responsable veut pouvoir visualiser la liste des arbitres avec les informations le concernant (état-civil, adresse, numéros de téléphone, adresse mail, nom du club auquel il appartient éventuellement).

Édition de la liste des arbitres

Afin que le **responsable** puisse générer les informations des **arbitres** au format **PDF**, nous avons téléchargé la classe **FPDF** car elle va nous permettre de générer un fichier **PDF** en pur **PHP**.

Une fois téléchargé, dézipper puis placer dans un dossier **PDF**, nous avons créé à l'intérieur un fichier nommé **pdf**.

```
<?php
```

```

require("fpdf/fpdf.php");

class PDF extends FPDF {

    function Header() {

        $this->Ln(20);

        $this->SetFont('Helvetica','B',20);

        $this->setFillColor(230,230,230);

        $this->SetX(45);

        $this->Cell(120,20,'LISTE DES ARBITRES',0,1,'C',1);

        $this->Ln(10);

    }

}

```

Tout d'abord, nous avons inclus le fichier correspond à la classe **FPDF** puis nous avons créé la fonction **header**. Cette dernière nous permet de gérer la **police**, la **position** et le **cadre du titre** ainsi que sa **couleur**.

```

$pdf = new PDF('P','mm','A4');

$pdf->AddPage();

$pdf->SetFont('Helvetica','',9);

```

```
$pdf->SetTextColor(0);
```

```
$pdf->AliasNbPages();
```

Ensuite avec **new PDF** nous créons notre première page **PDF** et nous décidons de la **police** et de la **couleur** du tableau.

```
function entete_table($position_entete) {  
  
    global $pdf;  
  
    $pdf->SetDrawColor(183);  
  
    $pdf->SetFillColor(221);  
  
    $pdf->SetTextColor(0);  
  
    $pdf->SetY($position_entete);  
  
  
    $pdf->SetX(5);  
  
    $pdf->Cell(60,8,utf8_decode('État civil'),1,0,'C',1);  
  
  
    $pdf->SetX(65);  
  
    $pdf->Cell(70,8,'Adresse',1,0,'C',1);  
  
  
    $pdf->SetX(125);  
  
    $pdf->Cell(25,8,'Telephone',1,0,'C',1);  
  
  
    $pdf->SetX(150);  
  
    $pdf->Cell(30,8,'Mail',1,0,'C',1);  
  
  
    $pdf->SetX(180);
```



```
$pdf->Cell(25,8,'Nom du club',1,0,'C',1);

$pdf->Ln();

}
```

Comme le montre le code ci-dessus, nous déclarons à nouveau une fonction qui nous permettra d'y inclure le nom des entêtes du tableau. Nous pouvons voir qu'avant d'inscrire le nom des colonnes, nous incluons les paramètres concernant le design du tableau dont nous avons parlé précédemment.

Nous passons enfin au remplissage du tableau.

Ne sachant pas si nous devions faire la requête (récupérer toutes les données des arbitres) dans ce fichier ou conserver le modèle MVC, nous avons choisi la deuxième option.

De ce fait, dans le **Contrôleur principal**, nous avons rajouté une action concernant le **PDF**.

```
$lesActions["pdf"] = "pdf.php";
```

Puis dans le **Contrôleur** nous avons créé le fichier **pdf**.

```
<?php

include "modele/bd.pdf.php";

$Arbitre = getArbitre();

include "pdf/pdf.php";

?>
```

À l'intérieur, nous avons inclus le **bd.pdf** du **Modèle** dont nous parlerons plus tard. Nous avons aussi récupéré le résultat de la requête **getArbitre** dans la variable **\$Arbitre**. Enfin, nous incluons le fichier **pdf.php**.

```
function getArbitre()

{

    try {

        $cnx = connexionPDO();
```

```

        $req = $cnx->prepare("SELECT nom_arbitre, prenom_arbitre, date_naiss_arbitre,
adresse_arbitre, CP_arbitre, ville_arbitre, tel_fixe_arbitre, tel_port_arbitre, mailU,
nom_club from arbitre inner join club on arbitre.num_club = club.num_club");

        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;

}

```

À l'intérieur du fichier `bd.pdf`, il n'y a qu'une requête, `getArbitre`. Elle nous permet de récupérer l'**état-civil**, l'**adresse**, le **téléphone** et le **mail de l'arbitre**. Nous y avons aussi ajouté le **nom du club** auquel il appartient.

Revenons à présent au fichier `pdf`.

```

$position_entete = 70;

$pdf->SetFont('Helvetica','',9);

$pdf->SetTextColor(0);

$position_detail = 78;

foreach ($Arbitre as $UnArbitre) {

    $pdf->SetY($position_detail);

    $pdf->SetX(5);

    $pdf->MultiCell(20,8,$UnArbitre->nom_arbitre, 1,'C');

    $pdf->SetY($position_detail);

    $pdf->SetX(25);

    $pdf->MultiCell(20,8, utf8_decode($UnArbitre->prenom_arbitre), 1,'C');
}

```

```

$pdf->SetY($position_detail);

$pdf->SetX(45);

$pdf->MultiCell(20,8,$UnArbitre->date_naiss_arbitre, 1, 'C');

$pdf->SetY($position_detail);

$pdf->SetX(65);

$pdf->MultiCell(20,8, $UnArbitre->adresse_arbitre,1, 'C');

$pdf->SetY($position_detail);

$pdf->SetX(85);

$pdf->MultiCell(20,8,$UnArbitre->ville_arbitre,1, 'C');

$pdf->SetY($position_detail);

$pdf->SetX(105);

$pdf->MultiCell(20,8,$UnArbitre->CP_arbitre,1, 'C');

$pdf->SetY($position_detail);

$pdf->SetX(125);

$pdf->MultiCell(25,8,$UnArbitre->tel_port_arbitre,1, 'C');

$pdf->SetY($position_detail);

$pdf->SetX(150);

$pdf->MultiCell(30,8,$UnArbitre->mailU,1, 'C');

$pdf->SetY($position_detail);

$pdf->SetX(180);

$pdf->MultiCell(25,8,$UnArbitre->nom_club,1, 'C');

}

```

À l'aide de la boucle `foreach` nous faisons le lien avec la requête dont nous avons parlé précédemment.

Enfin, nous remplissons chacune des colonnes avec le nom des champs de la base de données, relié à la variable que nous créons dans le `foreach`.

Nous n'avons plus qu'à rajouter un lien dans le profil du **responsable**.

```
<a href="./?action=pdf"></a>
```



Une fois qu'il aura cliqué sur le lien, il verra apparaître le PDF qui se trouve ci-dessous.

LISTE DES ARBITRES

État civil			Adresse			Telephone	Mail	Nom du club
Bourgeois	Agnès	1976-01-01	Valadon	Limoges	87000	06-06-06-06-06	agnes@agnes.fr	Elan Sportif
Bogusz	Thierry	1976-02-02	Valadon	Limoges	87000	06-06-06-06-06	thierry@thierry.fr	Les arts quoi
Pasqualini	Claude	1962-03-03	Valadon	Saint-Junien	87200	06-06-06-06-06	claud@claud.fr	Elan Sportif
Techer	Charles	1976-04-04	Valadon	Limoges	87000	06-06-06-06-06	charles@charles.fr	Les arts quoi
Tournier	Alain	1976-05-05	Valadon	Limoges	87000	06-06-06-06-06	alain@alain.fr	Roux Sot

Ainsi, il verra un tableau répertoriant les informations des **arbitres**, qui se mettra automatiquement à jour en fonction d'un **ajout** et d'une **suppression** d'un **arbitre**.

Menu Général

Rappel des objectifs

Le responsable de la gestion des arbitres doit pouvoir s'authentifier grâce à une fenêtre de connexion. Ensuite, il aura accès à son menu qui comprendra les domaines D1, D2, D3. Un arbitre doit également s'identifier grâce à la même fenêtre de connexion. Il aura ensuite accès à son menu décrivant le domaine D4.

Menu General

Comme indiqué dans les objectifs, deux types de personnes doivent se connecter (**responsable** et **arbitre**). C'est pourquoi, nous avons créés deux espaces de **connexion** ainsi que des fichiers pour chaque catégorie.

Nous avons développé cela sous la forme **MVC**. De ce fait, nous allons commencer par la description des fichiers du **Modèle**.

Pour la **connexion** ainsi que la **déconnexion**, nous nous sommes inspiré des fichiers concernant le **projet resto-ressources** que nous développons en **Bloc 1**.

Nous avons, comme d'habitude la fonction **login**, qui nous permet de nous connecter et, la fonction **logout** qui nous permet de nous déconnecter. Mais il y a aussi les fonctions **getMailULoggedIn** et **isLoggedIn**.

```
function getMailULoggedInOnArbitre()
```

```
{
```

```
    if (isLoggedInOnArbitre()) {
```

```
        $return = $_SESSION["mail_arbitre"];
```

```
    } else {
```

```
        $return = "";
```

```
    }
```

```
    return $return ;
```

```
}
```

```
function isLoggedInOnArbitre()
```

```
{
```

```
    if (!isset($_SESSION)) {
```

```
        session_start();
```

```
    }
```

```
    $return = false;
```

```
    if (isset($_SESSION["mail_arbitre"])) {
```

```
        $util = getArbitreByMailU($_SESSION["mail_arbitre"]);
```

```
        if (
```

```
            $util["mail_arbitre"] == $_SESSION["mail_arbitre"] && $util["mdpU"] ==  
$_SESSION["mdpU"]
```

```
        ) {
```

```
            $return = true;
```

```
        }
```

```
    }
```

```
    return $return;
```

La première ne fonctionne que si la seconde, qui est de type **booléen**, (permet de vérifier si l'utilisateur a bien rentré son **email** et son **mot de passe**) retourne **true**.

Ces fonctions ont été développées pour l'**arbitre** comme nous le voyons ci-dessus, mais aussi pour le **responsable**.

Nous passons ensuite au fichier du **Contrôleur**.

```
$lesActions["default"] = "hello.php";

$lesActions["inscriptionA"] = "inscriptionArbitre.php";

$lesActions["connexionA"] = "connexionArbitre.php";

$lesActions["deconnexionA"] = "deconnexionArbitre.php";

$lesActions["inscriptionR"] = "inscriptionResponsable.php";

$lesActions["connexionR"] = "connexionResponsable.php";

$lesActions["deconnexionR"] = "deconnexionResponsable.php";

$lesActions["**profil**A"] = "**profil**Arbitre.php";

$lesActions["**profil**R"] = "**profil**Responsable.php";
```

Comme nous nous occupons de l'**inscription**, la **connexion**, la **déconnexion** et le **profil** des **arbitres** et des **responsables** nous avons dû créer plusieurs actions dans le **Contrôleur principal**.

```
<?php

include_once "modele/authentication.inc.php";

if (!isset($_POST["mailU"]) || !isset($_POST["mdpU"])) {

    $titre = "authentication";

    include "Vue/entete.html.php";

    include "Vue/VueAuthenticationArbitre.php";

    include "Vue/pied.html.php";

} else {

    $mailU = $_POST["mailU"];

    $mdpU = $_POST["mdpU"];

    loginArbitre($mailU, $mdpU);
```

```

if (isLoggedInOnArbitre()) {

    $titre = "confirmation";

    include "Vue/VueHello.php";

} else {

    $titre = "authentification";

    include "Vue/entete.html.php";

    include "Vue/VueAuthentificationArbitre.php";

    include "Vue/pied.html.php";

}

}

```

Tout d'abord, nous incluons les fichiers du **Modèle** qui sont utiles. Puis, comme on peut le voir ci-dessus, nous avons tester l'éventualité que l'utilisateur n'ait pas bien entré son **email** ou son **mot de passe**. Dans ce cas, l'**arbitre** ou le **responsable** est redirigé sur la **Vue d'authentification**. Mais dans le cas inverse, où l'**email** ou le **mot de passe** sont exacts, la **connexion** peut aboutir et ils arrivent sur l'**accueil** avec leur **menu**.

Enfin, nous pouvons passer à la **Vue**.

```

<form action="./?action=connexionA" method="POST">

    <input name="mailU" type="**email**" placeholder="Entrer l'**email**"
required>

    </br></br>

    <input name="mdpU" type="password" placeholder="Entrer le **mot de
passe**" required>

    </br></br>

    <input type="submit" name="connection" value="Se connecter" />

</form>

```

Le fichier d'authentification dans la **Vue** est un formulaire classique qui comprend uniquement de l'**HTML**.

Une fois alliée au **CSS** il ressemble à cela.

CONNEXION ARBITRE

Se connecter

Cliquez ici pour vous [inscrire](#)

Même si ce n'était pas indiqué dans la consigne, nous avons également géré l'**inscription** pour l'**arbitre** et le **responsable**.

```
$Equipe = getEquipe();

$Club = getClub();

if (isset($_POST['inscription'])) {

    $pass = $_POST['motdepasseA'];

    $hash = password_hash($pass, PASSWORD_BCRYPT);

    insertArbitre(

        $_POST['nomArbitre'],

        $_POST['prenomArbitre'],

        $_POST['adrArbitre'],

        $_POST['cpArbitre'],

        $_POST['villeArbitre'],

        $_POST['datenaissArbitre'],

        $_POST['telfixeArbitre'],

        $_POST['telportArbitre'],

        $_POST['mail_arbitre'],
```



```

$_POST['listeClub'],

$_POST['listeEquipe'],

$hash

);

header('Location: ?action=connexionA');

}

```

Comme nous le voyons ci-dessus, il s'agit de l'exemple de l'**inscription** d'un **arbitre**.

Nous récupérons tout d'abord, grâce à deux variables les requêtes **getEquipe** et **getClub**. Puis, nous testons l'existence du bouton **inscription** que nous avons passé en **POST**. Nous avons décidé de rajouter deux champs utiles à l'**inscription** dans la table **arbitre** : l'**email** et le **mot de passe**. Avant de faire l'**insertion** nous hashons le **mot de passe**. Enfin, nous faisons un **header Location** afin que l'utilisateur soit redirigé sur la **connexion**, dès qu'il s'est bien inscrit.

Nous avons également fait cela pour le **responsable**. Cependant, ce dernier n'existant pas dans l'annexe 1 du **PDF**, il a dû être rajouté à notre base de données, en y insérant seulement les champs utiles (**id**, **email** et **mot de passe**).

Base de données: basketball » Table: responsable

✓ Affichage des lignes 0 - 1 (total de 2, traité en 0,0000 seconde(s))

SELECT * FROM `responsable`

Profilage [Éditer]

Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier sur l'index: Aucun(e)

	id_responsable	mailU	mdpU
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	7	melanie@melanie.fr	\$2y\$10\$ftP3tA473KEJhEynp.wRWenOKeQRgTTtwnBECJYOko3...
<input type="checkbox"/> Éditer <input type="checkbox"/> Copier <input type="checkbox"/> Supprimer	6	christian@christian.fr	\$2y\$10\$DB.4ofhCBfqZX7hmVBUt.ev6b77exz8nay0Dz7.QHEp...

Nous avons rajouté une fonction d'**insertion** pour chacun dans le **Modèle** afin de prendre en charge la requête **INSERT INTO**.

```

function insertResponsable($mailResponsable, $mdpResponsable)

{

    try {

        $cnx = connexionPDO();

```

```

        $req = $cnx->prepare("INSERT INTO responsable VALUES (NULL, :mailResponsable,
:mdpResponsable)");

        $req->execute(array(

            'mailResponsable' => $mailResponsable,

            'mdpResponsable' => $mdpResponsable

        ));

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}

```

Et également une **vueInscription** pour l'**arbitre** et le **responsable**.

```

<fieldset name="inscriptionresp">

    <h5>inscription responsable</h5>

    <input type="email" name="mail_responsable" placeholder="email" required
/></br>

    <input type="password" name="motdepasse" placeholder="mot de passe" required
/>

    <input type="submit" name="inscription" value="S'inscrire" />

</br>

</fieldset>

```

Une fois alliés au **CSS**, la **Vue** de l'**arbitre** ressemble à cela.

INSCRIPTION ARBITRE

A présent nous pouvons tester l'**inscription** d'un **arbitre**.

INSCRIPTION ARBITRE

Comme le montre la capture ci-dessus, nous inscrivons l'**arbitre Jacques Céler**.

Nous pouvons enfin vérifier si ce dernier a bien été enregistré dans la base de données.

num_arbitre	nom_arbitre	prenom_arbitre
25	Pasqualini	Claude
23	Bourgeois	Agnès
24	Bogusz	Thierry
26	Techer	Charles
27	Tournier	Alain
34	Céler	Jacques

C'est le cas. Donc, nous n'avons pas plus qu'à tester la **connexion**.



The screenshot shows a login interface titled "CONNEXION ARBITRE". It features a text input field containing the email "jacques@celert.fr", a password input field with masked characters ".....", and an orange "Se connecter" button. Below the button, there is a link that says "Cliquez ici pour vous inscrire".

La **connexion** fonctionnant également, ce dernier peut accéder à son **profil** et ainsi voir ses informations.



The screenshot shows a profile page titled "INFORMATIONS ARBITRE". It displays various user details in orange boxes: "Nom : Céler", "Prénom : Jacques", "Date de naissance : 1985-01-01", "Adresse : Autoroute", "Code Postal : 000000", "Ville : Partout", "Téléphone fixe : 05-05-05-05-05", "Téléphone portable : 06-06-06-06-06", "Adresse e-mail : jacques@celert.fr", and "Appartient au club : Elan Sportif".

Maintenant, nous allons nous occuper du **menu** que nous avons placé dans la **Vue entête**.

```
<ul class="menu">

    <a href="./?action=defaut">ACCUEIL</a>

    <a href="./?action=equipes">EQUIPES</a>

    <?php if (isLoggedInOnResponsable()) {
```

```

?>

<li><a href="#">**PROFIL** Responsable</a>

    <ul name=responsable>

        <a name=menuresponsable1 href="./?action=**profil**R">Mes
informations</a>

        <a name=menuresponsable2
href="./?action=deconnexionR">DECONNEXION</a>

    </ul>

</li>

<?php } elseif (isLoggedInOnArbitre()) {

?>

<li><a href="#">**PROFIL** Arbitre</a>

    <ul name=arbitre>

        <a name=menuarbitre1 href="./?action=**profil**A">Mes
informations</a>

        <a name=menuarbitre2
href="./?action=deconnexionA">DECONNEXION</a>

    </ul>

</li>

<?php } else { ?>

    <li><a href="./?action=connexion">CONNEXION</a>

    </li>

<?php }

?>

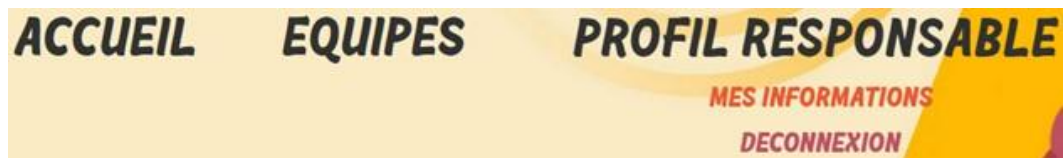
</ul>

```

Ce dernier comprend l'**accueil** et le lien « **LES EQUIPES** » qui amène à la **Gestion des équipes** dont nous avons parlé précédemment.

Comme nous le voyons ci-dessus, nous avons séparés le **menu** de l'**arbitre** et du **responsable**. La fonction **isLoggedInOn** nous permet de faire : si l'utilisateur est connecté, il a accès à son **profil** et au lien **déconnexion**. Et, dans le cas inverse, il n'a que accès au lien de **connexion**.

Une fois la **Vue** **entête** alliée au **CSS**, le menu du **responsable** ressemble à cela.



Mais si aucun des deux n'est connecté le **menu** est comme ci-dessous.

