

# **Projet "Gestion de Conférences" –**

# Présentation du Contexte

Le Centre de conférence de la ville de Poitiers accueille régulièrement des conférences ou des séminaires autour de thèmes variés. Profitant de la proximité du Futuroscope, ces événements permettent aux participants de combiner apprentissage et détente. Un séminaire est prévu pour le 12 octobre 2222, centré sur les questions urbaines, exclusivement ouvert aux acteurs des municipalités de divers départements.

## Objectifs

Gestion des inscriptions : Permettre aux représentants des municipalités de consulter le programme des conférences et de s'inscrire.

Exploitation des inscriptions : Fournir une interface sécurisée pour les administrateurs afin de visualiser et gérer les inscriptions.

## Technologies Utilisées

PHP : Langage de programmation côté serveur pour générer du contenu dynamique.

JSON : Format pour stocker et échanger les données de manière structurée.

HTML/CSS : Langages pour structurer et styliser les pages Web.

### Partie 1 : Gestion des inscriptions à un séminaire

#### 1. Contrôleur principal (controleurPrincipal.php) :

Définit les actions possibles et redirige vers les pages correspondantes.

```
<?php
function controleurPrincipal($action) {
    $lesActions = array();
    $lesActions["default"] = "inscription.php";
    $lesActions["inscription"] = "inscription.php";
    $lesActions["programme"] = "programme.php";
    $lesActions["choixconferences"] = "choixconferences.php";
    $lesActions["connexion"] = "connexion.php";
    $lesActions["inscrits"]="inscrits.php";
    $lesActions["validerConnexion"]="validerConnexion.php";

    if (array_key_exists ( $action , $lesActions )){
        return $lesActions[$action];
    }
    else{
        return $lesActions["default"];
    }
}
```

```
}  
?>
```

## 2. **Vue entete.php :**

- Affiche l'en-tête de la page avec des liens vers les différentes fonctionnalités.

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
    <link href="css/css.css" rel="stylesheet" type="text/css">  
    <title></title>  
  </head>  
<?php  
include_once "fonctions/fonctionsAccesDonnees.php";  
include_once "fonctions/fonctionsGestion.php";  
?  
<body>  
  <h1><?php echo donnerIntituleSeminaire(  
  
  );  
  
  ?></h1>  
  
  <table>  
    <tr>  
      <td><a href="./?action=inscription">Inscription</a></td>  
      <td><a href="./?action=programme">Voir le programme</a></td>  
      <td><a href="./?action=choixconferences">Choisir ses  
conférences</a></td>  
      <td><a href="./?action=connexion">Connexion admin</a></td>  
      <td><a href="./?action=inscrits">Voir les inscrits aux  
conférences</a></td>  
    </tr>  
  </table>
```

## Fonctions de chargement des données JSON

1. **chargeJSONseminaire() :**
  - Cette fonction charge les données du séminaire à partir du fichier JSON `seminaire.json`.
  - Elle retourne un tableau associatif représentant les données du séminaire.
2. **chargeJSONprofessions() :**
  - Charge les données des professions à partir du fichier JSON `professions.json`.
  - Retourne un tableau représentant les professions.
3. **chargeJSONadmin() :**
  - Charge les données de l'administrateur à partir du fichier JSON `admin.json`.
  - Retourne un tableau associatif représentant les données de l'administrateur.

## Fonctions pour sauvegarder les données JSON

1. **sauveJSONseminaire(\$doc) :**

- Cette fonction enregistre les données du séminaire dans le fichier `seminaire.json`.
- Prend en paramètre le tableau associatif contenant les données du séminaire.

## Fonctions pour récupérer des informations du séminaire

1. **donnerIntituleSeminaire()** :
  - Récupère l'intitulé du séminaire à partir des données chargées du fichier JSON.
  - Retourne une chaîne de caractères représentant l'intitulé du séminaire.
2. **donnerLesHeuresCreneaux()** :
  - Récupère la liste de tous les créneaux horaires (heures de début des conférences).
  - Retourne un tableau contenant les heures de début.
3. **donnerLesConferences(\$heure)** :
  - Récupère toutes les conférences commençant à l'heure donnée.
  - Prend en paramètre l'heure de début de la conférence.
  - Retourne un tableau contenant les informations sur les conférences.

## Fonctions pour gérer les inscriptions

1. **sauverDonneesInscription(\$nom, \$prenom, \$mail, \$ville, \$profession)** :
  - Sauvegarde les données d'inscription d'un visiteur dans le fichier `seminaire.json`.
  - Prend en paramètres le nom, le prénom, l'e-mail, la ville et la profession du visiteur.
2. **donnerLesProfessions()** :
  - Récupère toutes les professions possibles à partir du fichier JSON.
  - Retourne un tableau contenant les professions.
3. **estInscrit()** :
  - Vérifie si le visiteur a déjà rempli son formulaire d'inscription en vérifiant l'existence de variables de session.
  - Retourne un booléen indiquant si le visiteur est déjà inscrit.
4. **enregistre(\$lesChoix)** :
  - Enregistre les choix du visiteur dans le fichier `seminaire.json`.
  - Prend en paramètre un tableau associatif contenant les choix du visiteur.

```
3. <?php
/**
 * @access private
 * @return type
 */
function chargeJSONseminaire()
{
    // ***** A COMPLETER *****
    $json_source = file_get_contents('data/seminaire.json');
    $document = json_decode($json_source, true); // Convertit le JSON
    en un tableau associatif
    return $document;
}

/**
```

```

* @access private
* @return type
*/
function chargeJSONprofessions()
{
    // ***** A COMPLETER *****
    $json_source = file_get_contents('data/professions.json');
    $document = json_decode($json_source);
    return $document;
}

/**
* @access private
* @param type $doc
*/
function sauveJSONseminaire($doc) {
    // ***** A COMPLETER *****
    $json_data = json_encode($doc);
    file_put_contents('data/seminaire.json', $json_data);
}

/**
* @access private
* @return type
*/
function chargeJSONadmin()
{
    // ***** A COMPLETER *****
    $json_source = file_get_contents('data/admin.json');
    $document = json_decode($json_source, true);
    return $document;
}

/**
* Retourne l'intitulé du séminaire
* @return chaîne
*/
function donnerIntituleSeminaires()
{
    // ***** A COMPLETER *****
    $json_source = file_get_contents('data/seminaire.json');
    $document = json_decode($json_source);
    $intitule = $document->seminaire->intitule;
    return $intitule;
}

function donnerLesConferences($heure) {
    // ***** A COMPLETER *****
    //Ecrire la fonction fonction donnerLesConferences($heure) qui
    retourne toutes les conférences (id, description, intervenant, salle
    et nbplaces) commençant à l'heure donnée sous forme d'un tableau.
    $json_source = file_get_contents('data/seminaire.json');

    $document = json_decode($json_source);
    foreach ($document->seminaire->creneau as $value) {
        if ($value->heure == $heure) {
            return $value->conference;
        }
    }
}

```

```

    }
}

}

/**
 * Retourne la liste de tous les créneaux horaires, heures de début
des conférences
 * le tableau retourné commence à l'indice 0
 * @return tableau
 *
 */
function donnerLesHeuresCreneaux(){
    // ***** A COMPLETER *****
    //Ecrire la fonction fonction donnerLesHeuresCreneaux() qui
retourne la liste de tous les créneaux horaires (heures de début)
sous forme d'un tableau (le tableau retourné commence à l'indice 0).
    $json_source = file_get_contents('data/seminaire.json');

    $document = json_decode($json_source);

    return array_column($document->seminaire->creneau, 'heure');
}

/**
 * Retourne toutes les conférences commençant à l'heure donnée sous
forme d'un tableau
 * @param chaîne $heure
 * @return tableau
 *
 */
function donnerLesConfernces($heure){
    // ***** A COMPLETER *****
    //Ecrire la fonction fonction donnerLeslesConfernces($heure) qui
retourne toutes les conférences (id, description, intervenant, salle
et nbplaces) commençant à l'heure donnée sous forme d'un tableau.
    $json_source = file_get_contents('data/seminaire.json');

    $document = json_decode($json_source);
    foreach ($document->seminaire->creneau as $value) {
        if ($value->heure == $heure) {
            return $value->conference;
        }
    }
}

}

/**
 * Enregistre les informations d'un visiteur
 * @param chaîne $nom
 * @param chaîne $prenom
 * @param chaîne $mail
 * @param chaîne $ville
 * @param chaîne $profession
 *
 */
function sauverDonneesInscription($nom, $prenom, $mail, $ville,
$profession){
    // ***** A COMPLETER *****

    $json_source = file_get_contents('data/seminaire.json');
    $document = json_decode($json_source);
    $document->seminaire->participants[] = array(
        'nom' => $nom,

```

```

        'prenom' => $prenom,
        'mail' => $mail,
        'ville' => $ville,
        'profession' => $profession
    );
    $json_data = json_encode($document);
    file_put_contents('data/seminaire.json', $json_data);
}

/**
 * Retourne toutes les professions possibles
 * le tableau retourné commence à l'indice 0
 * @return tableau
 */
function donnerLesProfessions() {
    // ***** A COMPLETER *****
    //Ecrire la fonction fonction donnerLesProfessions() qui retourne
    toutes les professions possibles sous forme d'un tableau (le tableau
    retourné commence à l'indice 0).
    $json_source = file_get_contents('data/professions.json');
    $document = json_decode($json_source);
    return $document->professions;
}

/**
 * Vérifie si le visiteur a déjà rempli son formulaire d'inscription
 * @return boolean
 */
function estInscrit() {
    // ***** A COMPLETER *****
    //Ecrire la fonction fonction estInscrit() qui vérifie si le
    visiteur a déjà rempli son formulaire d'inscription.

    // Vérifiez l'existence de variables de session d'inscription
    return isset($_SESSION['nom']) &&
        isset($_SESSION['prenom']) &&
        isset($_SESSION['mail']) &&
        isset($_SESSION['ville']) &&
        isset($_SESSION['profession'])
    ;
}

/**
 * Enregistre un participant et ses choix de conférences
 * @param tableau $lesChoix : les choix du participant
 */
function enregistre($lesChoix) {
    $json_source = file_get_contents('data/seminaire.json');
    $document = json_decode($json_source);

    $participant = array(
        'nom' => $_SESSION['nom'],
        'prenom' => $_SESSION['prenom'],
        'profession' => $_SESSION['profession'],
        'ville' => $_SESSION['ville'],
        'mail' => $_SESSION['mail']
    );

```

```

        foreach ($lesChoix as $heure => $conferenceId) {
            foreach ($document->seminaire->creneau as &$creneau) {
                if ($creneau->heure == $heure) {
                    foreach ($creneau->conference as &$conference) {
                        if ($conference->id == $conferenceId) {
                            if (!isset($conference->participants)) {
                                $conference->participants = [];
                            }
                            $conference->participants[] = $participant;
                        }
                    }
                }
            }
        }

        $json_data = json_encode($document);
        file_put_contents('data/seminaire.json', $json_data);
    }

    /**
     * Retourne toutes conférences sous forme d'un tableau
     * Le tableau commence à l'indice 0
     * Chaque ligne du tableau contient les information sur une
     * conférence :
     * son id, son creneau, sa description, son intervenant, sa salle et
     * son nbPlaces
     * @return tableau
     */
    function donnerToutesLesConfernces() {
        //Ecrire la fonction fonction donnerToutesLesConferences() qui
        //retourne toutes conférences sous forme d'un tableau.
        // Chaque ligne du tableau contient les informations sur une
        // conférence : son id, son créneau, sa description, son intervenant, sa
        // salle et son nombre de places.
        $json_source = file_get_contents('data/seminaire.json');
        $document = json_decode($json_source);
        $conferences = array();
        foreach ($document->seminaire->creneau as $value) {
            foreach ($value->conference as $value2) {
                $conferences[] = array(
                    'id' => $value2->id,
                    'creneau' => $value->heure,
                    'description' => $value2->description,
                    'intervenant' => $value2->intervenant,
                    'salle' => $value2->salle,
                    'nbPlaces' => $value2->nbPlaces
                );
            }
        }
        return $conferences;
    }

    /**
     * Retourne tous les participants inscrits à une conférence dont on
     * fournit le numéro
     * Chaque ligne du tableau retourné contient le nom, le prénom, la
     * profession,
     * la ville et le mail d'un participant
     * @param entier $numConference

```



```

* @return tableau
*/
function donnerParticipants($numConference)
{
    // ***** A COMPLETER *****
    $json_source = file_get_contents('data/seminaire.json');
    $document = json_decode($json_source);
    $participants = array();
    foreach ($document->seminaire->participants as $value) {
        foreach ($value->choix as $value2) {
            if ($value2 == $numConference) {
                $participants[] = array(
                    'nom' => $value->nom,
                    'prenom' => $value->prenom,
                    'profession' => $value->profession,
                    'ville' => $value->ville,
                    'mail' => $value->mail
                );
            }
        }
    }
    return $participants;
}
?>

```

#### 4. Étapes du projet :

##### ○ Étape 1 : Programme du séminaire :

- Travail à faire : Écrire des fonctions pour récupérer les heures de créneaux et les conférences à partir des fichiers JSON.
- Travail à faire : Compléter la page `programme.php` pour afficher le programme du séminaire.
  - Inclusion des fichiers de fonctions (`fonctionsAccesDonnees.php` et `fonctionsGestion.php`) pour accéder aux données et gérer les erreurs.
  - Inclusion des fichiers de vue (`vueProgramme.php`, `entete.php`, et `pied.php`) pour afficher le contenu.
  - Utilisation de la fonction `donnerLesHeuresCreneaux()` pour obtenir la liste des heures de début des conférences.
  - parcourir chaque heure de début des conférences et afficher les détails de chaque conférence pour cette heure, y compris la description, l'intervenant et la salle.

```

<?php
include "../vue/entete.php";
include_once "fonctions/fonctionsAccesDonnees.php";
include_once "fonctions/fonctionsGestion.php";
?>

<table>
<?php
$heures = donnerLesHeuresCreneaux();
foreach ($heures as $heure) {
    $conferences = donnerLesConfernces($heure);
    if (!empty($conferences)) {
        echo "<tr>";
        echo "<th>$heure</th>";
        echo "<th>Intervenant</th>";
        echo "<th>Salle</th>";
        echo "</tr>";
    }
}

```

```

        foreach ($conferences as $conference) {
            echo "<tr>";
            echo "<td>$conference->description</td>";
            echo "<td>$conference->intervenant</td>";
            echo "<td>$conference->salle</td>";
            echo "</tr>";
        }
    }
    ?>
</table>

<?php
include "../vue/pied.php";
?>

```

## Seminaire du 12 octobre 2222, la ville et ses enjeux

Inscription	Voir le programme	Choisir ses conférences	Connexion admin	Voir les inscrits aux conférences
<b>10h</b>			<b>Intervenant</b>	<b>Salle</b>
Enjeux émergents, enjeux à construire pour la recherche urbaine			Alain Lecastel	salle Mozart
L'activité urbaine et l'environnement: de nouveaux modèles à construire			Yves Renard	salle Berlioz
Des infrastructures et des services urbains pour une ville économe?			Anne-Marie Breilinsky	salle Beethoven
<b>11h30</b>			<b>Intervenant</b>	<b>Salle</b>
Métropoles et aménagements métropolitains			Joseph Demongeot	salle Berlioz
La ville pour tous			Marc Touati	salle Mozart
Quelles dynamiques d'innovations pour les villes ?			Hervé Bachelard	salle Beethoven
<b>14h</b>			<b>Intervenant</b>	<b>Salle</b>
Urbanisme et environnement: des exemples en Europe			Yann Berlinne	salle Berlioz
Nouvelles vulnérabilités, nouvelles gouvernances			Hamed Djalaoui	salle Beethoven
<b>16h</b>			<b>Intervenant</b>	<b>Salle</b>
La ville et ses périphéries, quelles relations ?			Omar Azdine	salle Berlioz
La ville, quel avenir, quelles contraintes ?			Sophie Radon	salle Beethoven

Site réalisé par fatima

### 5. Formulaire d'inscription :

- Travail à faire : Intégrer la gestion des erreurs et la vérification des données saisies dans `inscription.php`.
- Travail à faire : Sauvegarder les données d'inscription dans des variables de session.
  - Inclusion des fichiers de fonctions (`fonctionsAccesDonnees.php` et `fonctionsGestion.php`) pour accéder aux données et gérer les erreurs.
  - Inclusion du fichier de vue `vueVoirInscriptions.php` pour afficher les inscriptions existantes.
  - Inclusion du fichier `entete.php` pour l'en-tête de la page.
  - Utilisation d'un formulaire pour saisir les informations d'un nouvel inscrit (nom, prénom, e-mail, ville, profession).
  - Utilisation de la fonction `donnerLesProfessions()` pour obtenir la liste des professions possibles et les afficher dans un menu déroulant.
  - Gestion de la soumission du formulaire avec la méthode POST.
  - Affichage des boutons "Valider" et "Annuler" pour envoyer ou réinitialiser le formulaire.
  - Inclusion du fichier `pied.php` pour le pied de page.

6.

```
<?php
if (isset($_POST["btn"])){
    $btn = $_POST["btn"];
}

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

    <table>
        <tr>
            <td>Nouvelle inscription</td>
        </tr>
        <tr>
            <td> Nom*: </td>
            <td><input type="text" name="nom" value="" size="30"
required></td>
        </tr>
        <tr >
            <td> Prenom*: </td>
            <td><input type="text" name="prenom" value="" size="30"
required></td>
        </tr>
        <tr >
            <td> mail*: </td>
            <td><input type="email" name="mail" value="" size="30"
required></td>
        </tr>
        <tr>
            <td> Ville*: </td>
            <td><input type="text" name="ville" value="" size="30"
required></td>
        </tr>
        <tr>
            <td> Profession*: </td>
            <td><SELECT name= "profession" required>
                <?php
                    $lesProfessions = donnerLesProfessions();
                    foreach ($lesProfessions as $profession) {
                        echo "<option
value='$profession'>$profession</option>";
                    }
                ?>
            </SELECT>
        </td>
        </tr> </table>
    <br>

    <table >
        <tr>
            <td ><input type="submit" value="valider" name="btn">
            </td>
            <td ><input type="reset" value="Annuler" name="btn">
            </td>
        </tr>
    </table>
</form>
<?php
include "../vue/pied.php";
?>
```

- La session est démarrée, et les fichiers de fonctions sont inclus.
- Si l'utilisateur est déjà inscrit (indiqué par la variable de session `$_SESSION['inscrit']`), un message indiquant qu'il est déjà inscrit est affiché, avec un lien pour procéder au choix des conférences.
- Sinon, si l'utilisateur a soumis le formulaire d'inscription et a appuyé sur le bouton "valider", les données du formulaire sont récupérées.
- Les données du formulaire sont vérifiées à l'aide de la fonction `verifierDonneesInscription()`. Si des erreurs sont détectées, elles sont affichées à l'utilisateur.
- Si aucune erreur n'est détectée, les données sont stockées dans des variables de session (`$_SESSION['nom']`, `$_SESSION['prenom']`, etc.).
- L'utilisateur est redirigé vers la page de choix des conférences (`header("location: ../?action=choixconferences")`).

```

7. <?php
    session_start();
    include "../vue/entete.php";
    include_once "fonctions/fonctionsAccesDonnees.php";
    include_once "fonctions/fonctionsGestion.php";

    if (isset($_SESSION['inscrit']) && $_SESSION['inscrit'] === true) {
        echo "<h2>Vous êtes déjà inscrit</h2>";
        echo "<a href='../?action=choixconferences'>Procéder au choix des
conférences</a>";
    } else {
        if (isset($_POST['btn']) && $_POST['btn'] == "valider") {
            // Récupération des données du formulaire
            $nom = $_POST['nom'];
            $prenom = $_POST['prenom'];
            $mail = $_POST['mail'];
            $ville = $_POST['ville'];
            $profession = $_POST['profession'];

            // Vérification des données saisies
            verifierDonneesInscription($nom, $prenom, $mail, $ville);

            // Vérification du nombre d'erreurs
            $nbErreurs = donnerNbErreurs();

            if ($nbErreurs > 0) {
                afficherErreurs();
            } else {
                // Stockage des données dans des variables de session
                $_SESSION['nom'] = $nom;
                $_SESSION['prenom'] = $prenom;
                $_SESSION['mail'] = $mail;
                $_SESSION['ville'] = $ville;
                $_SESSION['profession'] = $profession;
                $_SESSION['inscrit'] = true;

                // Redirection vers la page de choix des conférences
                header("location: ../?action=choixconferences");
                exit;
            }
        }
    }
?>

<form action="" method="POST">
    <table>

```

```

        <tr>
            <td>Nom*: </td>
            <td><input type="text" name="nom" size="30"
required></td>
        </tr>
        <tr>
            <td>Prénom*: </td>
            <td><input type="text" name="prenom" size="30"
required></td>
        </tr>
        <tr>
            <td>Email*: </td>
            <td><input type="email" name="mail" size="30"
required></td>
        </tr>
        <tr>
            <td>Ville*: </td>
            <td><input type="text" name="ville" size="30"
required></td>
        </tr>
        <tr>
            <td>Profession*: </td>
            <td>
                <select name="profession" required>
                    <?php
                        $lesProfessions = donnerLesProfessions();
                        foreach ($lesProfessions as $profession) {
                            echo "<option
value='$profession'>$profession</option>";
                        }
                    <?>
                </select>
            </td>
        </tr>
    </table>
    <br>
    <input type="submit" value="valider" name="btn">
    <input type="reset" value="annuler" name="btn">
</form>

    <?php
}
include "../vue/pied.php";
?>

```

## Seminaire du 12 octobre 2222, la ville et ses enjeux

Inscription	Voir le programme	Choisir ses conférences	Connexion admin	Voir les inscrits aux conférences
-------------	-------------------	-------------------------	-----------------	-----------------------------------

  

Nom*:	<input type="text"/>
Prénom*:	<input type="text"/>
Email*:	<input type="text"/>
Ville*:	<input type="text"/>
Profession*:	<input type="text" value="Directeur de l'urbanisme"/>

  

## Partie 2 : Exploitation des inscriptions à un séminaire

- Fonctions de gestion des erreurs (`fonctionsGestion.php`) :**
  - Contient des fonctions pour vérifier les données saisies et gérer les erreurs.
  - Travail à faire : Comprendre le fonctionnement des fonctions existantes.
- Page `choixconferences.php` :**
  - Travail à faire : Afficher la liste des conférences et des salles pour chaque créneau.
  - Travail à faire : Enregistrer les choix du visiteur dans le fichier JSON.
- Fonctions d'accès aux données :**
  - Travail à faire : Écrire des fonctions pour enregistrer les choix du visiteur et vérifier s'il est déjà inscrit.

Le premier bloc de code semble être destiné à gérer le choix des conférences par les participants déjà inscrits. Voici un résumé de ce qu'il fait :

- Démarrage de la session et inclusion des fichiers de fonctions (`fonctionsAccesDonnees.php` et `fonctionsGestion.php`).
- Vérification si l'utilisateur est inscrit en session. S'il ne l'est pas, un message est affiché et l'exécution s'arrête.
- Si l'utilisateur a soumis le formulaire de choix de conférences et a appuyé sur le bouton "valider", les choix sont récupérés et enregistrés à l'aide de la fonction `enregistre()`. Un message de confirmation est affiché.
- Affichage d'un formulaire permettant de choisir les conférences par créneau horaire, avec des boutons de validation et d'annulation.

Le deuxième bloc de code semble être destiné à gérer la présentation du formulaire de choix de conférences pour les nouveaux inscrits. Voici un résumé de ce qu'il fait :

- Inclusion du fichier `entete.php`.
- Vérification si l'utilisateur est déjà inscrit. Si oui, un message indiquant qu'il est déjà inscrit est affiché. Sinon, un message indiquant que l'inscription a été prise en compte et qu'il faut procéder au choix des conférences est affiché.

- Affichage d'un formulaire permettant de choisir les conférences par créneau horaire, avec des boutons de validation et d'annulation.

```
<?php
session_start();
include "../vue/entete.php";
include_once "fonctions/fonctionsAccesDonnees.php";
include_once "fonctions/fonctionsGestion.php";

if (!isset($_SESSION['inscrit']) || $_SESSION['inscrit'] !== true) {
    echo "<h2>Vous devez d'abord vous inscrire</h2>";
    echo "<a href='../?action=inscription'>Retour à l'inscription</a>";
    include "../vue/pied.php";
    exit;
}

if (isset($_POST['btn']) && $_POST['btn'] == "valider") {
    $lesChoix = [];
    foreach ($_POST as $key => $value) {
        if ($key != 'btn') {
            $heure = $key;
            $conferenceId = $value;
            $lesChoix[$heure] = $conferenceId;
        }
    }
    enregistre($lesChoix);
    echo "<h2>Vos choix de conférences ont été enregistrés</h2>";
    include "../vue/pied.php";
    exit;
}
?>

<form action="" method="POST">
    <table>
        <?php
        $heures = donnerLesHeuresCreneaux();
        foreach ($heures as $heure) {
            $conferences = donnerLesConferences($heure);
            if ($conferences) {
                echo "<tr><th>$heure</th></tr>";
                foreach ($conferences as $conference) {
                    echo "<tr>";
                    echo "<td>{$conference->description}</td>";
                    echo "<td>{$conference->salle}</td>";
                    echo "<td><input type='radio' name='$heure'";
                    echo "<td>";
                    echo "<td>{$conference->id}</td>";
                    echo "<td></td>";
                    echo "</tr>";
                }
            }
        }
        ?>
    </table>
    <br>
    <input type="submit" value="valider" name="btn">
    <input type="reset" value="annuler" name="btn">
</form>

<?php
include "../vue/pied.php";
?>
```

```

4. <?php
include "../vue/entete.php";
if (estInscrit()) {
    echo "<h2>Vous êtes déjà inscrit</h2>";
} else {
    echo "<h2>Votre inscription a été prise en compte, il faut
procéder au choix des conférences</h2>";
}

?>

<form action="./?action=choixconferences" method="POST">
    <table>
        <?php
        $heures = donnerLesHeuresCreneaux();
        foreach ($heures as $heure) {
            $conferences = donnerLesConfernces($heure);
            if ($conferences) {
                echo "<tr>";
                echo "<th>$heure</th>";

                echo "</tr>";
                foreach ($conferences as $conference) {
                    echo "<tr>";
                    echo "<td>$conference->description</td>";
                    echo "<td>$conference->salle</td>";
                    echo "<td><input type='radio' name='$heure'
value='$conference->id'></td>";
                }
            }
        }
        ?>
    </tr>
</table>
    <br><br>
    <table>
        <tr>
            <td><input type="submit" value="valider" name="btn"></td>
            <td><input type="reset" value="annuler" name="btn"></td>
        <tr>

```



```

    </table>
</form>

</table>

<?php
include "../vue/pied.php";
?>

```

Inscription	Voir le programme	Choisir ses conférences	Connexion admin	Voir les inscrits aux conférences
<b>10h</b>				
Enjeux émergents, enjeux à construire pour la recherche urbaine		salle Mozart	<input checked="" type="radio"/>	
L'activité urbaine et l'environnement: de nouveaux modèles à construire		salle Berlioz	<input type="radio"/>	
Des infrastructures et des services urbains pour une ville économe?		salle Beethoven	<input type="radio"/>	
<b>11h30</b>				
Métropoles et aménagements métropolitains		salle Berlioz	<input type="radio"/>	
La ville pour tous		salle Mozart	<input checked="" type="radio"/>	
Quelles dynamiques d'innovations pour les villes ?		salle Beethoven	<input type="radio"/>	
<b>14h</b>				
Urbanisme et environnement: des exemples en Europe		salle Berlioz	<input type="radio"/>	
Nouvelles vulnérabilités, nouvelles gouvernances		salle Beethoven	<input type="radio"/>	
<b>16h</b>				
La ville et ses périphéries, quelles relations ?		salle Berlioz	<input type="radio"/>	
La ville, quel avenir, quelles contraintes ?		salle Beethoven	<input type="radio"/>	

une page d'administration permettant aux utilisateurs ayant des privilèges d'administrateur d'accéder à certaines fonctionnalités spécifiques, telles que la visualisation des inscriptions aux conférences.

1. La session est démarrée, et les fichiers de fonctions sont inclus.
2. Si l'utilisateur n'est pas un administrateur (vérifié à l'aide de la fonction `estAdmin()`), un message "Accès interdit" est affiché, et le script s'arrête.
3. Une table est affichée pour afficher les informations sur les conférences et les participants.
4. Pour chaque heure de conférence, les conférences programmées à cette heure sont récupérées à l'aide de la fonction `donnerLesConferences()`.
5. Pour chaque conférence, les détails tels que l'identifiant de la conférence, l'heure, la description et la salle sont affichés dans une ligne de tableau.
6. Si des participants sont inscrits à la conférence, leurs informations sont également affichées dans des lignes de tableau supplémentaires.
7. Si aucun participant n'est inscrit, un message "Aucun participant inscrit" est affiché.

```

8. <?php
    session_start();

    include "../vue/entete.php";
    include_once "fonctions/fonctionsAccesDonnees.php";
    include_once "fonctions/fonctionsGestion.php";
    if (!estAdmin()) {
        echo "<h2>Accès interdit</h2>";
        include "../vue/pied.php";
        exit;
    }
?>

```

```

<table >

<?php
$heures = donnerLesHeuresCreneaux();
foreach ($heures as $heure) {
    $conferences = donnerLesConferences($heure);
    if (!empty($conferences)) {
        foreach ($conferences as $conference) {
            echo "<tr>";
            echo "<th>{$conference->id}</th>";
            echo "<th>{$heure}</th>";
            echo "<th>{$conference->description}</th>";
            echo "</tr>";
            if (!empty($conference->participants)) {
                foreach ($conference->participants as
$participant) {
                    echo "<tr>";
                    echo "<td>{$participant->nom}</td>";
                    echo "<td>{$participant->prenom}</td>";
                    echo "<td>{$participant->profession}</td>";
                    echo "<td>{$participant->ville}</td>";
                    echo "<td>{$participant->mail}</td>";
                    echo "<td>{$conference->salle}</td>";

                }
            } else {
                echo "<td>Aucun participant inscrit</td>";
                echo "</tr>";
            }
        }
    }
}
?>
</tbody>
</table>
<?php
include "../vue/pied.php";
?>

```

Inscription	Voir le programme	Choisir ses conférences	Connexion admin	Voir les inscrits aux conférences		
<b>1</b>	<b>10h</b>	<b>Enjeux émergents, enjeux à construire pour la recherche urbaine</b>				
Roberto	Jean	Directeur de l'urbanisme		Rennes	jroberto@free.fr	salle Mozart
ABOUDA	FATIMA	Technicien territorial		Limoges	email@email.com	salle Mozart
<b>2</b>	<b>10h</b>	<b>L'activité urbaine et l'environnement: de nouveaux modèles à construire</b>				
Aucun participant inscrit						
<b>3</b>	<b>10h</b>	<b>Des infrastructures et des services urbains pour une ville économe?</b>				
Aucun participant inscrit						
<b>4</b>	<b>11h30</b>	<b>Métropoles et aménagements métropolitains</b>				
Aucun participant inscrit						
<b>5</b>	<b>11h30</b>	<b>La ville pour tous</b>				
Roberto	Jean	Directeur de l'urbanisme		Rennes	jroberto@free.fr	salle Mozart
NOM	PRENOM	Chef de projet urbanisme		City	nom@prenom.com	salle Mozart
<b>6</b>	<b>11h30</b>	<b>Quelles dynamiques d'innovations pour les villes ?</b>				
Aucun participant inscrit						
<b>7</b>	<b>14h</b>	<b>Urbanisme et environnement: des exemples en Europe</b>				
ABOUDA	FATIMA	Technicien territorial		Limoges	email@email.com	salle Berlioz

gérer à la fois l'authentification de l'utilisateur et l'affichage de la page de connexion.

Voici un aperçu de son fonctionnement :

1. Les fichiers de fonctions pour l'accès aux données et la gestion sont inclus.
2. La session est démarrée avec `session_start()`.
3. Si des données de formulaire sont soumises via la méthode POST (vérifié par `isset($_POST['submit'])`), le code tente de vérifier les informations de connexion (nom d'utilisateur et mot de passe) en appelant la fonction `verifier()`.
  - Si la vérification réussit, un message "Vous êtes connecté en tant qu'administrateur." est affiché.
  - Sinon, un message d'erreur est affiché.
4. Ensuite, la vue de connexion (`vueConnexion.php`) est incluse pour afficher le formulaire de connexion.
5. Un autre formulaire de connexion est affiché à l'utilisateur, où il peut saisir son nom d'utilisateur et son mot de passe.
6. Si l'utilisateur est déjà connecté en tant qu'administrateur (vérifié par `estAdmin()`), un message de bienvenue et un lien pour voir les inscriptions sont affichés.
7. Le formulaire de connexion affiche les champs pour le login et le mot de passe, avec des boutons "Valider" et "Annuler".

```
<?php

include_once "../fonctions/fonctionsAccesDonnees.php";
include_once "../fonctions/fonctionsGestion.php";

// Assurez-vous d'appeler session_start() au début de chaque page où vous
voulez utiliser les sessions.
session_start();

if (isset($_POST['submit'])) {
    $login = $_POST['login'];
    $mdp = $_POST['mdp'];

    if (verifier($login, $mdp)) {
        // L'utilisateur est connecté en tant qu'administrateur
        echo "Vous êtes connecté en tant qu'administrateur.";
    } else {
        // Erreur de connexion, affichez un message d'erreur
        echo "Erreur de connexion. Veuillez réessayer.";
    }
}

include "../vue/vueConnexion.php";
?>
<?php
include "../vue/entete.php";
include_once "fonctions/fonctionsAccesDonnees.php";
include_once "fonctions/fonctionsGestion.php";
?>
<form method="POST" action="./?action=validerConnexion"
>

    <?php
    //si il est déjà connecté on affiche un message de bienvenue et un lien
```

```

pour voir les inscriptions
    if (estAdmin()) {
        echo "<h2>Bienvenue</h2>";
        echo "<p>Vous êtes connecté en tant qu'administrateur</p>";
        echo "<p>Vous pouvez maintenant voir les inscriptions</p>";
        echo "<a href='./?action=inscrits'>Voir les inscriptions</a>";
    }
?>
<table>
    <tr>
        <td>Connexion</td>
    </tr>
    <tr>
        <td>Login*: </td>
        <td><input type="text" name="login" value="" size="15"></td>
    </tr>
    <tr >
        <td>Mot de passe*: </td>
        <td><input type="password" name="mdp" value="" size="15"></td>
    </tr>
</table>
<br>
<table >
    <tr>
        <td ><input type="submit" value="Valider" name="btn">
        </td>
        <td ><input type="reset" value="Annuler" name="btn">
        </td>
    </tr>
</table>

</form>
<?php
include "../vue/pied.php";
?>

```

Connexion	
Login*:	<input type="text"/>
Mot de passe*:	<input type="password"/>
<input type="submit" value="Valider"/> <input type="reset" value="Annuler"/>	

## Bienvenue

Vous êtes connecté en tant qu'administrateur

Vous pouvez maintenant voir les inscriptions

Voir les inscriptions

Connexion	
Login*:	<input type="text"/>
Mot de passe*:	<input type="password"/>
<input type="submit" value="Valider"/> <input type="reset" value="Annuler"/>	