

# Séminaire

## Présentation du contexte

Le Centre de conférence de Poitiers accueille régulièrement des événements, dont un séminaire prévu le 12 octobre 2222. Ce séminaire d'une journée prévoit plusieurs conférences qui, pour certaines, se déroulent en parallèle. Les données sont stockées dans des fichiers JSON, et le site web utilise PHP pour manipuler ces informations et afficher les formulaires.

## Structure MVC du projet

Le projet utilise une architecture **MVC** (Modèle-Vue-Contrôleur) pour gérer les inscriptions aux conférences et leur affichage.

## Partie 1 : Gestion des inscriptions

### Étape 1 : Récupération des horaires des créneaux et des conférences

#### **donnerLesHeuresCreneaux()**

Cette fonction récupère les heures des créneaux des conférences à partir d'un fichier JSON :

```
function donnerLesHeuresCreneaux() {  
    $document = chargeJSONseminaire();  
    $heuresCreneaux = array();  
    foreach($document->seminaire->creneau as $creneau) {  
        $heuresCreneaux[] = $creneau->heure;  
    }  
    return $heuresCreneaux;  
}
```

#### **donnerLesConferences(\$heure)**

Cette fonction retourne les conférences programmées pour une heure donnée :

```

function donnerLesConferences($heure) {
    $document = chargeJSONseminaire();
    $conferences = array();
    foreach ($document->seminaire->creneau as $creneau) {
        if ($creneau->heure == $heure) {
            foreach ($creneau->conference as $conf) {
                $conferences[] = $conf;
            }
        }
    }
    return $conferences;
}

```

### Contrôleur de programme

Le contrôleur récupère les heures des créneaux et les conférences associées :

```

include "fonctions/fonctionsAccesDonnees.php";
$lesCreneaux = donnerLesHeuresCreneaux();
$lesConferencesParCreneau = array();
foreach ($lesCreneaux as $creneau) {
    $lesConferencesParCreneau[$creneau] = donnerLesConferences($creneau);
}
include_once "vue/vueProgramme.php";

```

### vueProgramme

Affiche un tableau des conférences par créneau horaire :

```

<table>
<?php
    foreach ($lesCreneaux as $creneau) {

```

```

        echo "<tr><th>$creneau</th><th>Intervenant</th><th>Salle</th></tr>";
        foreach ($lesConferencesParCreneau[$creneau] as $conference) {
            echo "<tr><td>$conference->description</td><td>$conference-
>intervenant</td><td>$conference->salle</td></tr>";

        }
    }
?>
</table>

```

## Étape 2 : Enregistrement des données d'inscription

### **sauverDonneesInscription()**

Cette fonction enregistre les informations personnelles de l'utilisateur dans une session :

```

function sauverDonneesInscription($nom, $prenom, $mail, $ville, $profession)
{
    session_start();

    $_SESSION['nom'] = $nom;
    $_SESSION['prenom'] = $prenom;
    $_SESSION['mail'] = $mail;
    $_SESSION['ville'] = $ville;
    $_SESSION['profession'] = $profession;
}

```

### **donnerLesProfessions()**

Retourne une liste des professions depuis un fichier JSON :

```

function donnerLesProfessions(){
    $document = chargeJSONprofessions();

    $professions = array();
}

```

```
foreach ($document->professions as $profession) {  
    $professions[] = $profession;  
}  
return $professions;  
}
```

### Contrôleur d'inscription

Gère les actions liées au formulaire d'inscription (annulation ou validation)

```
$btn = "Inscription";  
if (isset($_POST["btn"])) { $btn = $_POST["btn"]; }  
switch ($btn) {  
    case "Annuler": /* Réinitialise les champs */  
    case "Valider":  
        $nom = $_POST['nom'];  
        $prenom = $_POST['prenom'];  
        /* ... validation et sauvegarde des données ... */  
        break;  
}
```

### vueInscription

Formulaire HTML d'inscription :

```
<form action="/?action=inscription" method="POST">  
    <table>  
        <tr><td>Nom: </td><td><input type="text" name="nom"  
required></td></tr>  
        <tr><td>Prénom: </td><td><input type="text" name="prenom"  
required></td></tr>
```

```

    <tr><td>Email: </td><td><input type="text" name="mail"
required></td></tr>

    <tr><td>Ville: </td><td><input type="text" name="ville"
required></td></tr>

    <tr><td>Profession*: </td><td>

        <SELECT name="profession" required>

            <?php foreach ($lesProfessions as $profession) { echo
"<option>$profession</option>"; } ?>

        </SELECT>

    </td></tr>

</table>

<input type="submit" value="Valider">

</form>

```

### Étape 3 : Choix des conférences

#### choixconferences.php

Récupère et affiche les conférences disponibles pour que l'utilisateur puisse faire ses choix :

```

session_start();

include "fonctions/fonctionsAccesDonnees.php";

if (!estInscrit()) {

    echo "Erreur : Vous devez d'abord vous inscrire.";

    exit;

}

$lesConferences = donnerToutesLesConferences();

$heures = donnerLesHeuresCreneaux();

```

#### enregistre(\$lesChoix)

Enregistre les choix de conférences de l'utilisateur dans le fichier JSON :

```

function enregistre($lesChoix) {
    $document = chargeJSONseminaire();
    foreach ($lesChoix as $choix) {
        foreach ($document->seminaire->creneau as $creneau) {
            foreach ($creneau->conference as $conference) {
                if ($conference->id == $choix) {
                    $conference->participants[] = [
                        'nom' => $_SESSION['nom'],
                        'prenom' => $_SESSION['prenom'],
                        'mail' => $_SESSION['mail'],
                        'ville' => $_SESSION['ville'],
                        'profession' => $_SESSION['profession']
                    ];
                }
            }
        }
    }
    sauveJSONseminaire($document);
}

```

### **vueChoixconferences**

Affiche les conférences et permet de sélectionner celles auxquelles l'utilisateur souhaite participer :

```

<form action="./?action=choixconferences" method="POST">
    <table>
        <?php foreach ($heures as $heure) {
            echo "<th>$heure</th>";
            foreach ($conferencesParHeure[$heure] as $conference) {

```

```
        echo "<tr><td>{$conference->description}</td><td>{$conference->salle}</td>";

        echo "<td><input type='radio' name='choix_{$heure}'
value='{$conference->id}'></td></tr>";

    }

} ?>

</table>

<input type="submit" value="valider">

</form>
```

## Partie 2 : Exploitation des inscriptions

### Fonctions d'administration

**chargeJSONAdmin()** Cette fonction permet de charger le fichier admin.json et de le transformer en objet PHP :

```
function chargeJSONAdmin() {

    $json_source = file_get_contents('data/admin.json');

    $document = json_decode($json_source);

    return $document;

}
```

### verifier(\$login, \$mdp)

Cette fonction vérifie si les identifiants (login et mot de passe) fournis sont corrects en les comparant avec ceux présents dans le fichier admin.json :

```
function verifier($login, $mdp) {

    $json_source = file_get_contents('data/admin.json');

    $document = json_decode($json_source, true);
```

```

foreach ($document['users'] as $user) {
    if ($user['login'] === $login && $user['mdp'] === $mdp) {
        $_SESSION['login'] = $login;
        $_SESSION['mdp'] = $mdp;
        $_SESSION['admin'] = 1;
        return true;
    }
}
return false;
}

```

**estAdmin()** Cette fonction retourne true si l'utilisateur est connecté en tant qu'administrateur :

```

function estAdmin() {
    return isset($_SESSION['admin']) && $_SESSION['admin'] == 1;
}

```

### Contrôleur de connexion.php

Ce contrôleur gère la logique de connexion de l'administrateur :

```

session_start();
$btn = "validerConnexion";
if (isset($_POST["btn"])) {
    $btn = $_POST["btn"];
    $login = $_POST["login"];
    $mdp = $_POST["mdp"];
}

```



```

switch ($btn) {
    case "Annuler":
        $login = "";
        $mdp = "";
        break;

    case "Valider":
        if (!empty($login) && !empty($mdp)) {
            if (verifier($login, $mdp)) {
                $_SESSION['admin'] = 1;
                echo "<h2>Connexion réussie</h2>";
            } else {
                echo "<h2>Identifiants incorrects</h2>";
            }
        } else {
            echo "<h2>Veuillez remplir tous les champs</h2>";
        }
        break;
}

```

**Vue de connexion** Le formulaire HTML de connexion à l'administration :

```

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

```

## Fonctions liées aux conférences et participants

**donnerToutesLesConferences()** Cette fonction retourne un tableau contenant toutes les conférences disponibles :

```

function donnerToutesLesConferences() {
    $document = chargeJSONseminaire();
    $conferences = array();

    foreach ($document->seminaire->creneau as $creneau) {
        foreach ($creneau->conference as $conference) {
            $conferences[] = array(
                "id" => $conference->id,
                "creneau" => $creneau->heure,
                "description" => $conference->description,
                "intervenant" => $conference->intervenant,
                "salle" => $conference->salle,
                "nbplaces" => $conference->nbplaces,
            );
        }
    }
}

```

```
    );  
    }  
}  
  
return $conferences;  
}
```

### **donnerParticipants(\$numConference)**

Cette fonction retourne les participants inscrits à une conférence en fonction de son numéro :

```
function donnerParticipants($numConference) {  
    $document = chargeJSONseminaire();  
    $tabPart = [];  
    foreach ($document->seminaire->creneau as $creneau) {  
        foreach ($creneau->conference as $conference) {  
            if ($conference->id == $numConference) {  
                foreach ($conference->participants as $participant) {  
                    $tabPart[] = $participant;  
                }  
            }  
        }  
    }  
    return $tabPart;  
}
```

### **Contrôleur voirinscription.php**

Ce contrôleur vérifie si l'utilisateur est administrateur avant d'afficher les inscriptions :

```
session_start();
```

```
if (estAdmin()) {  
    include "../vue/vueVoirInscription.php";  
} else {  
    header("Location: ../?action=connexion");  
}
```

### **Vue vueVoirInscription.php**

Cette vue affiche les inscriptions pour chaque conférence, ainsi que les détails des participants :

```
$LesConferences = donnerToutesLesConferences();  
foreach ($LesConferences as $uneConference) {  
    $lesParticipants = donnerParticipants($uneConference["id"]);  
  
    echo "<table>";  
    echo "<th>" . $uneConference["id"] . "</th>";  
    echo "<th>" . $uneConference["creneau"] . "</th>";  
    echo "<th>" . $uneConference["description"] . "</th>";  
  
    if (!empty($lesParticipants)) {  
        foreach ($lesParticipants as $unParticipant) {  
            echo "<tr>";  
            echo "<td>" . $unParticipant->nom . "</td>";  
            echo "<td>" . $unParticipant->prenom . "</td>";  
            echo "<td>" . $unParticipant->ville . "</td>";  
            echo "<td>" . $unParticipant->profession . "</td>";  
            echo "<td>" . $unParticipant->mail . "</td>";  
            echo "</tr>";  
        }  
    }  
}
```

```
    } else {  
        echo "<tr><td>Il n'y a aucun participant pour cette  
conférence</td></tr>";  
    }  
    echo "</table>";  
}
```

## **Conclusion :**

En résumé, ce projet de gestion de séminaire en PHP permet de manipuler des données stockées dans des fichiers JSON via une architecture MVC. Les fonctionnalités incluent l'inscription des participants, la gestion des conférences, et la vérification des administrateurs pour accéder aux données sensibles. L'application permet également de visualiser les inscriptions à chaque conférence avec les détails des participants. Le système repose sur des fonctions pour charger, vérifier et enregistrer des données, tout en offrant une interface simple pour la connexion des administrateurs et la gestion des participants.

Jordan Desbordes