# Documentation Technique

# Introduction :

### Sommaire :

Le but du projet est de faire une application web pour l’inscription et la gestion d’un séminaire. Le projet est élaboré suivant l’architecture MVC (Modèle Vue Contrôleur) pour garantir une séparation claire des responsabilités entre la logique, la gestion des données et l’affichages

Architecture du projet :

* contrôleur : gestion des fonctions
* datas : il y a tous les fichiers json
* fonctions : toutes les fonctions du projet
* vue : affichage sur le web

## Partie 1 : Gestion des Inscriptions

#### Etape 1 : Programme du Séminaire

Une image contenant capture d’écran, texte, nombre, ligne

Description générée automatiquement

**function donnerLesHeuresCreneaux()** retourne un tableau contenant tous les créneaux horaires disponibles (heures de début).

**function donnerLesConferences($heure)** retourne toutes les conférences programmées pour l'heure donnée. Les informations incluent l'identifiant, la description, l'intervenant, la salle et le nombre de places disponibles.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Dans le controleur programme.php on appel la fonction **donnerLesHeuresCrenaux()** et **donnerLesConferences()** en fonction de l’heure.

Dans le contrôleur programme.php, les fonctions **donnerLesHeuresCreneaux()** et **donnerLesConferences**() sont appelées en fonction de l'heure sélectionnée. Le résultat est ensuite affiché dans la vue, où un foreach permet de lister toutes les conférences avec leurs intervenants et la salle correspondante.

Puis dans la vue on fait un foreachqui permet que pour chaque heure, on obtient toutes les conférences avec le nom, l’intervenant et la salle.

Base pour JSON

function chargeJSONseminaire()

{

    $json\_source = file\_get\_contents('data/seminaire.json');

    $document = json\_decode($json\_source);

    return $document;

}

function sauveJSONseminaire($doc){

    $json\_data = json\_encode($doc, JSON\_PRETTY\_PRINT);

    file\_put\_contents('data/seminaire.json', $json\_data);

}

#### Etape 2 : Inscription des visiteurs

Une image contenant texte, capture d’écran, nombre, Parallèle

Description générée automatiquement

**function sauverDonneesInscription($nom, $prenom,$mail,$ville, $profession)** enregistre les informations d'inscription du visiteur dans une variable de session. Chaque paramètre représente respectivement : le nom, prénom, email, ville et profession du visiteur.

**function donnerLesProfessions()** on utilise la fonction **chargeJSONprofessions()** qui contient toutes les professions puis dans la vue on fait une boucle avec un select pour afficher chaque professions.

Dans le contrôleur inscription on appelle la fonction **donnerLesProfessions()** puis on utilise un switch pour le bouton si on clique sur annuler ça ne fait rien et sur valider ça vérifie les données de l’inscription puis ça les sauvegarde

#### Etape 3 : Choix des conférences

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**function estInscrit()** vérifie l'existence d'une variable de session pour déterminer si le visiteur a déjà rempli le formulaire d'inscription. Retourne un booléen (`true` ou `false`).

**function enregistre($lesChoix)** qui enregistre un participant et ses choix de conférences dans le fichier seminaire.json.

**Résumé Partie 1 :**

1. Affichage des créneaux horaires disponibles.

2. Inscription des visiteurs via la sauvegarde des données dans une session.

3. Enregistrement des choix de conférences dans le fichier `seminaire.json`.

## Partie 2 : Exploitation des inscriptions

#### Etape 1 : Connexion Admin

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

**function chargeJSONadmin()** :

Cette fonction charge le contenu du fichier admin.json et le transforme en un objet JSON. Ce fichier contient les informations de connexion des administrateurs, telles que leur identifiant et mot de passe.

**function verifier($login,$mdp)** :

Cette fonction vérifie la validité des informations de connexion. Elle compare le login et le mot de passe fournis par l'utilisateur avec ceux présents dans le fichier admin.json. Si les informations correspondent, l'accès est autorisé.

**function estAdmin()** Vérifie si l'utilisateur est un administrateur connecté. Elle retourne true si l'administrateur a validé ses informations de connexion et est autorisé à visualiser les inscriptions.

Le contrôleur vérifie si le bouton **Valider** a été cliqué.

Si l'administrateur a saisi les bonnes informations et cliqué sur **Valider**, il est redirigé vers la page de la liste des inscrits.

Si les informations ne sont pas valides ou si le bouton n'a pas été cliqué, l'administrateur reste sur la page de connexion pour essayer à nouveau.

#### Etape 2 : Visualisation des Inscriptions

Une image contenant capture d’écran, texte, logiciel, Icône d’ordinateur

Description générée automatiquement

**function donnerToutesLesConferences()** Cette fonction retourne un tableau contenant toutes les conférences disponibles. Chaque conférence inclut des informations telles que l'identifiant, le nom, le lieu et le nombre de participants inscrits.

**function donnerParticipants($numConference)** Cette fonction retourne la liste de tous les participants inscrits à une conférence spécifique, identifiée par son numéro (numConference). Les informations des participants incluent leur nom, prénom, ville, profession, et éventuellement d'autres détails pertinents.

Le contrôleur vérifie d'abord si l'utilisateur connecté est un administrateur à l'aide de la fonction estAdmin().

Si l'utilisateur est bien un administrateur, le contrôleur appelle la fonction donnerToutesLesConferences() pour récupérer la liste complète des conférences.

Pour chaque conférence, l'administrateur peut également accéder à la liste des participants en utilisant la fonction donnerParticipants($numConference).

**Résumé Partie 2 :**

1. **Connexion Administrateur** : Validation des informations de connexion et vérification des droits d'accès.
2. **Visualisation des conférences** : Affichage de toutes les conférences disponibles.
3. **Consultation des participants** : Accès à la liste des participants inscrits pour chaque conférence.