# Documentation Technique du Projet Séminaire

#### **Sommaire**

- 1. Introduction
- 2. Structure du Projet
- 3. Détails des Dossiers et Fichiers
  - o 1. Controleur
  - o 2. Data
  - o 3. Fonctions
  - o 4. Vue

## Introduction

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 une séparation claire entre la logique, la gestion des données et l'affichage.

# Structure du Projet

Le projet est organisé en plusieurs dossiers principaux :

- controleur : Contient les fichiers qui gèrent la logique de l'application.
- data : Contient les fichiers JSON pour la gestion des données.
- fonctions : Regroupe les fonctions PHP pour accéder aux données et gérer la logique.
- vue : Contient les fichiers qui s'occupent de l'affichage des pages.

### Détails des Dossiers et Fichiers

#### 1. Controleur

Ce dossier contient les fichiers qui gèrent la logique de l'application, en contrôlant le flux de données entre les modèles et les vues.

- **choixconferences.php**: Gère la logique pour le choix des conférences auxquelles l'utilisateur souhaite participer.
- connexion.php : S'occupe de la gestion de la connexion des utilisateurs.
- **controleurPrincipal.php**: Point central de coordination qui décide quel contrôleur et quelle vue doivent être utilisés selon l'action demandée.
- deconnexion.php : Gère la déconnexion des utilisateurs.
- inscription.php : Gère l'inscription des utilisateurs au séminaire.
- **programme.php** : Gère la logique d'affichage du programme du séminaire.
- voirInscrit.php: Permet de voir les inscriptions des utilisateurs.

#### 2. Data

Ce dossier contient des fichiers JSON qui stockent les données nécessaires au fonctionnement de l'application.

- admin.json : Contient les informations concernant les administrateurs.
- professions.json: Stocke la liste des professions disponibles lors de l'inscription.
- **seminaire.json** : Contient les informations sur le séminaire, y compris les conférences et les détails.
- **seminaireSauve.json** : Sauvegarde les informations mises à jour concernant le séminaire.

## 3. Fonctions

Ce dossier regroupe des fichiers de fonctions qui facilitent la gestion et l'accès aux données.

Il est constitué de 2 fichiers, le premier :

- fonctionsAccesDonnees.php : Contient les fonctions permettant d'accéder aux données, notamment les lectures et écritures dans les fichiers JSON.

Qui contient ces fonctions-là:

- donnerToutesLesConferences() : Retourne toutes les 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.

- donnerParticipants(\$numConference): Retourne tous les participants inscrits à une conférence spécifique. Chaque ligne du tableau retourné contient le nom, le prénom, la profession, la ville et le mail d'un participant.
- **chargeJSONadmin()**: Transfère le contenu du fichier `admin.json` dans un objet contenant l'arborescence du fichier JSON.
- donnerLesProfessions(): Retourne toutes les professions possibles sous forme d'un tableau, en chargeant les données du fichier `professions.json`.
- donnerLesHeuresCreneaux() : Retourne la liste de tous les créneaux horaires sous forme d'un tableau.
- donnerLesConferences(\$heure): Retourne toutes les conférences commençant à une heure donnée sous forme d'un tableau, incluant l'id, la description, l'intervenant, la salle, et le nombre de places.
- sauverDonneesInscription(\$nom, \$prenom, \$mail, \$ville, \$profession): Sauvegarde les données d'inscription des participants dans un fichier JSON pour une gestion future.
- enregistre(\$lesChoix): Enregistre un participant et ses choix de conférences dans le fichier `seminaire.json`. Par exemple, `\$lesChoix` contient un tableau indiquant les conférences sélectionnées pour chaque créneau horaire.

#### Et le second fichier:

- fonctions Gestion.php : Contient des fonctions de gestion générales, comme la validation des entrées utilisateur et la logique commune de l'application.

# Qui contient ces fonctions:

- **verifier(\$login, \$mdp)**: Permet de vérifier le login et le mot de passe pour autoriser ou non l'accès à l'application. Si les informations correspondent à l'admin, une variable de session `\$\_SESSION['admin']` est chargée.
- **estAdmin()** : Retourne `true` si le visiteur est un administrateur connecté et autorisé à visualiser les inscriptions.
- verifierDonneesInscription(\$nom, \$prenom, \$mail, \$ville): Vérifie les données saisies lors de l'inscription. Les champs sont vérifiés pour être non vides et l'adresse e-mail est vérifiée pour avoir un format valide.
- **ajouterErreur(\$msg)** : Ajoute un message d'erreur à la liste des erreurs si une erreur est détectée lors de la saisie des données.
- donnerNbErreurs(): Retourne le nombre de messages d'erreurs de saisie.

- afficherErreurs(): Affiche toutes les erreurs de saisie sous forme de liste.

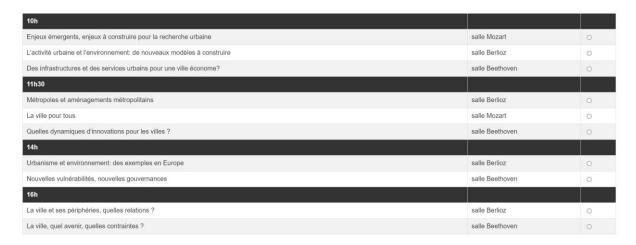
## 4. Vue

Les vues sont responsables de l'affichage des pages et de l'interaction avec l'utilisateur.

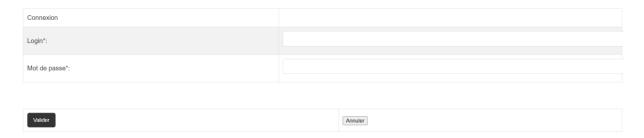
- **entete.php** : Fichier inclus au début de chaque page pour l'en-tête commune (menu de navigation, logo, etc.).



- pied.php: Fichier inclus à la fin de chaque page pour le pied de page commun.
- **vueChoixconferences.php**: Affiche le formulaire permettant aux utilisateurs de choisir les conférences auxquelles ils souhaitent participer.



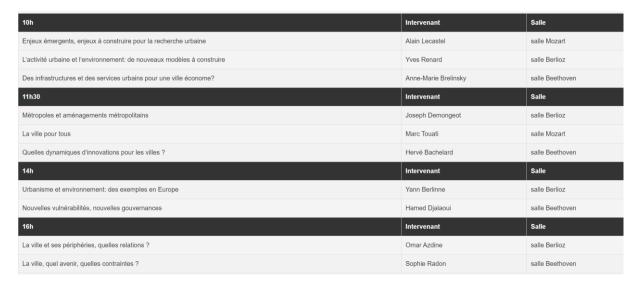
- vueConnexion.php: Formulaire de connexion pour les admin.



- vuelnscription.php: Formulaire d'inscription pour les nouveaux participants.

Nouvelle inscription			
Nom:			
Prenom:			
mail:			
Ville:			
Profession*:	Directeur de l'urbanisme		~
Valider		Annuler	

- vueProgramme.php : Affiche le programme du séminaire.



- **vueVoirInscription.php**: Permet aux utilisateurs admin de voir les inscrits aux conférences.

