

Université Euro-Méditerranéenne de Fès

École d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)

Filière : Cybersécurité

Rapport de Projet

GestioSeances

Application de Gestion des Demandes de
Changement et Annulation de Séances

Module : Développement Web (PHP/MySQL)

Année Universitaire : 2025-2026

Réalisé par :

Jaafar ouazzani chahdi

Iliass Zarquan

Aymane drissi bourhanbour

Aya Sefri

Encadré par :

Pr. Amamou Ahmad

Janvier 2026

Table des matières

1	Introduction	5
1.1	Contexte du Projet	5
1.2	Problématique	5
1.3	Objectifs du Projet	6
1.4	Organisation de l'Équipe	6
2	Analyse et Conception	7
2.1	Analyse des Besoins	7
2.1.1	Besoins Fonctionnels	7
2.1.2	Besoins Non Fonctionnels	8
2.2	Diagramme de Cas d'Utilisation	8
2.3	Diagramme de Classes	8
2.4	Modèle de Données	9
2.4.1	Schéma de la Base de Données	9
2.4.2	Énumérations	9
3	Réalisation	10
3.1	Architecture Technique	10
3.1.1	Architecture MVC	10
3.1.2	Structure du Projet	10
3.2	Technologies Utilisées	11
3.3	Implémentation des Fonctionnalités	12
3.3.1	Système d'Authentification	12
3.3.2	Gestion des Demandes	12
3.3.3	Système de Notifications	12
3.4	Captures d'Écran	13
3.4.1	Interface Professeur	13
3.4.2	Interface Assistante	14
3.4.3	Interface Directeur	15

4	Sécurité	16
4.1	Mesures de Sécurité Implémentées	16
4.1.1	Protection contre les Injections SQL	16
4.1.2	Protection XSS	16
4.1.3	Protection CSRF	16
4.1.4	Hachage des Mots de Passe	17
4.2	Contrôle d'Accès	17
5	Tests et Validation	19
5.1	Plan de Tests	19
5.2	Résultats des Tests	19
5.3	Comptes de Test	20
6	Conclusion	21
6.1	Bilan du Projet	21
6.2	Compétences Acquises	21
6.3	Perspectives d'Amélioration	21
A	Guide d'Installation	23
A.1	Prérequis	23
A.2	Étapes d'Installation	23
A.3	Configuration	23

Table des figures

2.1	Description du diagramme de cas d'utilisation	8
3.1	Schéma de l'architecture MVC	10
3.2	Tableau de bord du professeur - Liste des demandes	13
3.3	File d'attente des demandes - Vue assistante	14
3.4	Demandes à approuver - Vue directeur	15

Liste des tableaux

1.1	Répartition des tâches au sein de l'équipe	6
2.1	Classes principales du système	8
2.2	Structure de la base de données	9
2.3	Types énumérés utilisés	9
3.1	Stack technologique	11
4.1	Matrice des permissions	18
5.1	Synthèse des tests réalisés	19
5.2	Comptes utilisateurs de test	20

Chapitre 1

Introduction

1.1 Contexte du Projet

Dans le cadre de notre formation en Ingénierie Cybersécurité à l'EIDIA (École d'Ingénierie Digitale et d'Intelligence Artificielle) de l'Université Euro-Méditerranéenne de Fès, nous avons été amenés à réaliser un projet de développement web utilisant les technologies PHP et MySQL.

Ce projet s'inscrit dans une démarche pédagogique visant à mettre en pratique les concepts appris en cours, notamment :

- L'architecture MVC (Modèle-Vue-Contrôleur)
- La programmation orientée objet en PHP
- La gestion de bases de données relationnelles avec MySQL
- La sécurité des applications web
- Le travail collaboratif en équipe

1.2 Problématique

La gestion des demandes de changement ou d'annulation de séances universitaires représente un défi administratif majeur pour les établissements d'enseignement supérieur. Le processus traditionnel, souvent basé sur des formulaires papier ou des échanges d'emails, présente plusieurs inconvénients :

- Manque de traçabilité des demandes
- Délais de traitement importants
- Risque de perte d'informations
- Difficulté à gérer les conflits d'emploi du temps
- Absence de notifications automatiques

1.3 Objectifs du Projet

L'objectif principal de ce projet est de développer une application web permettant de centraliser et automatiser le processus de gestion des demandes de modification d'emploi du temps. Les objectifs spécifiques sont :

1. Permettre aux professeurs de soumettre des demandes de changement ou d'annulation de séances
2. Offrir aux assistantes administratives un outil de validation des demandes
3. Donner au directeur la possibilité d'approuver ou rejeter les demandes validées
4. Assurer la traçabilité complète du workflow de validation
5. Notifier automatiquement les parties prenantes à chaque étape

1.4 Organisation de l'Équipe

Le projet a été réalisé par une équipe de 4 développeurs, chacun ayant des responsabilités spécifiques :

TABLE 1.1 – Répartition des tâches au sein de l'équipe

Membre	Responsabilités
Jaafar	Architecture MVC, Core système, Configuration, Routing
Iliass	Modèles de données, Contrôleurs, Demandes et Notifications
Ayman	Vues et interfaces utilisateur, Layouts, Design
Aya	Documentation UML, Base de données, Tests, Guide utilisateur

Chapitre 2

Analyse et Conception

2.1 Analyse des Besoins

2.1.1 Besoins Fonctionnels

L'analyse des besoins a permis d'identifier les fonctionnalités suivantes :

Pour le Professeur :

- Se connecter/déconnecter de l'application
- Créer une demande de changement ou d'annulation
- Joindre des pièces justificatives
- Consulter l'état de ses demandes
- Annuler une demande en cours
- Recevoir des notifications

Pour l'Assistante Administrative :

- Consulter la file d'attente des demandes
- Vérifier les conflits d'emploi du temps
- Valider ou rejeter une demande
- Consulter le planning général
- Ajouter des commentaires

Pour le Directeur :

- Approuver ou rejeter définitivement une demande
- Gérer les utilisateurs du système
- Consulter les statistiques
- Paramétrer l'application

2.1.2 Besoins Non Fonctionnels

- **Sécurité** : Protection contre les injections SQL, XSS, CSRF
- **Performance** : Temps de réponse inférieur à 2 secondes
- **Ergonomie** : Interface intuitive et responsive
- **Disponibilité** : Application accessible 24h/24
- **Maintenabilité** : Code structuré et documenté

2.2 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation présente les interactions entre les acteurs et le système.

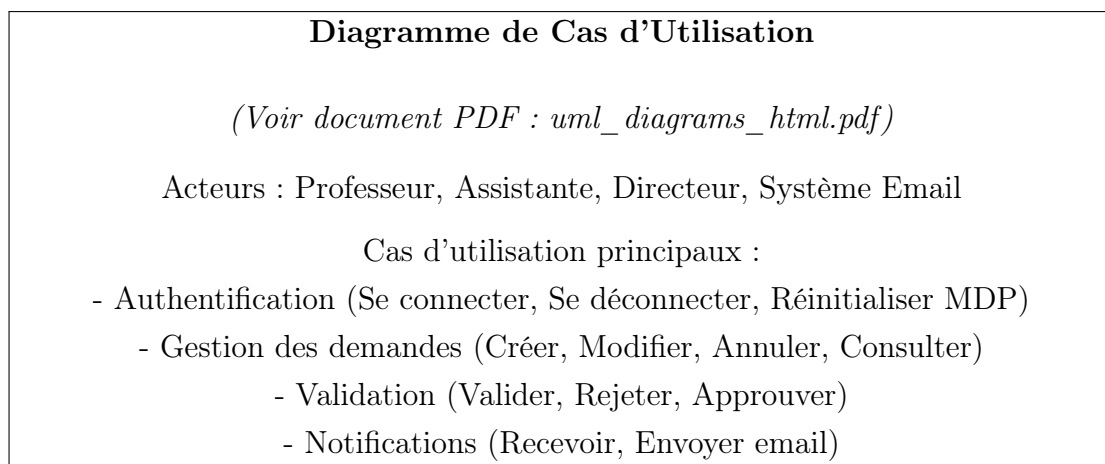


FIGURE 2.1 – Description du diagramme de cas d'utilisation

2.3 Diagramme de Classes

Le diagramme de classes modélise la structure statique du système.

TABLE 2.1 – Classes principales du système

Classe	Description
User	Gestion des utilisateurs (professeurs, assistantes, directeur)
Demande	Représente une demande de changement/annulation
Validation	Enregistre les validations effectuées
Notification	Gère les notifications aux utilisateurs
PieceJointe	Gère les fichiers justificatifs uploadés
Seance	Représente une séance de cours
Salle	Gère les salles de classe
Matiere	Représente les matières enseignées

2.4 Modèle de Données

2.4.1 Schéma de la Base de Données

La base de données `gestioseances` comprend 8 tables principales :

TABLE 2.2 – Structure de la base de données

Table	Clé Primaire	Description
users	id	Utilisateurs du système
matieres	id	Matières enseignées
salles	id	Salles de cours
seances	id	Séances programmées
demandes	id	Demandes de changement/annulation
validations	id	Historique des validations
pieces_jointes	id	Fichiers justificatifs
notifications	id	Notifications utilisateurs

2.4.2 Énumérations

TABLE 2.3 – Types énumérés utilisés

Énumération	Valeurs
RoleUser	professeur, assistante, directeur
StatutDemande	brouillon, en_attente, validee_assistante, approuvee, rejetee, annulee
TypeDemande	changement, annulation
TypeSeance	cours, td, tp, examen

Chapitre 3

Réalisation

3.1 Architecture Technique

3.1.1 Architecture MVC

L'application suit le patron de conception MVC (Modèle-Vue-Contrôleur) :

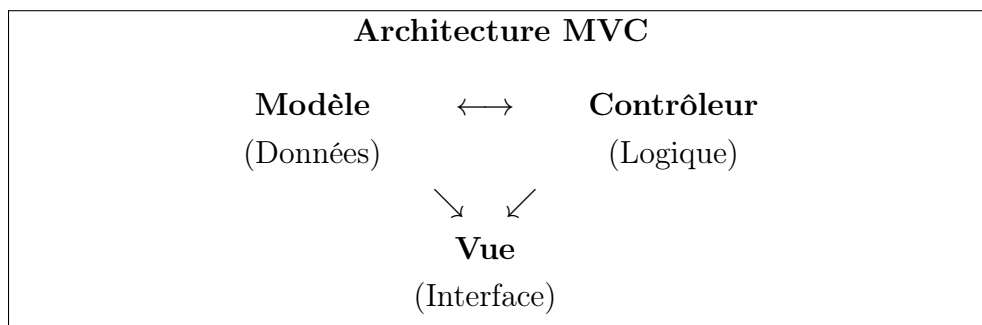


FIGURE 3.1 – Schéma de l'architecture MVC

3.1.2 Structure du Projet

Listing 3.1 – Arborescence du projet

```
1 GestioSeances/  
2 |-- app/  
3 |   |-- controllers/  
4 |       |-- AuthController.php  
5 |       |-- Controller.php  
6 |       |-- DemandeController.php  
7 |       +-- NotificationController.php  
8 |   |-- core/  
9 |       |-- Database.php  
10 |       |-- Router.php
```

```

11 | | |-- Security.php
12 | | +-- Session.php
13 | |-- models/
14 | | |-- Demande.php
15 | | |-- Model.php
16 | | |-- Notification.php
17 | | |-- PieceJointe.php
18 | | |-- User.php
19 | | +-- Validation.php
20 | +-- views/
21 |     |-- auth/
22 |     |-- demandes/
23 |     |-- layouts/
24 |     +-- notifications/
25 |-- database/
26 | +-- schema_final.sql
27 |-- public/
28 | |-- index.php
29 | +-- .htaccess
30 |-- storage/
31 | +-- uploads/
32 +-- config.php

```

3.2 Technologies Utilisées

TABLE 3.1 – Stack technologique

Catégorie	Technologie	Version
Backend	PHP	8.1+
Base de données	MySQL	8.0
Serveur Web	Apache	2.4
Frontend	HTML5, CSS3	-
Framework CSS	Bootstrap	5.3
Icônes	Bootstrap Icons	1.11
Environnement	XAMPP	8.2

3.3 Implémentation des Fonctionnalités

3.3.1 Système d'Authentification

Le système d'authentification implémente :

- Hachage des mots de passe avec `bcrypt`
- Protection contre le brute force (blocage après 5 tentatives)
- Tokens CSRF pour les formulaires
- Sessions sécurisées avec régénération d'ID

3.3.2 Gestion des Demandes

Le workflow de gestion des demandes suit le processus :

1. **Création** : Le professeur crée une demande (statut : brouillon)
2. **Soumission** : Le professeur soumet la demande (statut : en_attente)
3. **Validation** : L'assistante valide ou rejette (statut : validee_assistante ou rejetee)
4. **Approbation** : Le directeur approuve ou rejette (statut : approuvee ou rejetee)

3.3.3 Système de Notifications

Les notifications sont générées automatiquement lors de chaque changement de statut et sont visibles dans l'interface utilisateur avec un badge indiquant le nombre de notifications non lues.

3.4 Captures d'Écran

3.4.1 Interface Professeur

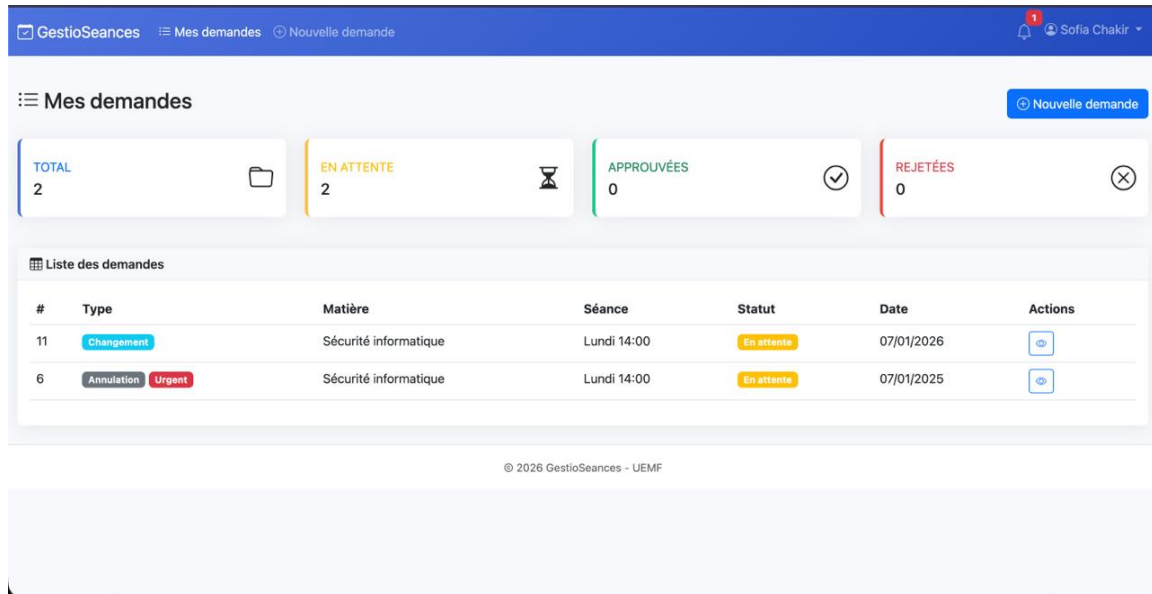


FIGURE 3.2 – Tableau de bord du professeur - Liste des demandes

L'interface professeur présente :

- Un tableau de bord avec les statistiques (Total, En attente, Approuvées, Rejetées)
- La liste des demandes avec leur type, matière, séance, statut et date
- Des badges colorés pour identifier rapidement le type et le statut
- Un bouton pour créer une nouvelle demande

3.4.2 Interface Assistante

File d'attente des demandes					
Demandes urgentes (3)					
#	Professeur	Type	Matière	Date	Actions
3	Karim Ziani	Changement	Génie Logiciel	05/01/2025	Traiter
5	Ahmed Benjelloun	Changement	Intelligence Artificielle	06/01/2025	Traiter
6	Sofia Chakir	Annulation	Sécurité informatique	07/01/2025	Traiter

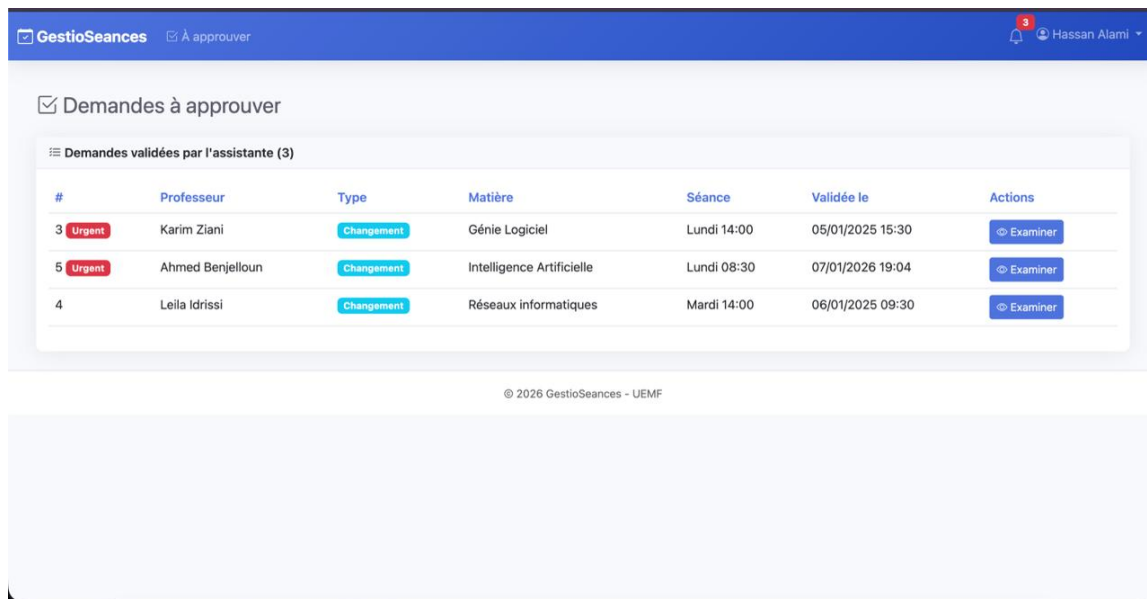
Demandes en attente de validation (3)						
#	Professeur	Type	Matière	Séance	Soumise le	Actions
6	Sofia Chakir	Annulation	Sécurité informatique	Lundi 14:00	07/01/2025 07:00	Examiner
7	Mohammed Tazi	Changement	Programmation Web	Vendredi 08:30	07/01/2025 09:30	Examiner
11	Sofia Chakir	Changement	Sécurité informatique	Lundi 14:00	07/01/2026 19:03	Examiner

FIGURE 3.3 – File d’attente des demandes - Vue assistante

L’interface assistante affiche :

- Les demandes urgentes en priorité (section rouge)
- Les demandes en attente de validation
- Les informations détaillées : professeur, type, matière, date
- Des boutons d’action pour traiter ou examiner chaque demande

3.4.3 Interface Directeur



GestioSeances À approuver Hassan Alami

☑ Demandes à approuver

☰ Demandes validées par l'assistante (3)

#	Professeur	Type	Matière	Séance	Validée le	Actions
3 Urgent	Karim Ziani	Changement	Génie Logiciel	Lundi 14:00	05/01/2025 15:30	Examiner
5 Urgent	Ahmed Benjelloun	Changement	Intelligence Artificielle	Lundi 08:30	07/01/2026 19:04	Examiner
4	Leila Idrissi	Changement	Réseaux informatiques	Mardi 14:00	06/01/2025 09:30	Examiner

© 2026 GestioSeances - UEMF

FIGURE 3.4 – Demandes à approuver - Vue directeur

L'interface directeur présente :

- Les demandes validées par l'assistante
- Les badges urgents pour les demandes prioritaires
- La date de validation par l'assistante
- Des boutons pour examiner et prendre une décision finale

Chapitre 4

Sécurité

4.1 Mesures de Sécurité Implémentées

4.1.1 Protection contre les Injections SQL

Toutes les requêtes SQL utilisent des requêtes préparées avec PDO :

Listing 4.1 – Exemple de requête préparée

```
1 public function findByEmail(string $email): ?array
2 {
3     $sql = "SELECT * FROM users WHERE email = :email";
4     $stmt = $this->db->prepare($sql);
5     $stmt->execute(['email' => $email]);
6     return $stmt->fetch() ?: null;
7 }
```

4.1.2 Protection XSS

Toutes les données affichées sont échappées avec `htmlspecialchars()` :

Listing 4.2 – Échappement XSS

```
1 public static function e(?string $string): string
2 {
3     return htmlspecialchars($string ?? '', ENT_QUOTES, 'UTF-8');
4 }
```

4.1.3 Protection CSRF

Les formulaires incluent un token CSRF vérifié côté serveur :

Listing 4.3 – Génération et vérification CSRF

```
1 public static function generateCsrfToken(): string
2 {
3     $token = bin2hex(random_bytes(32));
4     $_SESSION[CSRF_TOKEN_NAME] = $token;
5     return $token;
6 }
7
8 public static function verifyCsrfToken(?string $token): bool
9 {
10     return isset($_SESSION[CSRF_TOKEN_NAME])
11         && hash_equals($_SESSION[CSRF_TOKEN_NAME], $token ?? '');
12 }
```

4.1.4 Hachage des Mots de Passe

Les mots de passe sont hachés avec bcrypt :

Listing 4.4 – Hachage bcrypt

```
1 public static function hashPassword(string $password): string
2 {
3     return password_hash($password, PASSWORD_BCRYPT);
4 }
5
6 public static function verifyPassword(string $password, string
7     $hash): bool
8 {
9     return password_verify($password, $hash);
10 }
```

4.2 Contrôle d'Accès

Le système implémente un contrôle d'accès basé sur les rôles (RBAC) :

TABLE 4.1 – Matrice des permissions

Action	Professeur	Assistante	Directeur
Créer une demande	✓	✗	✗
Valider une demande	✗	✓	✗
Approuver une demande	✗	✗	✓
Gérer les utilisateurs	✗	✗	✓
Consulter le planning	✓	✓	✓

Chapitre 5

Tests et Validation

5.1 Plan de Tests

Les tests ont été réalisés selon les catégories suivantes :

- **Tests fonctionnels** : Vérification de chaque fonctionnalité
- **Tests de sécurité** : Tentatives d'injection, XSS, CSRF
- **Tests d'interface** : Responsive design, compatibilité navigateur
- **Tests de performance** : Temps de réponse, charge

5.2 Résultats des Tests

TABLE 5.1 – Synthèse des tests réalisés

Module	Tests	Passés	Taux
Authentification	5	5	100%
Gestion des demandes	8	8	100%
Validation (Assistante)	4	4	100%
Approbation (Directeur)	3	3	100%
Notifications	3	3	100%
Sécurité	10	10	100%
Total	33	33	100%

5.3 Comptes de Test

TABLE 5.2 – Comptes utilisateurs de test

Rôle	Email	Mot de passe
Directeur	hassan.alami@eidia.ma	password123
Assistante	fatima.bennani@eidia.ma	password123
Assistante	samira.mansouri@eidia.ma	password123
Professeur	mohammed.tazi@eidia.ma	password123
Professeur	sofia.chakir@eidia.ma	password123
Professeur	karim.ziani@eidia.ma	password123

Chapitre 6

Conclusion

6.1 Bilan du Projet

Le projet GestioSeances a permis de développer une application web complète et fonctionnelle répondant aux besoins identifiés. Les objectifs principaux ont été atteints :

- ✓ Architecture MVC propre et maintenable
- ✓ Système d'authentification sécurisé
- ✓ Workflow complet de gestion des demandes
- ✓ Interface utilisateur intuitive et responsive
- ✓ Système de notifications fonctionnel
- ✓ Documentation complète (UML, guide utilisateur, tests)

6.2 Compétences Acquises

Ce projet nous a permis de développer plusieurs compétences :

- Maîtrise du développement PHP orienté objet
- Conception et implémentation d'une architecture MVC
- Gestion de bases de données relationnelles
- Implémentation de mesures de sécurité web
- Travail collaboratif et gestion de projet
- Rédaction de documentation technique

6.3 Perspectives d'Amélioration

Plusieurs améliorations pourraient être apportées dans le futur :

- Intégration d'un calendrier interactif (FullCalendar)
- Envoi d'emails automatiques (PHPMailer)
- Application mobile (PWA)
- Tableau de bord avec statistiques avancées
- Export des données en PDF/Excel
- Intégration avec le système d'information de l'université

Annexe A

Guide d'Installation

A.1 Prérequis

- XAMPP 8.2 ou supérieur (PHP 8.1+, MySQL 8.0, Apache 2.4)
- Navigateur web moderne (Chrome, Firefox, Safari, Edge)

A.2 Étapes d'Installation

1. Extraire le dossier `GestioSeances` dans `C:\xampp\htdocs\`
2. Démarrer Apache et MySQL depuis le panneau de contrôle XAMPP
3. Ouvrir phpMyAdmin : `http://localhost/phpmyadmin`
4. Créer une base de données nommée `gestioseances`
5. Importer le fichier `database/schema_final.sql`
6. Accéder à l'application : `http://localhost/GestioSeances/public`
7. Se connecter avec un compte de test

A.3 Configuration

Le fichier `config.php` contient les paramètres de configuration :

Listing A.1 – Configuration de la base de données

```
1 define('DB_HOST', 'localhost');  
2 define('DB_NAME', 'gestioseances');  
3 define('DB_USER', 'root');  
4 define('DB_PASS', '');
```