Projet Soutenance Cahier de charge

Cahier des Charges – Application Web de Gestion des Étudiants

(Version enrichie : Déploiement Internet + Sécurité)

---

1. Introduction

1.1 Contexte

Le projet consiste à développer une application web destinée à gérer de manière centralisée les informations des étudiants. La version initiale tourne en local via WAMP. Cependant, l'objectif final est de la rendre accessible **en ligne**, pour permettre une utilisation à distance par les administrateurs autorisés.

1.2 Objectifs

- Gestion complète des étudiants via une interface web.
- Accès sécurisé pour les administrateurs.
- Application modulable pouvant évoluer vers un usage national ou institutionnel.
- Possibilité de déploiement sur un hébergement Internet (mode production).

---

2. Description du projet

2.1 Fonctionnalités principales

- Connexion administrateur.
- Tableau de bord statistique.
- CRUD complet des étudiants.
- Gestion des administrateurs (facultatif au lancement).
- Système de sessions sécurisées.

---

3. Architecture générale

3.1 Architecture logicielle

- Front-end : HTML5, CSS3, Bootstrap.
- Back-end : PHP 7+ / PHP 8 (compatibles).
- Base de données : MySQL.
- Serveur de développement : WAMP.
- Serveur de production (Internet) :
- Hébergement mutualisé ou VPS (selon budget).
- Apache / Nginx + PHP.
- MySQL en production.

---

4. Architecture de la base de données

Tables principales

- utilisateurs
- etudiants
- Tables futures possibles : classes, paiements, historique_actions, etc.

---

5. Modes de déploiement

5.1 Déploiement local (déjà en place)

- Utilisation de WAMP.
- Projet accessible via http://localhost/gestion_etudiants/.
- Base MySQL locale et fichiers PHP sur le disque.

5.2 Déploiement sur Internet

Deux approches possibles :

A) Hébergement mutualisé (facile, pas cher)

Exemples :

- Hostinger
- OVH
- LWS
- IONOS

Étapes :

1. Acheter un nom de domaine (ex : gestion-ecole.com).
2. Acheter un hébergement PHP/MySQL.
3. Importer la base MySQL locale vers MySQL en ligne via phpMyAdmin.
4. Uploader les fichiers PHP via FTP (FileZilla).
5. Modifier db.php avec les nouveaux identifiants MySQL distants.
6. Tester la connexion.

Avantages : facile, prend peu de temps.
Inconvénients : moins flexible, sécurité limitée par l'hébergeur.

———

B) Déploiement sur un VPS (professionnel, très sécurisé)

Exemples de VPS :

- DigitalOcean
- Contabo
- OVH Cloud
- Hetzner

Étapes :

1. Installer un serveur Linux (Ubuntu Server).
2. Configurer Apache ou Nginx.
3. Installer PHP et MySQL.
4. Configurer un pare-feu (UFW).
5. Installer un certificat SSL (Let's Encrypt).
6. Déployer l'application via Git ou SFTP.
7. Sécuriser MySQL (accès interdit depuis Internet).
8. Activer les journaux d'accès et erreurs.

Avantages : sécurité élevée, grande évolutivité.
Inconvénients : configuration plus technique.

———

6. Exigences de sécurité

6.1 Sécurité des mots de passe

- **En développement : mot de passe en clair** (ce que tu utilises maintenant).
- **En production : obligatoire** de passer à :

- Hashage : password_hash()
- Vérification : password_verify()
- Politique de mot de passe strict (minimum 8 caractères, chiffres, majuscule/minuscule).

6.2 Sécurisation des sessions

- Sessions PHP avec :
- session_regenerate_id() après connexion.
- Durée limitée (15–30 minutes).
- Cookies protégés : httponly, secure, samesite.

6.3 Sécurité de la base de données

- Toutes les requêtes doivent utiliser **PDO ou MySQLi préparé** (ce que tu fais déjà).
- Aucune donnée utilisateur ne doit passer directement dans les requêtes.
- Accès MySQL limité strictement à localhost en production.

6.4 Sécurité réseau

Sur serveur en ligne :

- HTTPS obligatoire.
- Pare-feu activé (UFW ou iptables).
- Interdire l'accès SSH root.
- Sauvegardes automatiques quotidiennes de la base.

6.5 Sécurité applicative

- Protection CSRF (token).
- Validation des formulaires côté serveur.
- Échapper toute sortie avec htmlspecialchars() (déjà présent dans dashboard).
- Journaux d'erreurs activés uniquement en fichier (jamais affichés au public).

---

7. Evolutions possibles

- Ajout du module “Paiements”.
- Gestion PDF des bulletins.
- Historique des actions (logs).
- Accès pour les professeurs.
- Interface pour les étudiants.

---

8. Planning prévisionnel

(Identique mais rajout d’étape de déploiement)

Étape	Délai
Développement (local)	2 à 3 semaines
Tests et corrections	3 jours
Sécurisation	2 jours
Déploiement Internet	2 à 4 jours
Formation & documentation	2 jours

9. Livrables

- Code complet.
- Base MySQL.
- Documentation technique.
- Manuel utilisateur.
- Cahier des charges.
- Rapport final de soutenance.
- Version en ligne sécurisée.