SysPulse

Documentation Technique

Application web de supervision de parc informatique

Auteur : Matthieu Campagna

Version : 1.0

Date : Mai 2026

Technologie : PHP 8.2 — MySQL — Bootstrap 5

1. Présentation du projet

SysPulse est une application web développée en PHP selon le patron de conception MVC (Modèle-Vue-Contrôleur). Elle permet de catégoriser et d'afficher sous forme de tableaux les composants matériels et logiciels d'un réseau informatique.

L'accès à l'application est sécurisé par un système d'authentification avec gestion des rôles. Les données des postes informatiques sont stockées dans un fichier JSON.

1.1 Objectifs

- Centraliser les informations matérielles et logicielles du parc informatique
- Offrir une interface de consultation filtrée par salle et par recherche textuelle
- Gérer les comptes utilisateurs avec deux niveaux de rôle : utilisateur et admin
- Sécuriser les accès par authentification avec mots de passe hashés

1.2 Technologies utilisées

Technologie	Rôle
PHP 8.2	Langage serveur principal
MySQL / MariaDB	Base de données des utilisateurs
Bootstrap 5.3	Framework CSS (via CDN)
PDO	Accès sécurisé à la base de données
JSON	Stockage des données du parc informatique
XAMPP	Serveur local de développement (Apache)

2. Architecture MVC

Le projet suit une architecture MVC stricte. Les fichiers sont organisés en dossiers distincts selon leur rôle.

2.1 Structure des dossiers

Dossier / Fichier	Description
config/database.php	Connexion PDO à la base de données
model/model.php	Fonctions d'accès aux données (BDD)
controller/RegisterController.php	Traitement de la création de compte
controller/EditController.php	Traitement de la modification de compte
controller/rss.php	Récupération du flux RSS ZATAZ
view/login.php	Page de connexion
view/index.php	Tableau de bord principal
view/hardware.php	Inventaire matériel
view/software.php	Inventaire logiciel
view/register.php	Formulaire de création de compte
view/edit.php	Formulaire de modification de compte
view/logout.php	Déconnecte l'utilisateur
view/header.php	En-tête HTML commun
view/footer.php	Pied de page HTML commun
data.json	Données des postes informatiques
js/filter.js	Filtrage dynamique des tableaux
rss.xml	Flux RSS de l'application

2.2 Flux de fonctionnement

- L'utilisateur accède à login.php via son navigateur
- Le formulaire envoie les données en POST vers login.php
- getLogin() dans model.php vérifie les credentials via PDO
- En cas de succès, la session est initialisée et l'utilisateur est redirigé vers index.php
- Chaque page protégée vérifie la présence de $_SESSION['user_id']
- Les contrôleurs traitent les actions (création, modification) puis redirigent

3. Base de données

La base de données bdsyspulse contient une seule table : utilisateurs.

3.1 Table utilisateurs

Champ	Type	Description
id	varchar(10)	Identifiant unique généré automatiquement
login	text	Identifiant de connexion
password	text	Mot de passe hashé (password_hash)
nom	text	Nom de famille
prenom	text	Prénom
role	enum	Rôle : utilisateur ou admin

3.2 Connexion PDO — config/database.php

Le fichier database.php établit la connexion à la base via PDO avec le charset UTF-8.

```
$host     = 'localhost';
$dbname   = 'bdsyspulse';
$username = 'root';
$password = '';
$pdo = new PDO("mysql:host=$host;dbname=$dbname;charset=utf8",
$username, $password);
```

3.3 Génération de l'identifiant utilisateur

L'identifiant est généré automatiquement dans RegisterController.php :

```
$id_genere = strtolower(substr($nom, 0, 1)) . rand(0, 999);
```

Exemple : pour le nom "Dupont", l'id pourrait être "d317".

4. Modèle — model.php

Le fichier model.php regroupe toutes les fonctions d'accès à la base de données.

4.1 getLogin()

Authentifie un utilisateur. Cherche le login en BDD puis vérifie le mot de passe hashé avec password_verify().

Paramètre	Description
$_POST['login']	Identifiant saisi dans le formulaire
$_POST['password']	Mot de passe saisi dans le formulaire

En cas de succès : initialise la session et redirige vers index.php.

En cas d'échec : affiche une alerte Bootstrap "Identifiants incorrects".

4.2 getRegister()

Enregistre un nouvel utilisateur. Vérifie d'abord que le login n'existe pas déjà.

Paramètre	Type
$id_genere	string
$prenom	string
$nom	string
$login	string
$password	string (hashé)

Retourne true si l'insertion réussit, false si le login est déjà pris.

4.3 getUsers()

Retourne la liste de tous les utilisateurs triés par nom (ORDER BY nom ASC).

4.4 getUserById()

Retourne les données d'un utilisateur selon son identifiant unique.

4.5 updateUser()

Met à jour les informations d'un utilisateur existant (prénom, nom, login, password hashé, rôle).

5. Contrôleurs

5.1 RegisterController.php

Traite la création d'un nouveau compte utilisateur.

- Vérifie que l'utilisateur connecté a le rôle admin
- Récupère les données POST du formulaire
- Hash le mot de passe avec password_hash()
- Génère un identifiant unique
- Appelle getRegister() et redirige avec ?success=1 ou ?error=1

5.2 EditController.php

Traite la modification d'un compte utilisateur existant.

- Vérifie que l'utilisateur connecté a le rôle admin
- Récupère les données POST du formulaire
- Hash le nouveau mot de passe avec password_hash()
- Appelle updateUser() et redirige avec ?success=1 ou ?error=1

5.3 rss.php

Récupère les 4 derniers articles du flux RSS de ZATAZ (cybersécurité) et retourne un tableau avec titre, lien, description et image.

6. Vues

6.1 header.php

En-tête HTML commun à toutes les pages. Charge Bootstrap 5.3 via CDN.

6.2 footer.php

Pied de page HTML commun. Affiche le copyright avec lien vers le portfolio de l'auteur.

6.3 login.php

Page de connexion accessible à l'URL : http://localhost/SysPulse/view/login.php

- Affiche le logo, la bio du projet et le formulaire de connexion
- Affiche les 4 dernières actualités cybersécurité ZATAZ (avec image, titre, description)
- Appelle getLogin() pour traiter la soumission du formulaire

6.4 index.php

Tableau de bord principal après connexion. Accès protégé par session.

- Affiche le prénom et nom de l'utilisateur connecté
- Liens vers hardware.php et software.php
- Boutons Créer un compte et Modifier un compte visibles uniquement pour le rôle admin
- Bouton de déconnexion

6.5 hardware.php

Inventaire matériel du parc informatique. Données lues depuis data.json.

Colonne	Source JSON	Description
Poste	nomDuPoste	Nom du poste informatique
Emplacement	emplacement	Salle (Info1, Info2, Info3)
CPU	hardware.cpu	Processeur
RAM	hardware.ram	Mémoire vive en MB
GPU	hardware.gpu	Carte graphique
Disques	hardware.lecteurs	Nom, espace libre / total en MB

Filtrage dynamique via filter.js : recherche textuelle + cases à cocher par salle.

6.6 software.php

Inventaire logiciel du parc informatique. Données lues depuis data.json.

Colonne	Source JSON	Description
Poste	nomDuPoste	Nom du poste informatique
Emplacement	emplacement	Salle
Adresse MAC	software.adresseMac	Adresse MAC réseau
Adresse IP	software.adresseIp	Adresse IP réseau
OS	software.osName	Nom du système d'exploitation
Version OS	software.osVersion	Version du système d'exploitation

6.7 register.php

Formulaire de création de compte. Accès réservé au rôle admin uniquement.

- Champs : Prénom, Nom, Login, Mot de passe
- Soumission en POST vers RegisterController.php
- Affiche un message de succès ou d'erreur selon le résultat

6.8 edit.php

Formulaire de modification d'un compte existant. Accès réservé au rôle admin.

- Liste déroulante pour sélectionner l'utilisateur à modifier
- Formulaire pré-rempli avec prénom, nom, login, rôle
- Le champ mot de passe est vide (ne jamais afficher le hash)
- Soumission en POST vers EditController.php

7. Sécurité

7.1 Hashage des mots de passe

Les mots de passe sont hashés avec l'algorithme bcrypt via les fonctions natives PHP :

Fonction PHP	Utilisation
password_hash($mdp, PASSWORD_DEFAULT)	À la création et modification de compte
password_verify($mdp, $hash)	À la connexion dans getLogin()

7.2 Protection des pages

Chaque page protégée vérifie la session en début de fichier :

```
if (!isset($_SESSION['user_id'])) {
    header('Location: ../view/login.php');
    exit;
}
```

7.3 Protection par rôle

Les pages d'administration (register.php, edit.php, contrôleurs) vérifient le rôle admin :

```
if (!isset($_SESSION['role']) || $_SESSION['role'] !== 'admin') {
    header('Location: ../view/index.php');
    exit;
}
```

7.4 Requêtes préparées PDO

Toutes les requêtes SQL utilisent des requêtes préparées avec paramètres liés (?), ce qui protège contre les injections SQL.

8. Flux RSS

8.1 RSS externe — ZATAZ

Le fichier controller/rss.php récupère les 4 derniers articles du flux RSS de ZATAZ (actualités cybersécurité française). Il est affiché sur la page de connexion.

Clé retournée	Description
titre	Titre de l'article
lien	URL de l'article
desc	Description courte (100 caractères max)
image	URL de l'image (fallback Unsplash si absente)

8.2 RSS interne — rss.xml

Le fichier rss.xml à la racine du projet est le flux RSS propre à SysPulse. Il décrit l'application et ses mises à jour. L'URL sera à mettre à jour avec le domaine réel lors de la mise en ligne.

9. Déploiement

9.1 Environnement local — XAMPP

Élément	Valeur
Serveur	Apache 2.4.58
PHP	8.2.12
Base de données	MariaDB 10.4.32
URL locale	http://localhost/SysPulse/view/login.php
Dossier projet	C:\xampp\htdocs\SysPulse\

9.2 Mise en ligne gratuite

Le projet utilise PHP et MySQL, incompatible avec GitHub Pages (statique uniquement). Options recommandées :

Hébergeur	Avantages
InfinityFree	PHP + MySQL gratuit, phpMyAdmin inclus
FreeHosting	PHP + MySQL gratuit, 10 Go
Railway	Déploiement via Docker, 500h/mois gratuit

9.3 Étapes de mise en ligne

- 1. Créer un compte sur l'hébergeur choisi
- 2. Créer la base de données et importer bdsyspulse.sql via phpMyAdmin
- 3. Mettre à jour config/database.php avec les credentials de l'hébergeur
- 4. Uploader les fichiers via FTP (FileZilla)
- 5. Mettre à jour l'URL dans rss.xml avec le vrai domaine
- 6. Soumettre le sitemap.xml dans Google Search Console

9.4 SEO

Un fichier sitemap.xml et un fichier robots.txt sont à placer à la racine pour référencer l'application. Les pages protégées (index, hardware, software, edit) sont exclues de l'indexation.

10. Résumé des fichiers

Fichier	Type	Rôle
config/database.php	Config	Connexion PDO MySQL
model/model.php	Modèle	CRUD utilisateurs
controller/RegisterController.php	Contrôleur	Création compte
controller/EditController.php	Contrôleur	Modification compte
controller/rss.php	Contrôleur	Flux RSS ZATAZ
view/header.php	Vue	En-tête HTML commun
view/footer.php	Vue	Pied de page commun
view/login.php	Vue	Page de connexion
view/index.php	Vue	Tableau de bord
view/hardware.php	Vue	Inventaire matériel
view/software.php	Vue	Inventaire logiciel
view/register.php	Vue	Création de compte
view/edit.php	Vue	Modification de compte
view/logout.php	Vue	Déconnecte l'utilisateur
data.json	Données	Parc informatique JSON
js/filter.js	Script	Filtrage tableaux
rss.xml	XML	Flux RSS SysPulse
sitemap.xml	XML	Plan du site SEO
bdsyspulse.sql	SQL	Structure BDD