

Julien MATHIAS  
Thomas BIEJ  
Rafael BEAULIEU  
Elijah FALLET  
Fares YAHIA CHERIF  
INFI-2-B



IUT de Vélizy-Rambouillet  
CAMPUS DE VÉLIZY-VILLACOUBLAY

# **Documentation**

# Sommaire

## 1. Analyse des besoins

- Diagramme de cas d'utilisation
- Accès et liaison logique entre pages
- Exigences fonctionnelles
- Cas d'utilisation par niveaux
- Exemples de cas d'utilisation détaillés

## 2. Gestion des équipements et des informations réutilisables

- Informations réutilisables (Constructeur, Système d'exploitation)
- Fonctionnement technique (Fichiers concernés)
- Mise à jour et ajout d'équipements

## 3. Mise en place et fonctionnalités du site web

- Création de la base de données
- Importation de données
- Fonctionnalités de gestion des équipements

## **Intro**

Ce document détaille le fonctionnement technique de l'application web de gestion de parc informatique. Il se concentre exclusivement sur les fonctionnalités requises pour les profils **Visiteur, Administrateur et Technicien**

Nous avons fait en sorte que l'administrateur puisse :

- Créer un technicien dans la base
- Créer une information qui peut être réutilisée par les techniciens, telles que :
  - Le nom des systèmes d'exploitation
  - Le constructeur de la machine

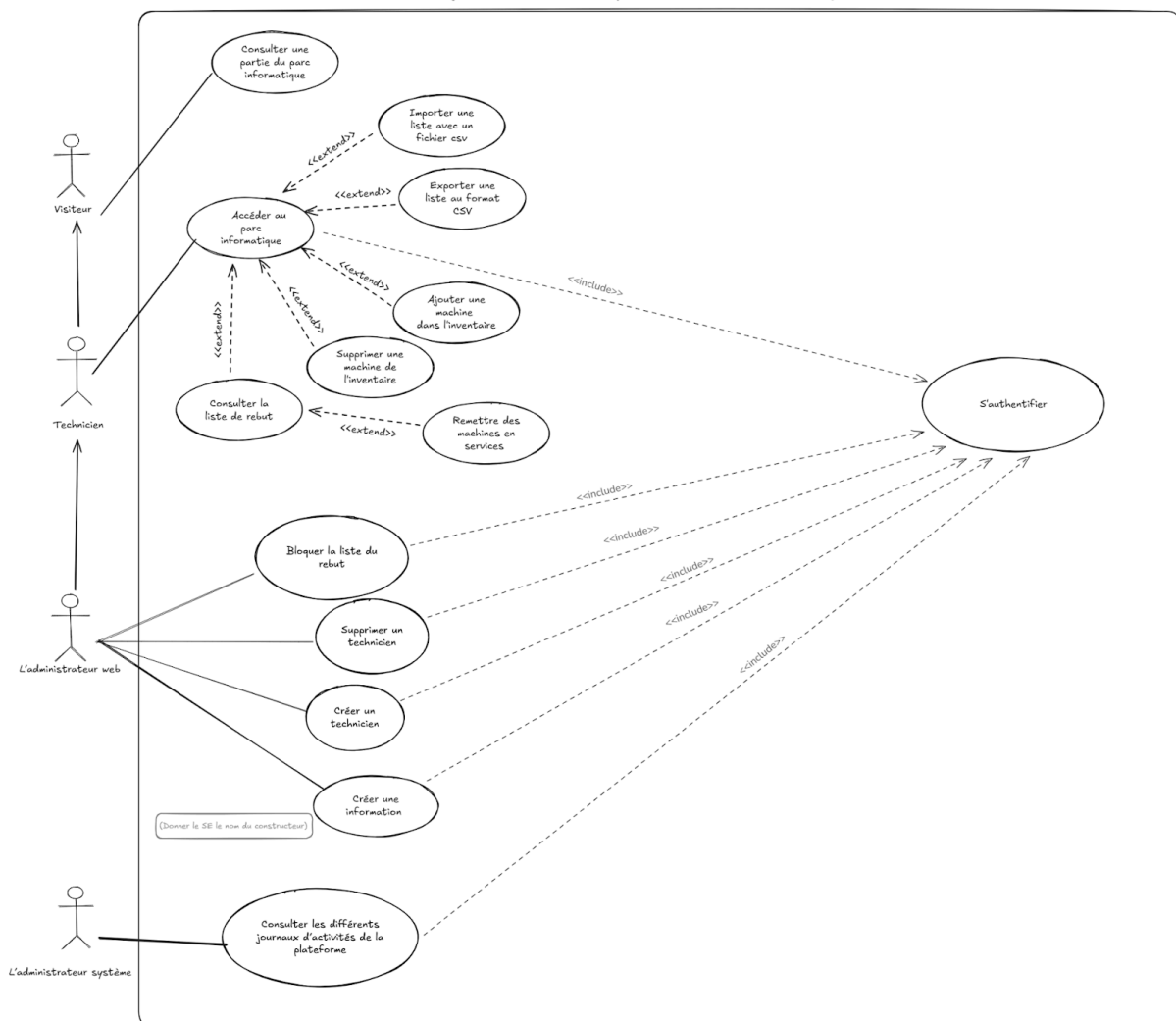
Nous avons fait en sorte que le technicien puisse :

- Consulter le parc informatique
- Modifier une information dans la liste du parc informatique
- Mettre une machine dans l'inventaire à partir d'un formulaire

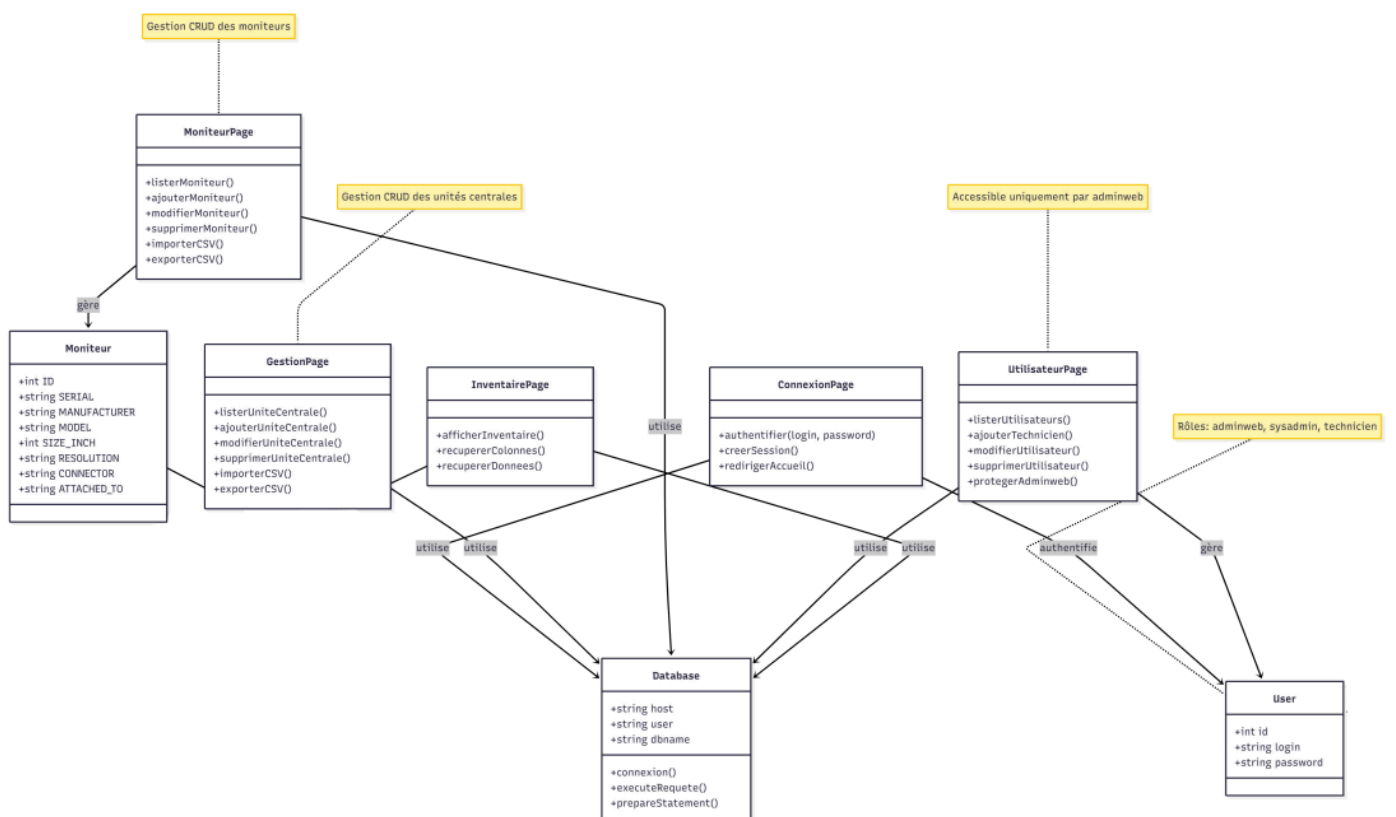
# Analyse des besoins

Pour mieux visualiser, nous avons réalisé le diagramme de cas d'utilisation pour définir les rôles et les actions de chaque utilisateur du système peut faire :

## Cas d'utilisation de gestion de parc informatique



Voici une description plus détaillée de l'accès de chaque utilisateur aux pages et de la liaison logique entre chaque page en schéma :



Nous avons rédigé les exigences fonctionnelles afin de définir précisément les fonctionnalités attendues du système, pour garantir qu'il réponde aux besoins spécifiques des utilisateurs et des administrateurs, tout en facilitant le développement et la mise en place des différentes fonctionnalités clés du projet.

## **1. Exigences fonctionnelles :**

### **1.1 Gestion de l'inventaire**

- Créer un matériel (UC ou écran)
- Consulter l'inventaire
- Mettre à jour un matériel
- Supprimer un matériel / envoyer au rebut
- Importer des matériels via fichier CSV
- Exporter l'inventaire au format CSV

### **1.2 Gestion des utilisateurs**

- Authentification selon rôle (Visiteur / Technicien / Admin web / Admin système)
- Création d'un technicien (Admin web)
- Suppression d'un technicien (Admin web)

### **1.3 Gestion des informations systèmes**

- Ajouter une information réutilisable (OS, constructeur)
- Consulter les différentes informations

## **1.4 Logs (administrateur système)**

- Consulter les journaux d'activité de la plateforme

Nous avons aussi structuré les cas d'utilisation en différentes catégories, organisées par niveau, afin de clarifier les objectifs et les responsabilités au sein du système.

### **Niveau stratégique (haut niveau) :**

Ce niveau regroupe les objectifs à long terme, permettant de définir la gestion globale du système :

1. Gérer le parc informatique
2. Gérer les utilisateurs techniques
3. Superviser la plateforme et les journaux système

### **Niveau utilisateur (objectifs utilisateur) :**

Ce niveau correspond aux fonctionnalités accessibles aux utilisateurs finaux, telles que les techniciens et administrateurs, pour gérer le parc informatique au quotidien :

1. Consulter l'inventaire
2. Ajouter un matériel
3. Modifier un matériel
4. Supprimer un matériel
5. Remettre en service un matériel
6. Importer un CSV
7. Exporter un CSV
8. S'authentifier
9. Gérer les techniciens

**Niveau sous-fonction :**

Ce niveau détaille les actions spécifiques et les processus internes qui soutiennent les fonctionnalités des niveaux supérieurs :

- S'authentifier
- Valider un formulaire
- Lire un fichier CSV
- Valider un matériel
- Écrire dans le fichier de log

Nous avons ensuite élaboré plusieurs cas d'utilisation afin de définir clairement les fonctionnalités principales du système. Ces cas d'utilisation couvrent des actions clés telles que l'authentification des utilisateurs, la consultation de l'inventaire, l'ajout de matériel et l'importation de fichiers CSV. Chaque cas d'utilisation a été conçu pour garantir une expérience fluide et sécurisée tout en respectant les rôles et permissions des utilisateurs, et en assurant la fiabilité du système. Ces scénarios sont documentés de manière détaillée afin de faciliter la compréhension des processus et de servir de base pour la mise en œuvre des fonctionnalités.



# CU 1 — S'authentifier

**Portée :** Système web de gestion du parc informatique (boîte noire)

**Niveau :** Sous-fonction

**Acteur principal :** Tout utilisateur (visiteur, technicien, admin web, admin système)

**Intervenants et intérêts :**

- **Utilisateur :** veut accéder aux fonctionnalités compatibles avec son rôle.
- **Université :** veut un accès sécurisé.
- **Système :** doit contrôler les accès et protéger les données.

**Préconditions :**

- Le système d'authentification est opérationnel.

**Garanties minimales :**

- Aucune session non autorisée n'est créée.
- Tentative d'accès enregistrée dans les logs.

**Garanties en cas de succès :**

- Une session associée au rôle utilisateur est active.
- L'utilisateur accède aux pages selon ses permissions.

**Déclencheur :**

- L'utilisateur saisit ses identifiants dans le formulaire de connexion.

**Scénario nominal :**

1. L'utilisateur accède à la page de connexion.

2. L'utilisateur saisit son identifiant et mot de passe.
3. Le système valide le format des données saisies.
4. Le système vérifie les identifiants dans la base.
5. Le système crée une session associée au rôle de l'utilisateur.
6. Le système redirige l'utilisateur vers son espace (visiteur, technicien, admin web, admin système).
7. Le système écrit dans les logs « connexion réussie ».

## **Extensions :**

### **2a — Identifiants manquants**

→ 2a1. Le système signale que tous les champs doivent être remplis. Retour à l'étape 1.

### **4a — Identifiants incorrects**

→ 4a1. Le système affiche un message d'erreur.  
→ 4a2. Le système demande de réessayer (max 3 tentatives).  
→ 4a3. Après 3 échecs, le système bloque temporairement le compte et inscrit l'événement dans les logs.

### **4b — Compte inexistant**

→ 4b1. Message : « utilisateur inconnu ». Retour à l'étape 1.

### **5a — Session impossible (erreur interne)**

→ 5a1. Le système affiche une erreur générique.  
→ 5a2. Le système enregistre l'erreur dans les logs.  
→ 5a3. Fin du CU par échec.

# CU 2 — Consulter l'inventaire

**Portée :** Système web

**Niveau :** Objectif utilisateur

**Acteur principal :** Visiteur ou Technicien

**Intervenants et intérêts :**

- **Visiteur** : souhaite visualiser une partie du parc.
- **Technicien** : souhaite visualiser l'intégralité du parc pour intervenir.
- **Université** : veut un suivi clair et fiable.

**Préconditions :**

- L'utilisateur est authentifié (sauf visiteur).

**Garanties minimales :**

- Aucune modification n'est effectuée.

**Garanties en cas de succès :**

- La liste des matériels est affichée selon les permissions du rôle.

**Déclencheur :**

- L'utilisateur clique sur « Inventaire ».

**Scénario nominal :**

1. L'utilisateur demande l'accès à l'inventaire.
2. Le système vérifie les droits d'accès.
3. Le système charge la liste des matériels depuis la base.
4. Le système affiche la liste des matériels avec les filtres éventuels.

## **Extensions :**

### **3a — Inventaire vide**

→ 3a1. Message : « Aucun matériel enregistré ».

### **3b — Erreur de connexion à la base**

→ 3b1. Le système affiche une erreur.

→ 3b2. Le système écrit l'erreur dans les logs.

→ 3b3. Fin du CU par échec.

### **4a — Filtre invalide**

→ 4a1. Le système ignore le filtre.

→ 4a2. Retour à l'affichage nominal.

### **4b — Recherche sans résultat**

→ 4b1. Message : « Aucun matériel ne correspond à votre recherche ».

# CU 3 — Ajouter un matériel

**Portée :** Système web

**Niveau :** Objectif utilisateur

**Acteur principal :** Technicien

**Intervenants et intérêts :**

- **Technicien :** veut enregistrer correctement le matériel.
- **Université :** doit disposer d'un inventaire fiable.
- **Système :** doit valider les données et écrire les logs.

**Préconditions :**

- Le technicien est authentifié.

**Garanties minimales :**

- Rien n'est ajouté si les données sont invalides.

**Garanties en cas de succès :**

- Le matériel est ajouté à l'inventaire.
- Une entrée est écrite dans les logs.

**Déclencheur :**

- Le technicien valide le formulaire de création.

**Scénario nominal :**

1. Le technicien ouvre le formulaire d'ajout.
2. Le technicien saisit toutes les informations obligatoires.
3. Le système valide le format des données.

4. Le système vérifie l'existence éventuelle du numéro de série.
5. Le système ajoute le matériel dans la base.
6. Le système confirme l'ajout.
7. Le système enregistre l'opération dans les logs.

## **Extensions :**

### **3a — Données invalides ou manquantes**

- 3a1. Message indiquant les champs incorrects.
- Retour à l'étape 1.

### **4a — Numéro de série déjà existant**

- 4a1. Message : « matériel déjà enregistré ».
- Retour à l'étape 1.

### **3b — OS ou constructeur manquant dans la base**

- 3b1. Le système propose d'en ajouter (CU : Créer une information).
- 3b2. Retour au scénario nominal.

### **5a — Erreur d'insertion en base**

- 5a1. Message d'erreur.
- 5a2. Log système.
- 5a3. Fin par échec.

# CU 4 — Importer un fichier CSV

**Portée :** Système web

**Niveau :** Objectif utilisateur

**Acteur principal :** Technicien

**Intervenants et intérêts :**

- **Technicien** : souhaite gagner du temps.
- **Université** : veut un inventaire complet et propre.
- **Système** : doit valider et traiter les lignes.

**Préconditions :**

- Le fichier CSV existe et suit la structure donnée.

**Garanties minimales :**

- Les lignes incorrectes ne sont pas importées.
- Tout incident est logué.

**Garanties en cas de succès :**

- Les matériels valides sont ajoutés.

**Déclencheur :**

- Le technicien charge un fichier CSV.

**Scénario nominal :**

1. Le technicien choisit le fichier CSV.
2. Le système vérifie l'extension et le format.

3. Le système lit le fichier ligne par ligne.
4. Pour chaque ligne valide, le système crée le matériel correspondant.
5. Le système affiche le nombre d'éléments importés.
6. Le système écrit l'ensemble des opérations dans les logs.

## **Extensions :**

### **2a — Mauvaise extension**

- 2a1. Message : « fichier non reconnu ».

### **3a — Ligne illisible**

- 3a1. La ligne est ignorée.
- 3a2. Message en fin d'import.

### **4a — Matériel déjà existant**

- 4a1. Le système ignore la ligne.
- 4a2. Note dans les logs.

### **4b — Données obligatoires manquantes**

- 4b1. Ignorer la ligne.

### **1a — Fichier trop volumineux**

- 1a1. Message d'erreur.
- 1a2. Fin par échec.

**Voici la gestion des équipements et des informations réutilisables dans l'application**



Les informations réutilisables sont gérés ainsi :

- **Fichiers concernés :** `gestion.php`, `moniteur.php`, `gestion_modifier.php`.
- **Fonctionnement technique :**
  - **Implémentation actuelle :** Dans le fichier `gestion.php`, les champs "MANUFACTURER" (Constructeur) et "OS" (Système d'exploitation) sont implémentés sous forme de champs texte libres (`<input type="text">`) lors de l'ajout ou de la modification, l'application peut soit accepter du texte libre, soit permettre de choisir parmi des options prédéfinies (options statiques "Option 1", "Option 2" dans le code actuel).

Le technicien a accès à une liste complète des équipements (Unités centrales et Moniteurs).

- **Fichiers concernés :** `gestion.php`, `moniteur.php`, `inventaire.php`.
- **Fonctionnement technique :**
  - **Récupération des données :** L'application établit une connexion à la base `rpiBD` et exécute une requête `SELECT * FROM inventaire` (pour les PC) ou `SELECT * FROM moniteur`.
  - **Affichage dynamique :** Une boucle `while ($ligne = mysqli_fetch_assoc($data))` parcourt les résultats pour générer dynamiquement les lignes (`<tr>`) d'un tableau HTML.
  - **Export/Import :** Le code inclut également des fonctionnalités pour exporter cette liste en CSV (`fputcsv`) ou importer des données depuis un CSV pour alimenter la consultation.

Le technicien peut mettre à jour les détails d'une machine existante.

- **Fichiers concernés :** `gestion.php`, `moniteur.php`.
- **Fonctionnement technique :**
  - **Pré-remplissage :** Lorsqu'on clique sur "Modifier", un formulaire est affiché et pré-rempli avec les données actuelles de la machine grâce à une requête `SELECT ... WHERE ID=$id`.

- **Mise à jour SQL** : Lors de la soumission (`mise_a_jour`), une requête `UPDATE` préparée met à jour l'ensemble des champs (RAM, OS, Localisation, etc.) pour l'ID ciblé.
- **Sécurité** : Utilisation de `intval()` sur les ID et de `mysqli_prepare` pour éviter les injections SQL.

Le technicien peut ajouter de nouveaux équipements via un formulaire.










- **Fichiers concernés** : `gestion.php`, `gestion_ajout.php`.
- **Fonctionnement technique** :
  - Le formulaire d'ajout récupère les caractéristiques techniques (CPU, RAM, Disque, Date d'achat, Garantie, etc.).
  - **Traitement** : Les données `$_POST` sont liées à des paramètres SQL (`bind_param`).
  - **Enregistrement** : La requête `INSERT INTO inventaire` crée la nouvelle entrée. Le code gère les types de données (chaînes de caractères vs entiers pour la RAM/Stockage).

## Mise en place et

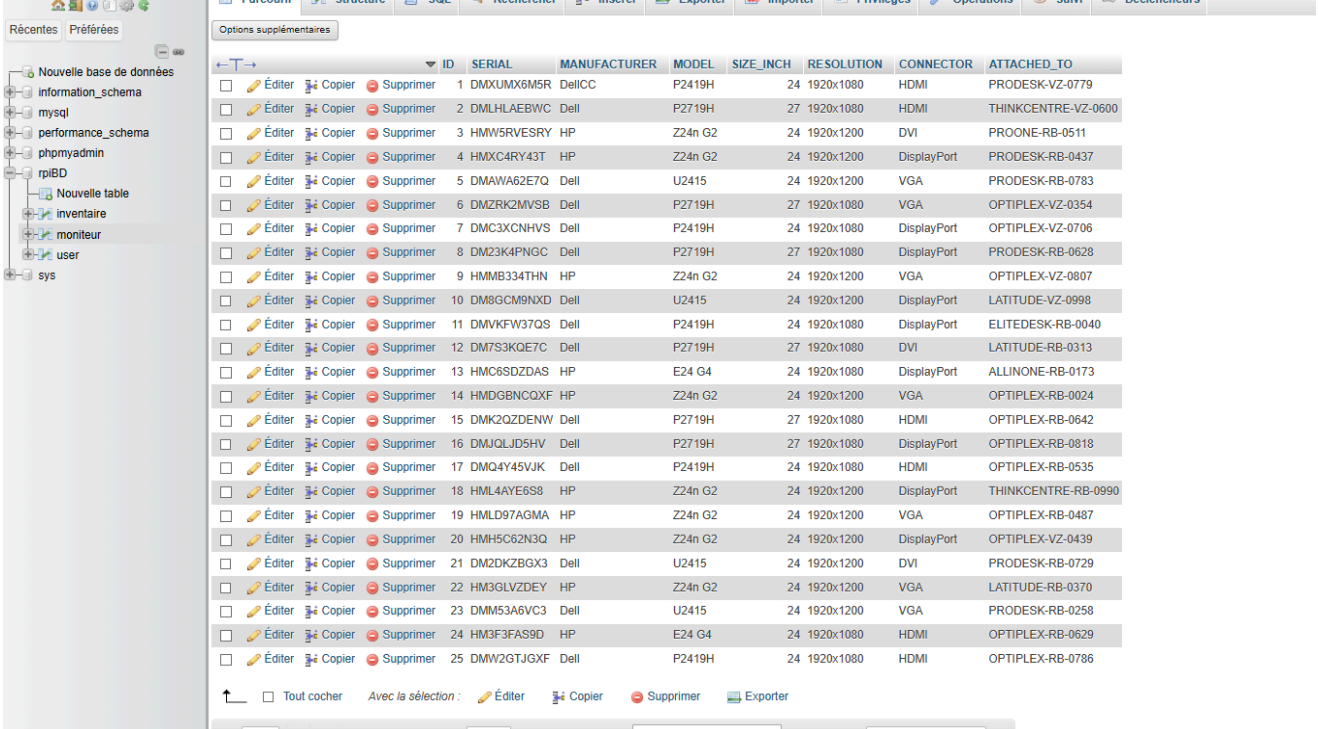
# fonctionnalités du site web

Voici comment créer la base de données, il vous suffit de vous connecter à phpmyadmin et exécuter les scripts de création de base de données “script\_à\_exécuter.php” qui se situe dans le dossier script, qui vont créer la base de données et les tables nécessaires au bon fonctionnement du site.

## Index of /script\_bd

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">creation_BD.php</a>	2025-12-05 17:39	393	
 <a href="#">creation_table_rebut_devices.php</a>	2025-12-05 17:39	879	
 <a href="#">creation_table_rebut_moniteur.php</a>	2025-12-05 17:39	699	
 <a href="#">creation_table_utilisateur.php</a>	2025-12-05 17:41	1.0K	
 <a href="#">inventory_devices.csv</a>	2025-12-05 17:39	173K	
 <a href="#">inventory_monitors2.csv</a>	2025-12-05 17:39	57K	
 <a href="#">script_creation_inventaire_devices.php</a>	2025-12-05 17:39	1.9K	
 <a href="#">script_creation_inventaire_monitors.php</a>	2025-12-05 17:39	2.2K	
 <a href="#">script_à_exécuter.php</a>	2025-12-05 17:39	317	

Voici le résultat dans rpi17/phpmyadmin toutes les tables seront créées:



The screenshot shows the phpMyAdmin interface with a table named 'moniteur' in the 'rpiBD' database. The table has 25 rows of data, each representing a monitor. The columns are: ID, SERIAL, MANUFACTURER, MODEL, SIZE\_INCH, RESOLUTION, CONNECTOR, and ATTACHED\_TO. Each row includes action icons for editing, copying, and deleting the record.

ID	SERIAL	MANUFACTURER	MODEL	SIZE_INCH	RESOLUTION	CONNECTOR	ATTACHED_TO
1	DMXUMX6M5R	Dell	P2419H	24	1920x1080	HDMI	PRODESK-VZ-0779
2	DMLHLAEBWC	Dell	P2719H	27	1920x1080	HDMI	THINKCENTRE-VZ-0600
3	HMW5RVESRY	HP	Z24n G2	24	1920x1200	DVI	PROONE-RB-0511
4	HMXC4RY43T	HP	Z24n G2	24	1920x1200	DisplayPort	PRODESK-RB-0437
5	DMAWA62E7Q	Dell	U2415	24	1920x1200	VGA	PRODESK-RB-0783
6	DMZRK2MVS	Dell	P2719H	27	1920x1080	VGA	OPTIPLEX-VZ-0354
7	DMC3KCNHVS	Dell	P2419H	24	1920x1080	DisplayPort	OPTIPLEX-VZ-0706
8	DM23K4PNGC	Dell	P2719H	27	1920x1080	DisplayPort	PRODESK-RB-0628
9	HMMB334THN	HP	Z24n G2	24	1920x1200	VGA	OPTIPLEX-VZ-0807
10	DM8GCM9N	Dell	U2415	24	1920x1200	DisplayPort	LATITUDE-VZ-0998
11	DMVKFW37QS	Dell	P2419H	24	1920x1080	DisplayPort	ELITEDESK-RB-0040
12	DM7S3KQETC	Dell	P2719H	27	1920x1080	DVI	LATITUDE-RB-0313
13	HMC6SDZDAS	HP	E24 G4	24	1920x1080	DisplayPort	ALLINONE-RB-0173
14	HMDGBNCQXF	HP	Z24n G2	24	1920x1200	VGA	OPTIPLEX-RB-0024
15	DMK2OZDENW	Dell	P2719H	27	1920x1080	HDMI	OPTIPLEX-RB-0642
16	DMJQLJD5HV	Dell	P2719H	27	1920x1080	DisplayPort	OPTIPLEX-RB-0818
17	DMQ4Y45VJK	Dell	P2419H	24	1920x1080	HDMI	OPTIPLEX-RB-0535
18	HML4AYE6S8	HP	Z24n G2	24	1920x1200	DisplayPort	THINKCENTRE-RB-0990
19	HMLD97AGMA	HP	Z24n G2	24	1920x1200	VGA	OPTIPLEX-RB-0487
20	HMH5C62N3Q	HP	Z24n G2	24	1920x1200	DisplayPort	OPTIPLEX-VZ-0439
21	DM2DKZBGX3	Dell	U2415	24	1920x1200	DVI	PRODESK-RB-0729
22	HM3GLVZDEY	HP	Z24n G2	24	1920x1200	VGA	LATITUDE-RB-0370
23	DMM53A6VC3	Dell	U2415	24	1920x1200	VGA	PRODESK-RB-0258
24	HM3F3FAS9D	HP	E24 G4	24	1920x1080	HDMI	OPTIPLEX-RB-0629
25	DMW2GTJGXF	Dell	P2419H	24	1920x1080	HDMI	OPTIPLEX-RB-0786

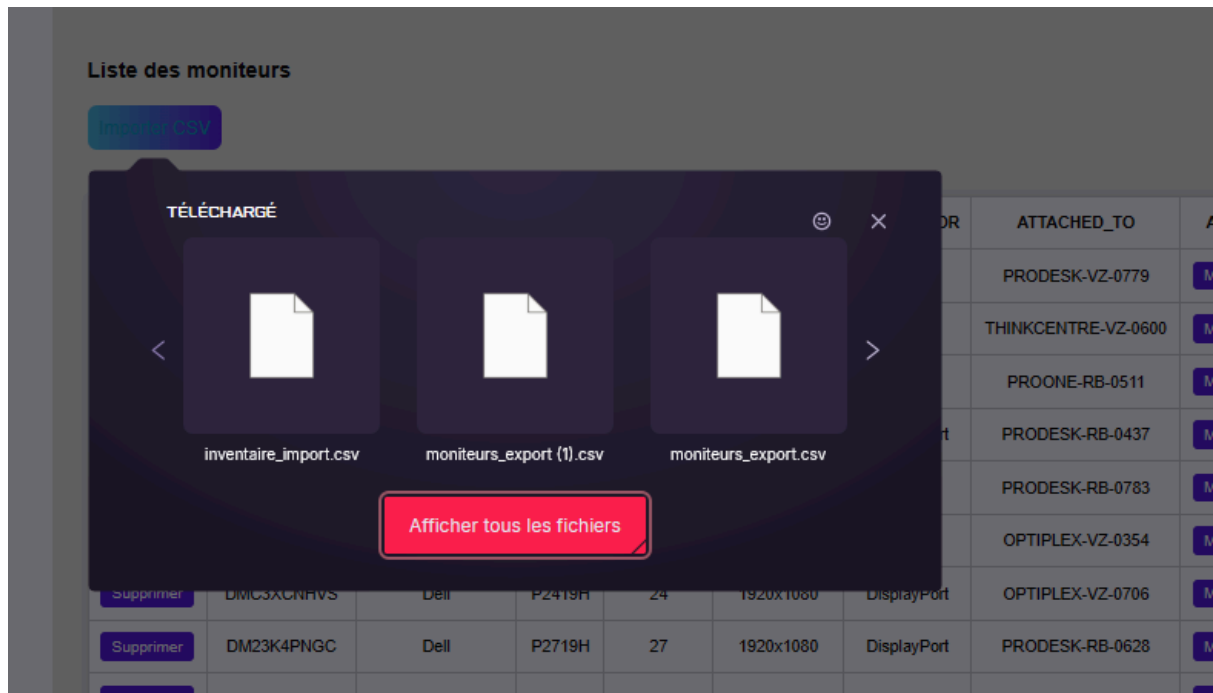
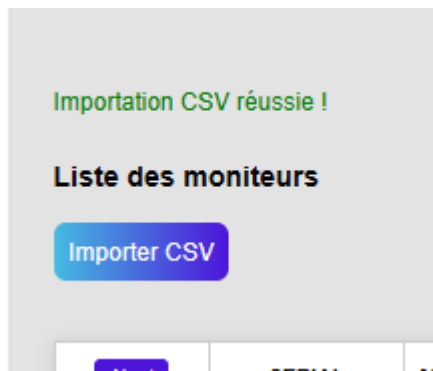
Nous avons fait la création à partir des deux fichiers csv des moniteurs et des unités centrales, vous pouvez les remplacer par votre propre CSV en gardant la même en-tête

 [inventory\\_devices.csv](#)

2025-12-05 17:39 173K

 [inventory\\_monitors2.csv](#)

2025-12-05 17:39 57K



Le bouton importer vous permet d'ajouter dans les tables moniteurs ou unités centrale les données de votre choix ayant l'en-tête correspondante.

Pour chaque ligne des éléments vous il y a un bouton supprimer et modifier qui fait apparaître un formulaire de modification.

Supprimer	DMER8JAVB7	Dell	P2419H	24	1920x1080	DVI	OPTIPLEX-RB-0741	Modifier
Supprimer	HMFCB3PF8A	HP	Z24n G2	24	1920x1200	DisplayPort	THINKCENTRE-VZ-0176	Modifier
Supprimer	HMNVDKCANV	HP	E24 G4	24	1920x1080	HDMI	OPTIPLEX-VZ-0514	Modifier

### Modifier un moniteur

SERIAL	<input type="text" value="DMXUMX8M5R"/>
MANUFACTURER	<input type="text" value="DellICC"/>
MODEL	<input type="text" value="P2419H"/>
SIZE_INCH	<input type="text" value="24"/>
RESOLUTION	<input type="text" value="1920x1080"/>
CONNECTOR	<input type="text" value="HDMI"/>
ATTACHED_TO	<input type="text" value="PRODESK-VZ-0779"/>
<input type="button" value="Modifier"/>	

### Liste des moniteurs