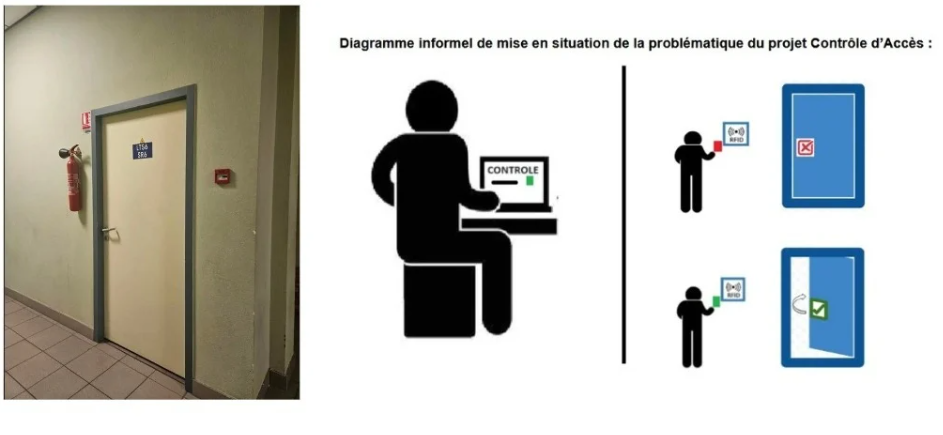
# **INTRODUCTION**

Le lycée Charles Poncet envisage de remplacer la serrure mécanique du local informatique par un système d’accès plus moderne et sécurisé, basé sur l’utilisation de badges.

La fabrication des clés mécaniques devenant de plus en plus coûteuse, l’adoption d’un système de cartes NFC couplé à un lecteur RFID fixé au mur constitue une alternative pertinente. Ce dispositif offrirait un accès simplifié et renforcé au local, grâce à l’intégration d’un système de supervision à distance et d’une interface Web dédiée à la gestion des accès.

À chaque lecture de carte, le lecteur RFID vérifiera l’authenticité du badge en consultant une base de données. L’accès sera accordé ou refusé en fonction de la validité des informations. De plus, le système enregistrera automatiquement les données suivantes : l’heure de la tentative d’accès, le résultat (succès ou échec), l’UID de la carte, ainsi que l’identité de son propriétaire, si celui-ci est préalablement enregistré dans la base.

Nous avons une contrainte budgétaire imposé à hauteur de 130 euros, vous trouverez en annexe un tableau récapitulatif de nos dépenses.

# **MATÉRIELS UTILISES**

Le fonctionnement du système repose sur un ensemble de composants matériels essentiels, dont l’élément central est le Raspberry Pi. Ce nano-ordinateur, ici dans sa version Raspberry Pi 4, est capable de se connecter à un moniteur et d’exécuter le code chargé de gérer l’authentification par badge, afin de contrôler l’accès au local.

Parmi les équipements complémentaires, on trouve :

* Un lecteur de carte RFID, encastré dans le mur, utilisé pour l’identification des utilisateurs autorisés.
* Un lecteur NFC distinct, dédié à l’administrateur, permettant l’enregistrement et la gestion des cartes d’accès.
* Des cartes NFC servant de badges pour les utilisateurs.
* Une gâche électromécanique pour le verrouillage et le déverrouillage de la porte.
* Un détecteur de porte filaire, permettant de connaître en temps réel l’état de la porte (ouverte ou fermée).
* Un détecteur de présence pour renforcer la sécurité ou déclencher certaines actions automatiques.

Enfin, un serveur est requis pour héberger l’ensemble des services nécessaires : le serveur web, la base de données, ainsi que l’interface de gestion en ligne.

# **Logiciels et Langages utilisés**

Le langage de programmation choisi pour le développement de la partie embarquée sur le Raspberry Pi est Python. Ce choix s’explique par la grande compatibilité de Python avec les périphériques tels que les capteurs RFID, les détecteurs de mouvement (PIR), les relais, et autres composants électroniques.

Python dispose d’un large éventail de bibliothèques puissantes, bien documentées et faciles à intégrer, ce qui facilite grandement le développement.

De plus, Python offre des fonctionnalités avancées pour la sécurisation du système : gestion des rôles et des droits d’accès, création de journaux d’événements (logs), envoi de notifications par e-mail, etc. Il s'agit donc d'un langage parfaitement adapté à ce type de projet, à la fois flexible, fiable et largement supporté par la communauté.

Le serveur est hébergé localement sur un poste informatique du lycée, équipé d’un système d’exploitation Linux. L’environnement de gestion utilisé est aaPanel, un panneau de contrôle web qui simplifie l’administration des services essentiels, tels que le serveur web, la base de données et le site internet. Cette solution permet une gestion centralisée, intuitive et efficace des différents composants du système.

Le développement du site web repose sur l’utilisation de plusieurs langages complémentaires.  
 La partie front-end, dédiée à l’interface utilisateur, a été conçue en HTML et CSS, afin de structurer les pages et d’en assurer le design visuel.

La partie back-end, responsable du traitement des données et de la communication avec la base de données, a été développée en PHP. Elle intègre également l’utilisation du langage SQL pour les requêtes vers la base de données, ainsi que des fonctionnalités avancées telles que l’envoi d’e-mails automatisés (notifications, alertes, confirmations, etc.).

# **D****iagramme de cas d’utilisation**

Dans ce diagramme de cas d'utilisation UML, les acteurs représentent les entités externes (humaines ou systèmes) qui interagissent avec le système de contrôle d’accès.

### **Acteurs principaux :**

Admin (Administrateur)

* Rôle : Il supervise et gère les accès au local.
* Cas d’utilisation associés :
* *Superviser les accès au local*
* *Surveiller l’état du local*

Employé

* Rôle : Utilisateur standard qui peut demander l’accès au local et y accéder selon les autorisations définies.
* Cas d’utilisation associés :
* *Demander autorisation d’accès au local*
* *Accéder au local*

Temps (représenté comme un système externe, pas une personne)

* Rôle : Fournit une valeur temporelle
* Cas d’utilisation associé :
* *Surveiller l’état du local*

### **Autres composants non-acteurs mais liés :**

* Local (représente le système physique : porte, serrure, détecteurs, etc.)
* BDD (la base de données, utilisée pour vérifier les accès, stocker les logs, etc.)

# **DIAGRAMME DE SÉQUENCE**

Nous allons désormais vous présenter les diagrammes de séquence pour chaque cas d’utilisation.

*Demander autorisation d’accès au local ;*

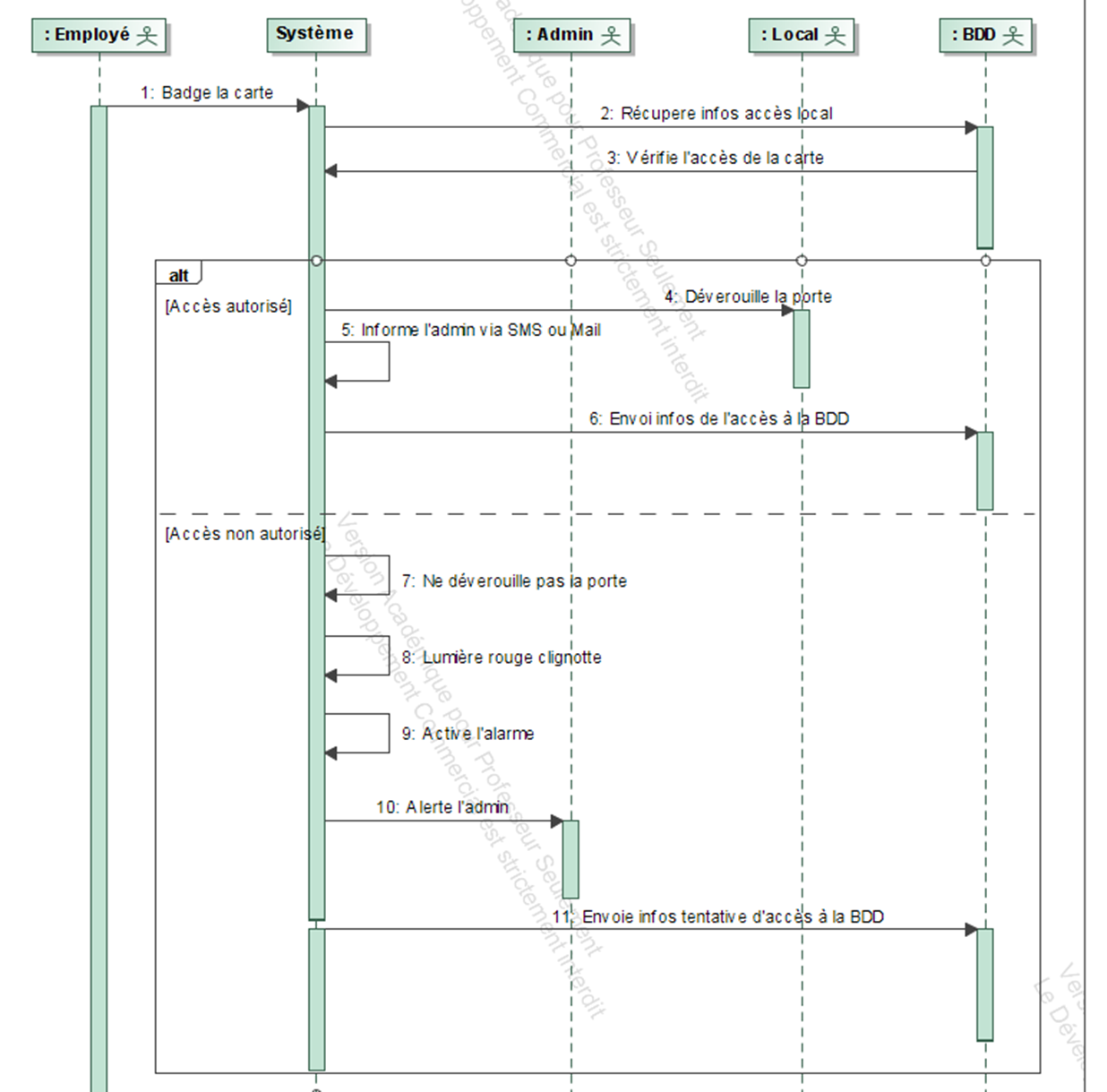
Une image contenant texte, diagramme, Parallèle, ligne

Le contenu généré par l’IA peut être incorrect.

Une image contenant texte, diagramme, capture d’écran, ligne

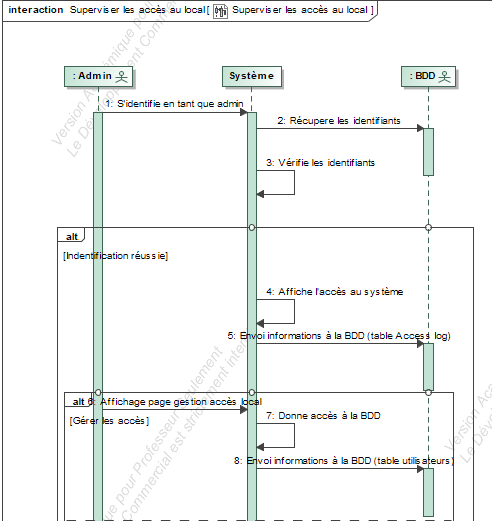
Le contenu généré par l’IA peut être incorrect.*Surveiller l’état du local ;*

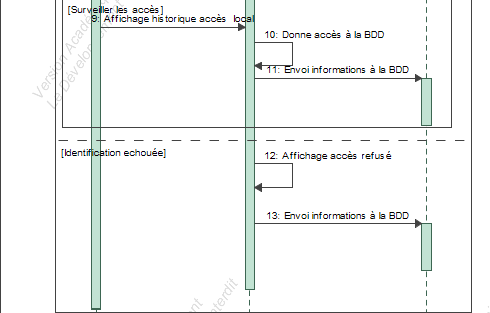
*Accéder au local ;*



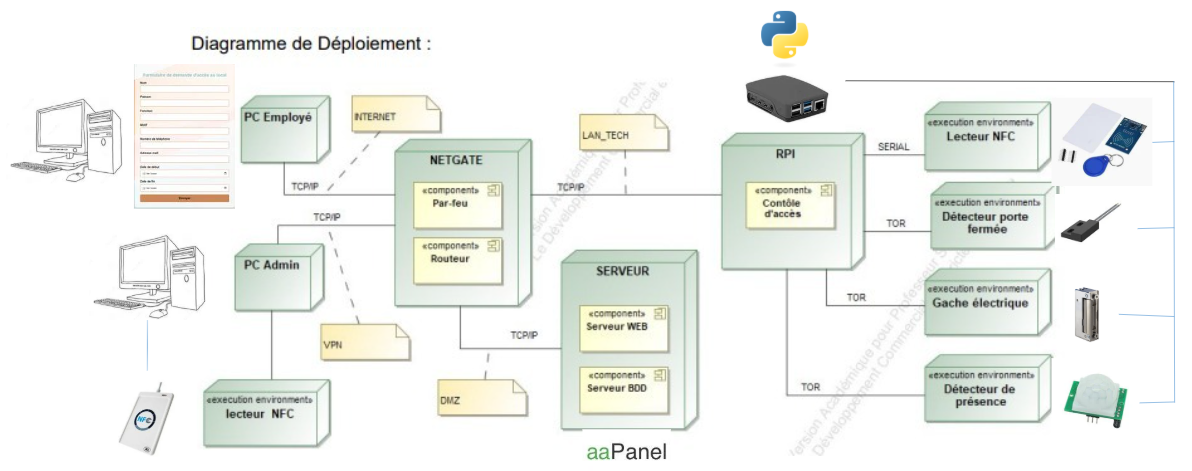
Une image contenant texte, diagramme, ligne, Tracé

Le contenu généré par l’IA peut être incorrect.

*Superviser les accès au local ;*



# **DIAGRAMME DE DÉPLOIEMENT**

 Le diagramme de déploiement ci-dessus présente l’architecture physique du système de contrôle d’accès, en détaillant la répartition et l’interconnexion des différents composants matériels et logiciels.

1. Postes clients :

PC Employé

Ce poste permet aux utilisateurs de soumettre des demandes d’accès au local via un formulaire web. Les données sont transmises au serveur à travers le réseau Internet, en passant par l'infrastructure de sécurité réseau.

PC Administrateur

Le poste administrateur est utilisé pour enregistrer les cartes NFC via un lecteur dédié et gérer les droits d’accès. Il permet également de superviser les journaux d’événements et l’état du système.

2. Infrastructure réseau :

Netgate (Pare-feu / Routeur)

Il assure la sécurité et le routage des connexions entre les différents segments du réseau. Il intègre une DMZ pour l’hébergement sécurisé des services web accessibles depuis l’extérieur, et établit une liaison VPN avec les postes internes.

3. Serveur :

Système d’exploitation : Linux

Outil de gestion : aaPanel

4. Services hébergés :

Serveur Web : héberge l’interface web utilisée par les employés et l’administrateur.

Serveur BDD (Base de Données) : stocke les informations relatives aux utilisateurs, aux badges, aux tentatives d’accès, et aux logs système.

5. Raspberry Pi (RPI) – Unité de contrôle d’accès :

Le Raspberry Pi 4 joue un rôle central dans le dispositif. Il exécute le programme Python responsable de la gestion locale des accès. Il est connecté à plusieurs capteurs et actionneurs :

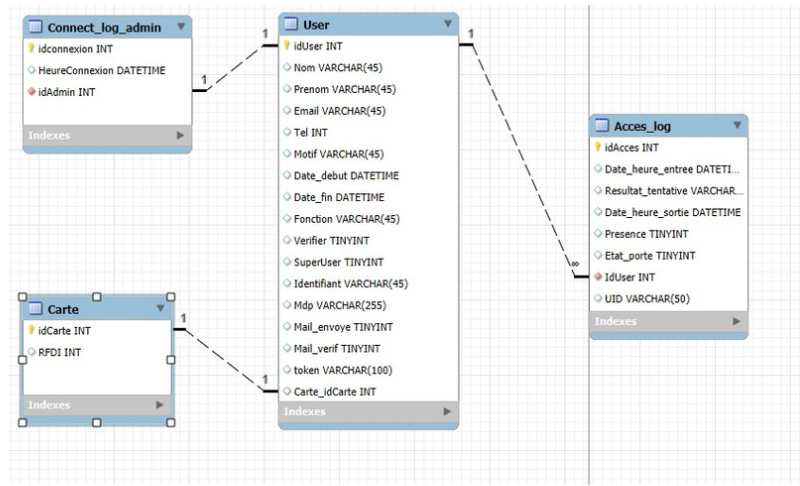
Lecteur NFC (série) : pour la lecture des badges à l’entrée.

Détecteur d’état de porte (TOR) : informe si la porte est ouverte ou fermée.

Gâche électromécanique (TOR) : actionnée pour verrouiller/déverrouiller l’accès.

Détecteur de présence (TOR) : permet de détecter une personne à proximité du local.

# **MODÈLE DE BASE DE DONNÉES**



Le système repose sur une base de données relationnelle structurée autour de quatre tables principales, chacune jouant un rôle essentiel dans le bon fonctionnement de l’architecture.

#### 1. Table User : Cette table constitue le noyau central de la base de données. Elle regroupe l’ensemble des utilisateurs du système, qu’il s’agisse d’administrateurs ou d’employés. Elle contient également les informations issues des formulaires de demande d’accès.

#### 2. Table Carte : Elle recense les différentes cartes d’accès enregistrées dans le système. Chaque enregistrement contient un identifiant de carte ainsi que son identifiant RFID. La clé primaire de cette table est utilisée comme clé étrangère dans la table User, ce qui permet d’associer chaque utilisateur autorisé à une carte d’accès unique.

#### 3. Table Connect\_log\_admin Cette table permet de tracer l’historique des connexions des administrateurs au système. Elle inclut des informations telles que la date, l’heure et l’identifiant de l’administrateur connecté. L'identifiant de l’utilisateur est lié à cette table via une clé étrangère issue de la table User, assurant un suivi sécurisé des accès à l’interface d’administration.

#### 4. Table Acces\_log Elle enregistre l’historique des tentatives d’accès au local, qu’elles soient réussies ou échouées. Parmi les champs stockés, on retrouve : l’heure de la tentative, son résultat (succès ou échec), et l’identifiant de l’utilisateur concerné, obtenu via une clé étrangère liée à la table User.

# **INSTALLATION SERVEUR**

Afin de lancer le projet, la première étape a consisté à mettre en place un environnement serveur pour l’hébergement des services web et de la base de données.

#### 1. Préparation de la machine

Un poste informatique du lycée a été dédié à cette fonction et entièrement réinitialisé. Le système d’exploitation **Ubuntu** (distribution Linux) a été installé pour sa stabilité, sa sécurité, et sa compatibilité avec les outils serveur open source.

#### 2. Installation d'aaPanel

Pour faciliter la gestion des services LAMP (Linux, Apache, MySQL, PHP), nous avons opté pour l’installation de **aaPanel**, un panneau de gestion web permettant d’administrer les différents services depuis une interface centralisée.

**Procédure d'installation :**  
  
L’installation d’aaPanel s’effectue via le terminal. La commande suivante a été exécutée sur le poste serveur :

*URL=https://www.aapanel.com/script/install\_7.0\_en.sh && if [ -f /usr/bin/curl ];then curl -ksSO "$URL" ;else wget --no-check-certificate -O install\_7.0\_en.sh "$URL";fi;bash install\_7.0\_en.sh aapanel*

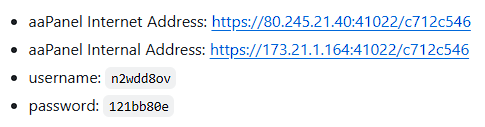
Cette commande télécharge et lance automatiquement le script d’installation. Une fois le processus terminé, aaPanel est accessible via une interface web.

#### 3. Accès à l’interface d’administration

Pour accéder à l’interface web d’aaPanel, la commande suivante est utilisée dans le terminal :

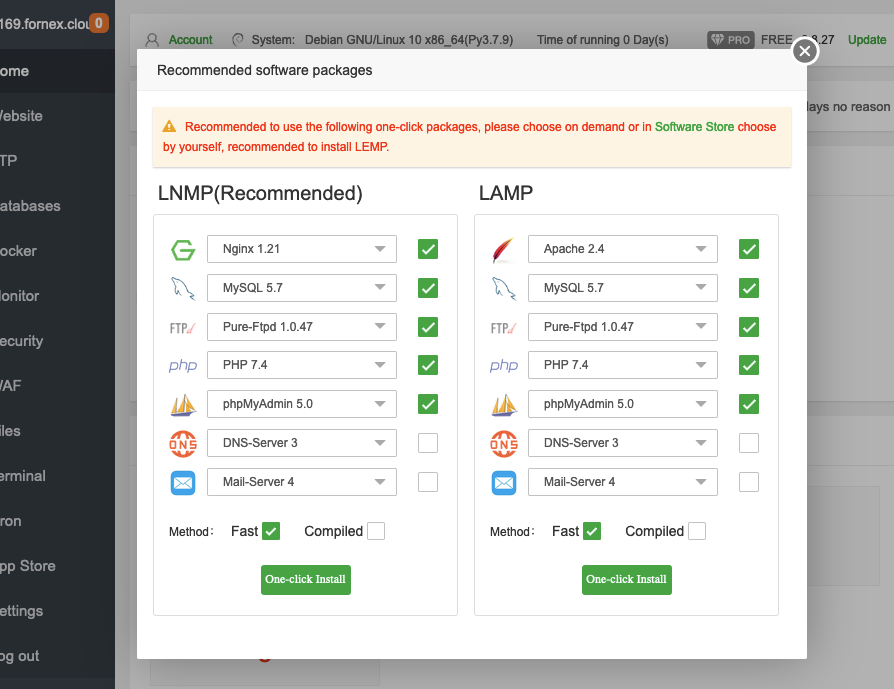
*sudo bt 14*

Cette commande permet d’afficher l’adresse IP d’accès à l’interface ainsi que les identifiants de connexion (nom d’utilisateur et mot de passe). Il est important de noter que ces identifiants ne sont affichés qu’une seule fois et doivent être conservés soigneusement.

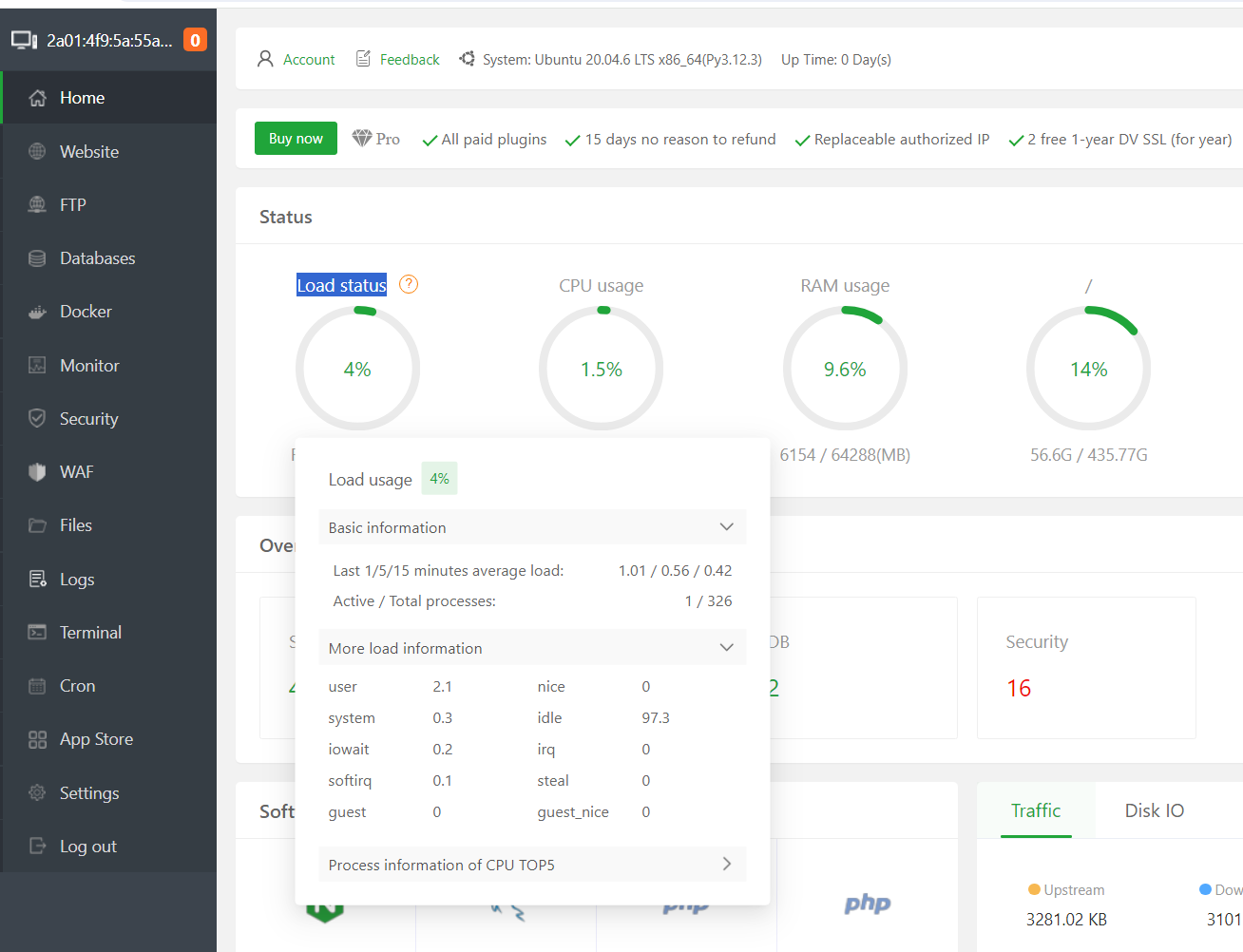


### Voici à quoi ressemble la page de connexion aaPanel :

### 

Une fois connecté sur la page d’accueil, aaPanel nous propose ainsi d’installer les services LAMP et bien d’autres.

Voici à quoi ressemble le dashboard de aaPanel :



Une fois connecté à l’interface web d’aaPanel, l’utilisateur accède à un tableau de bord complet. Sur la partie gauche de l’écran, un menu regroupe l’ensemble des onglets nécessaires à l’administration du serveur.

Parmi ces fonctionnalités, deux modules sont particulièrement essentiels au bon fonctionnement du projet :

* **Website** : permet de créer, configurer et gérer le site web directement depuis l’interface.
* **Databases** : permet de créer et administrer les bases de données MySQL ou MariaDB, facilitant l’hébergement et la gestion des données du système.

Cette interface graphique rend l’ensemble des opérations plus accessibles et fluides, même pour des utilisateurs ne maîtrisant pas les commandes en ligne. Elle centralise l’administration des services et simplifie le déploiement des composants du projet.

### Connexion base de données

La base de données à été réalisé préalablement sur le logiciels MySql Workbench, comme illustré plus tôt.



La base de données du projet a été créée à l’aide de **phpMyAdmin**, un outil d’administration MySQL accessible via l’interface web fournie par aaPanel.

Le **modèle de données** a été conçu et structuré à l’aide de **MySQL Workbench**, un outil dédié à la modélisation et à la gestion des bases de données relationnelles.   
Une fois le schéma finalisé, celui-ci a été **exporté au format SQL** puis importé dans phpMyAdmin.

Cette opération a permis d’instancier automatiquement les différentes tables, clés primaires, relations et contraintes nécessaires au bon fonctionnement du système.   
La base de données est ainsi **entièrement déployée, fonctionnelle et prête à être exploitée** par les différentes parties du système (site web, Raspberry Pi, interface administrateur).

# **FINALISATION**

Cette première partie du rapport a permis de présenter les fondations techniques et fonctionnelles du projet commun, de la mise en place de l'infrastructure serveur jusqu’à la conception de la base de données et de l’interface de gestion. Elle témoigne de la collaboration de l’ensemble du groupe sur les éléments essentiels au bon fonctionnement du système. Dans les parties suivantes, chaque membre du groupe présentera de manière détaillée sa contribution personnelle, en mettant en lumière les aspects spécifiques dont il a eu la responsabilité, ainsi que les compétences techniques mobilisées tout au long du projet.

**Annexes**