Dossier Technique du projet « Accès campus » - Partie PSW (Poste Serveur Web)

Table des matières

[I – Présentation 2](#_Toc199930118)

[1.1 - Contexte et objectifs 2](#_Toc199930119)

[1.2 - Description du système 3](#_Toc199930120)

[1.3 – Description de la partie personnelle 3](#_Toc199930121)

[II – Conception 5](#_Toc199930122)

[2.1 – Objectif 5](#_Toc199930123)

[2.2 – Mise en place PSW 6](#_Toc199930124)

[2.3 – Fonctionnement du site 6](#_Toc199930125)

[2.4 – Réalisation 7](#_Toc199930126)

[2.5 – Test 14](#_Toc199930127)

[2.5.1 – Test unitaire 14](#_Toc199930128)

[2.5.2 – Test d’intégration 15](#_Toc199930129)

[III- Conclusion 17](#_Toc199930130)

# I – Présentation

## 1.1 - Contexte et objectifs

Le projet Campus Accès vise à améliorer la gestion et la sécurité des accès aux salles du Campus Saint Aubin La Salle. Grâce à un système automatisé basé sur des badges RFID, les étudiants et le personnel peuvent accéder aux salles de manière sécurisée tout en enregistrant leurs entrées et sorties. L’objectif est d’assurer un contrôle centralisé des accès, optimisé grâce au Poste Serveur Web (PSW).



Figure 1 Campus Saint Aubin La Salle

Ma tâche personnelle dans ce projet de grande envergure, est de permettre la visualisation pour les élèves ainsi que le personnel des absences, retards et des informations des différentes salles. (*voir Figure 2*).

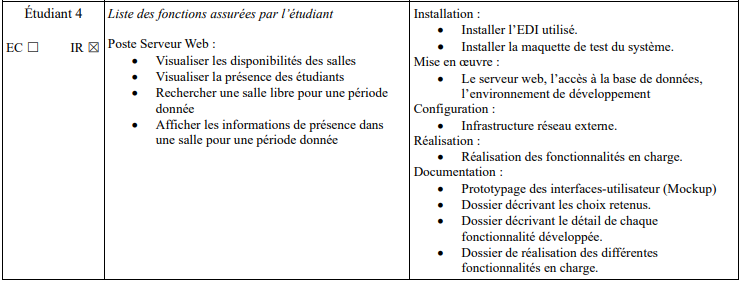


Figure 2 Mission étudiant 4

## 1.2 - Description du système

Le Poste Serveur Web (PSW) est au cœur du système, il hébergera une base de données, une API ainsi qu’un site web. Ma partie est celle du site web, il permettra aux élèves, aux professeurs, à la vie scolaire de se connecter via leurs comptes personnels pour y voir leurs absences, retards, et les indications des différentes salles du lycée. La seule communication du site sera avec l’API à l’aide de requêtes GET et POST.

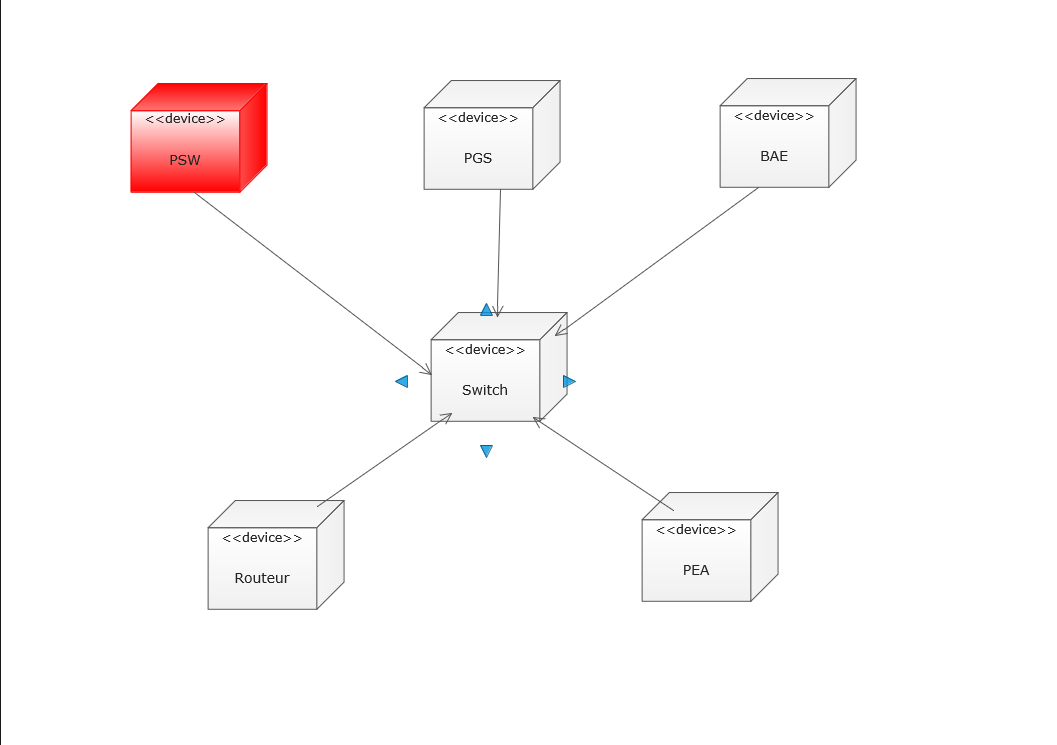


Figure 3 Diagramme de déploiement

## 1.3 – Description de la partie personnelle

A l’aide du diagramme des cas d’utilisations fourni dans le cahier des charges mes différentes tâches sont données.

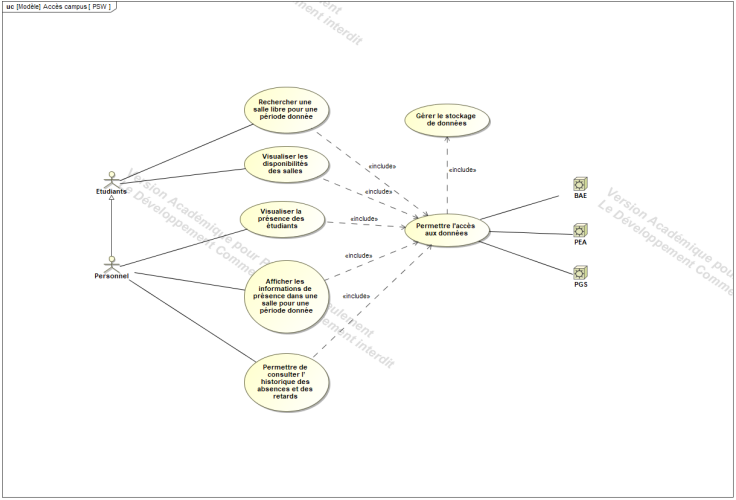


Figure 3 Diagramme des cas d’utilisations

Le PSW est divisé en 3 partie, BDD, API et site web, Lorick Fouquet s’occupant de la partie gestion de données, il nous donc a fallu travailler en binôme tout au long du projet. On a donc dès le début dû se mettre d’accord sur certains choix pour travailler sur le PSW tel que l’OS que nous allons utiliser qui est donc Debian 12. Nous avons fait le choix de travailler chacun sur une VM puis lorsque nous avions terminés une éventuelle tâche on assemblerait nos morceaux fonctionnels sur le serveur. Sur le PSW, nous avons d’abord installé différents paquets dont ufw comme pare feu qui nous a permis de fermé tous les ports inutilisés afin d’augmenter la sécurité sur le PSW, les ports ouvert sont écrit dans le fichier Installation.txt dans la partie PSW sur le GitHub du projet. En ce qui me concerne les paquets nodejs et npm ont été aussi installé pour le codage du site.

Afin de préciser les besoins pour le site PSW voici un diagramme de requierement :

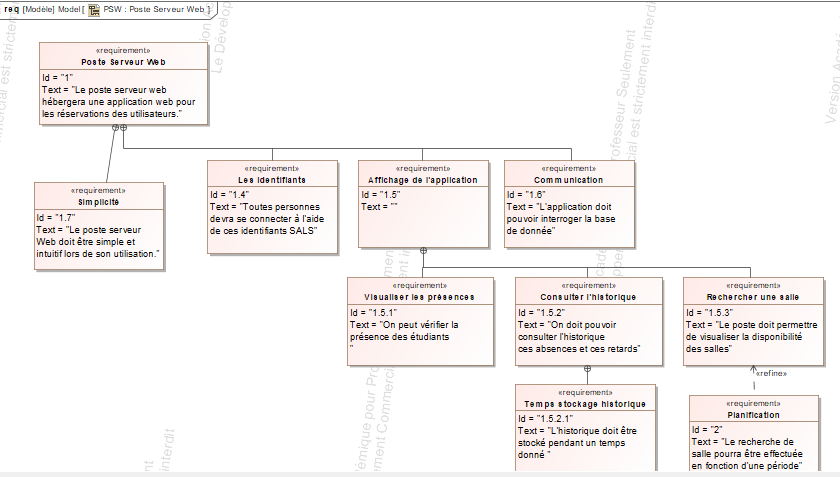


Figure 4 Diagramme de requierement

# II – Conception

## – Objectif

L’objectif de ce premier incrément pour ma part sera :

* Créer la base du site web
* Permettre de consulter l’historique des absences et des retards.
* Préparer le serveur

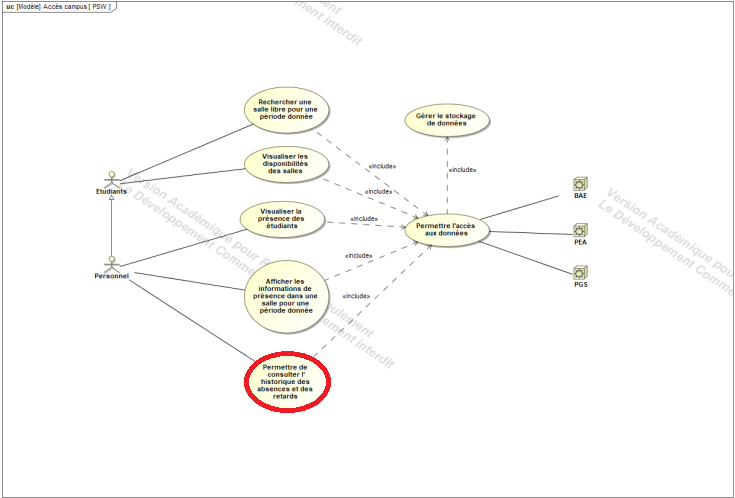


Figure 5 Diagramme des cas d’utilisations

## – Mise en place PSW

Pour cette partie je ne suis pas seul à travailler dessus, je continue de partager mon travail avec Lorick Fouquet.

On a choisi de partir sur Debian 12 comme OS, pour sa stabilité. C’est plus adéquat que Ubuntu pour du faire du 24h sur 24. On installe ssh dès le début pour ensuite pouvoir travailler à distance dessus puis mettons en place le pare-feu ufw.

Nous avons installé micro comme éditeur de texte qui est plus simple et agréable d’utilisation que vim ou nano.

## – Fonctionnement du site

Le site aura un fonctionnement simple, il devra être utilisable par les élèves, les professeurs, la vie scolaire, des personnes pas toujours habituées à utiliser l’informatique. Pour se faire, lors de l’utilisation du site il faudra obligatoirement se connecter à l’aide d’un compte personnel attribué à chacun. Une fois connecté, plusieurs possibilités seront possibles, elles dépendront de leurs « rôle » (professeur, élèves, administrateur…), par exemple les élèves pourront voir leurs propres absences et retards ainsi que les informations sur les salles tandis que le personnel auront accès à la liste des élèves complètes avec la possibilité de voir leurs absences et retards en plus d’avoir accès aux informations des différentes salles du lycée.

## – Réalisation

De ce qui est de la réalisation, le site web ne s’est évidemment pas fait en une fois, plusieurs versions existent. Elles sont disponibles sur le GitHub afin de permettre un retour en arrière en cas de gros problèmes.

Le premier cas d’utilisation était de visualiser les absences et retards, c’est donc par là que j’ai commencé.



Figure 6 Illustration absences / retards

Cependant au fur et à mesure il y a eu de l’évolution dans le projet, et la « figure 6 » s’est transformé en « figure 7 », plus esthétique et plus agréable à l’œil pour l’utilisateur.

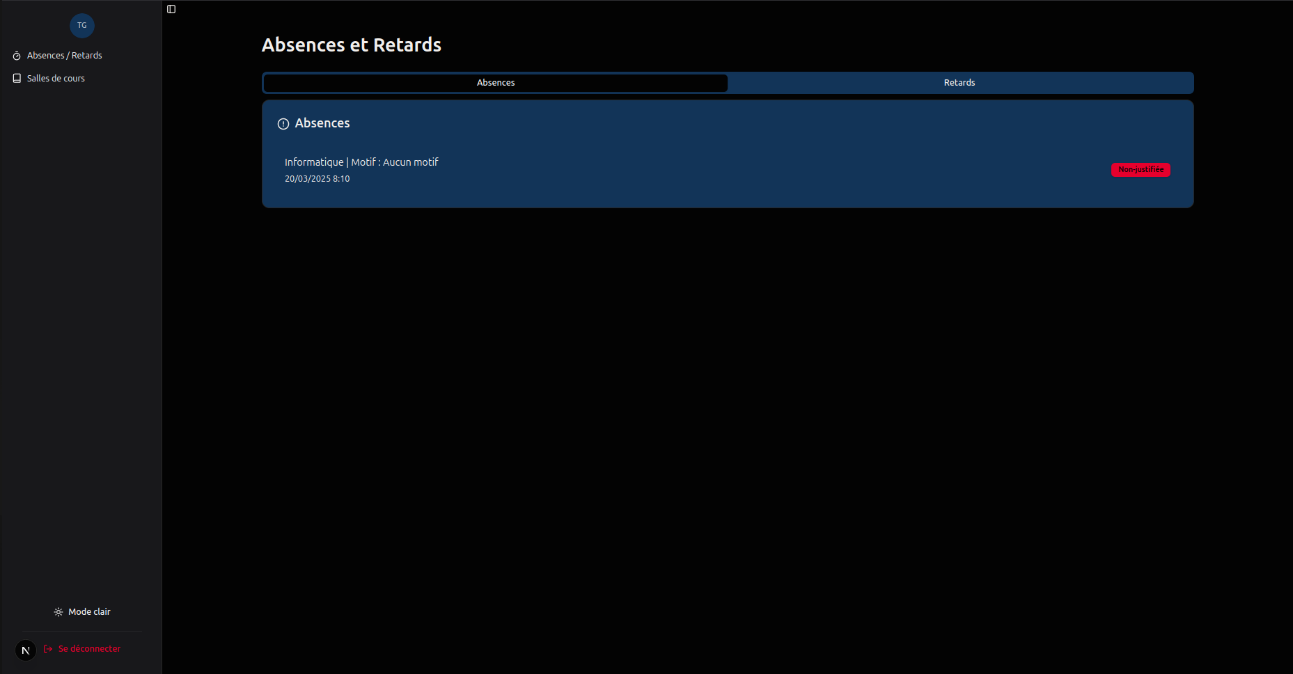


Figure 7 Illustration absences / retards

Pour récupérer les absences et retards j’utilise les routes données par Lorick Fouquet en utilisant la méthode GET dans mon code.



Figure 8 Requête GET absence

Pour faire les requêtes d’absence et de retard il fallait d’abord avoir l’id de l’utilisateur qui voulait les voir, pour se faire la récupération de l’id se fait lorsque l’utilisateur se connecte dans la page de login. Nous faisons une requête POST, suite à cela, la BDD renvoie les infos de l’utilisateur ainsi que son id.

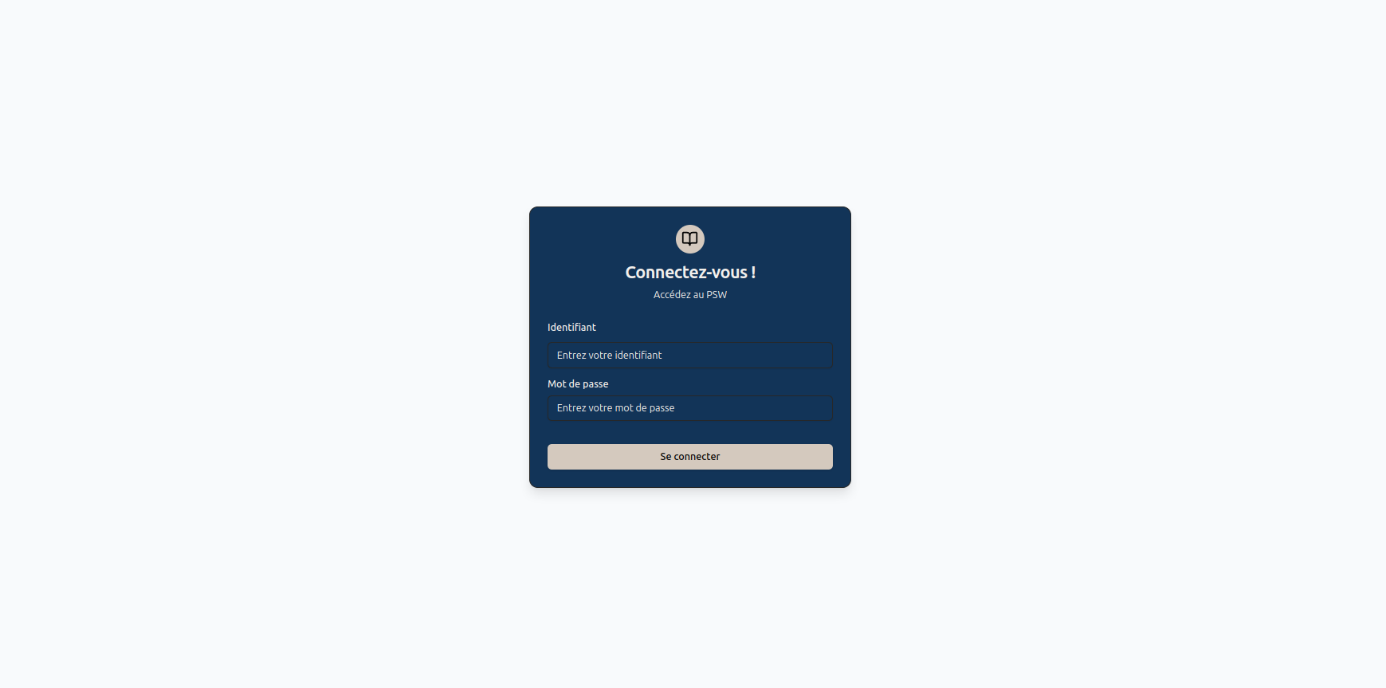


Figure 9 Page de login

Dans le code, nous faisons une requête

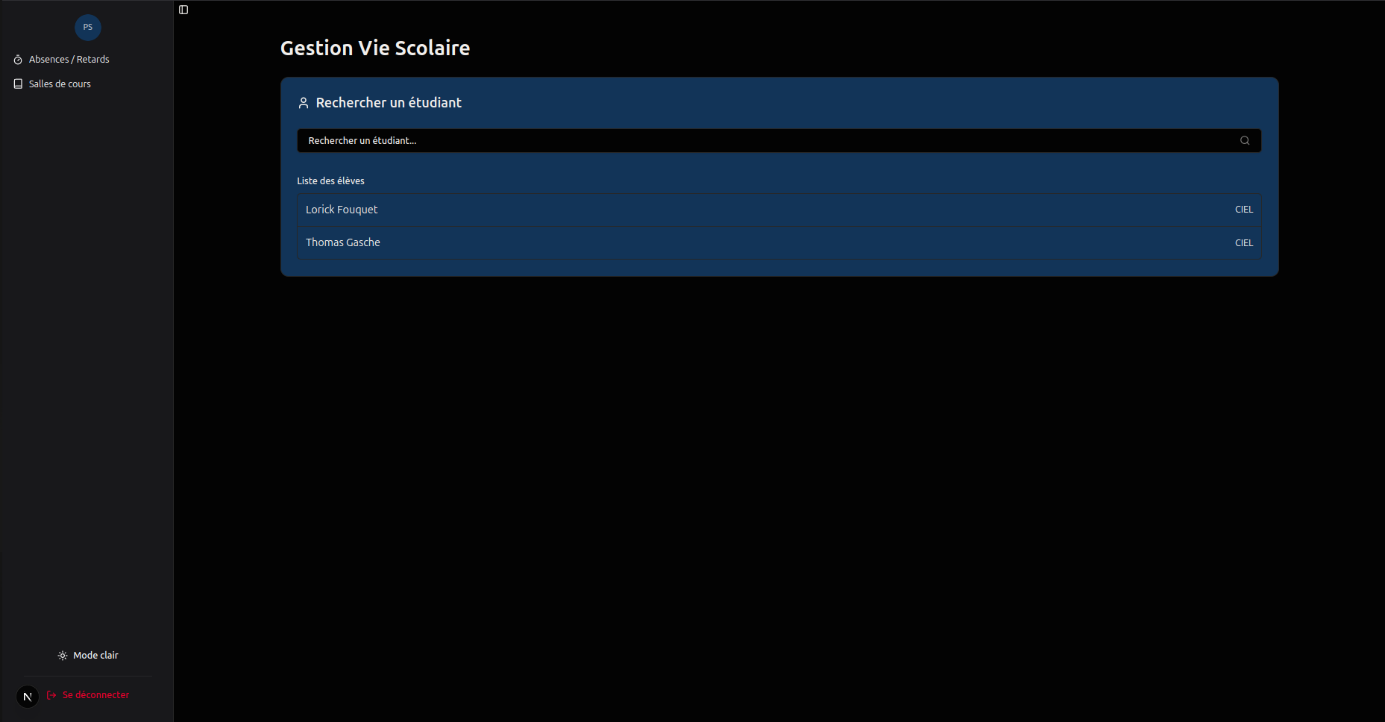
POST avec les identifiant personnel de

l’utilisateur puis créons un cookie qui sera

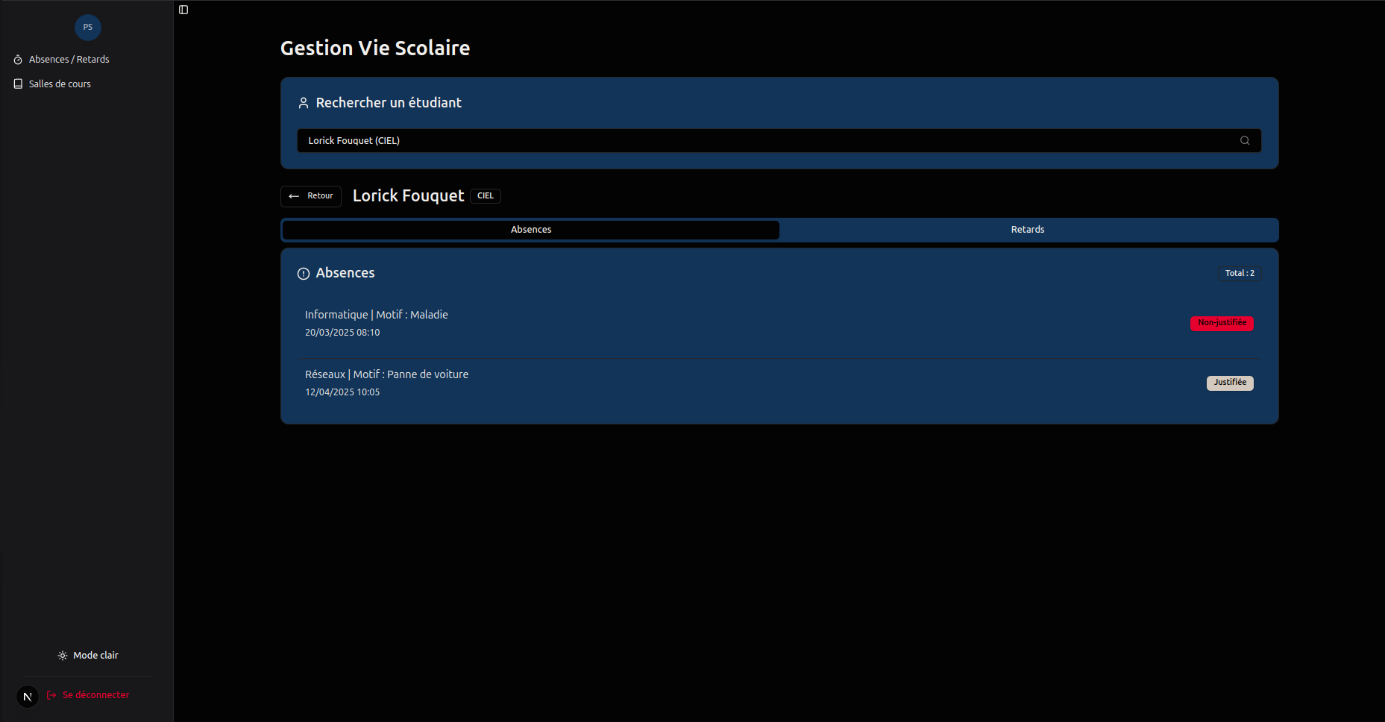
ensuite réutiliser pour accéder au site.

*Figure 10 Code requête login*

Bien sûr, le point de vue vie scolaire ou professeur n’est pas le même que celui des élèves. Le personnel à accès à la liste des élèves puis aux absences et retards de chacun d’entre eux.

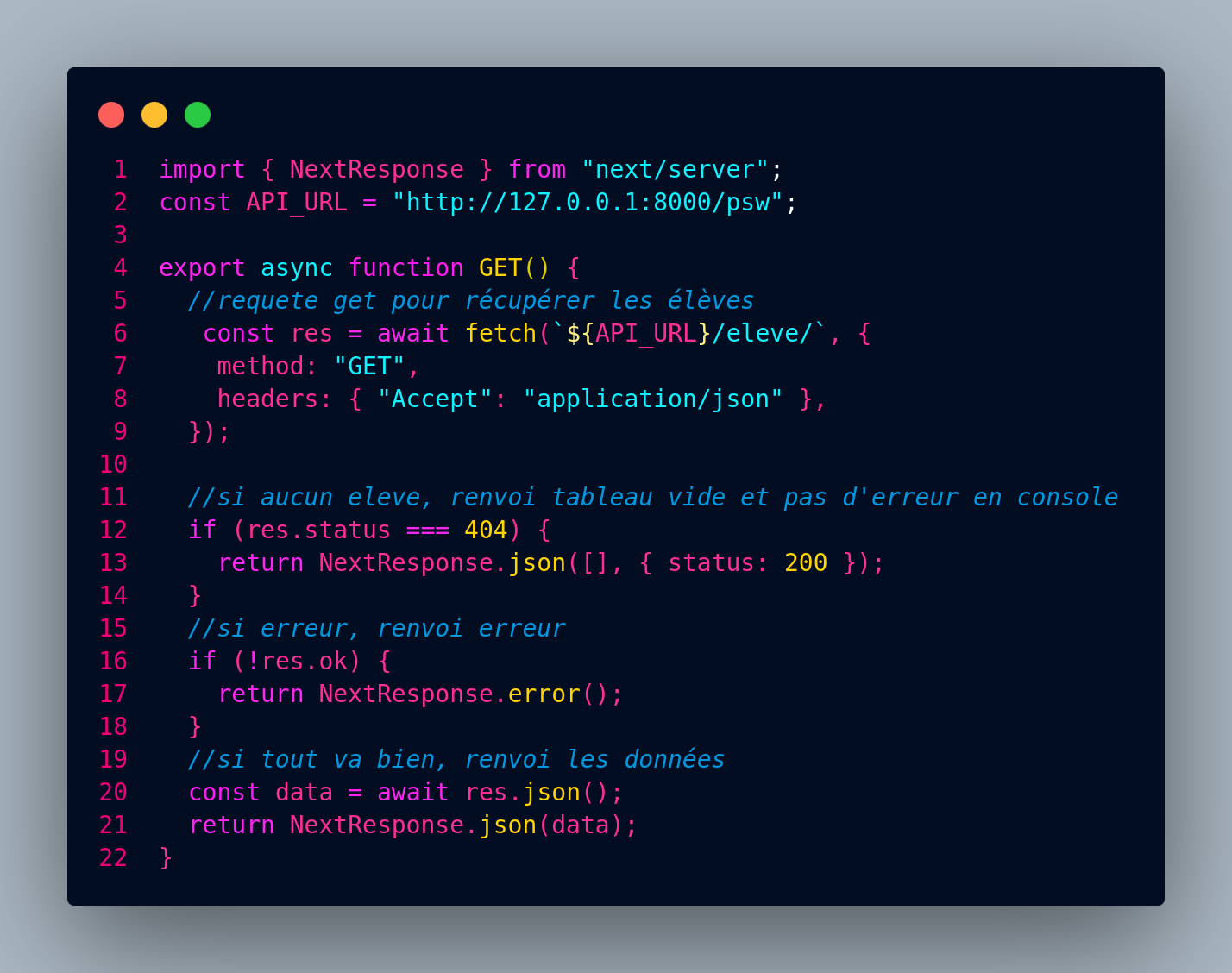


*Figure 11 Absences / Retards vie scolaire*



*Figure 12 Absences / Retards d’un élève*

Afin d’avoir la liste des élèves, Lorick Fouquet a créé une route pour me permettre une requête GET qui sert à afficher au personnel connecté d’afficher les élèves.

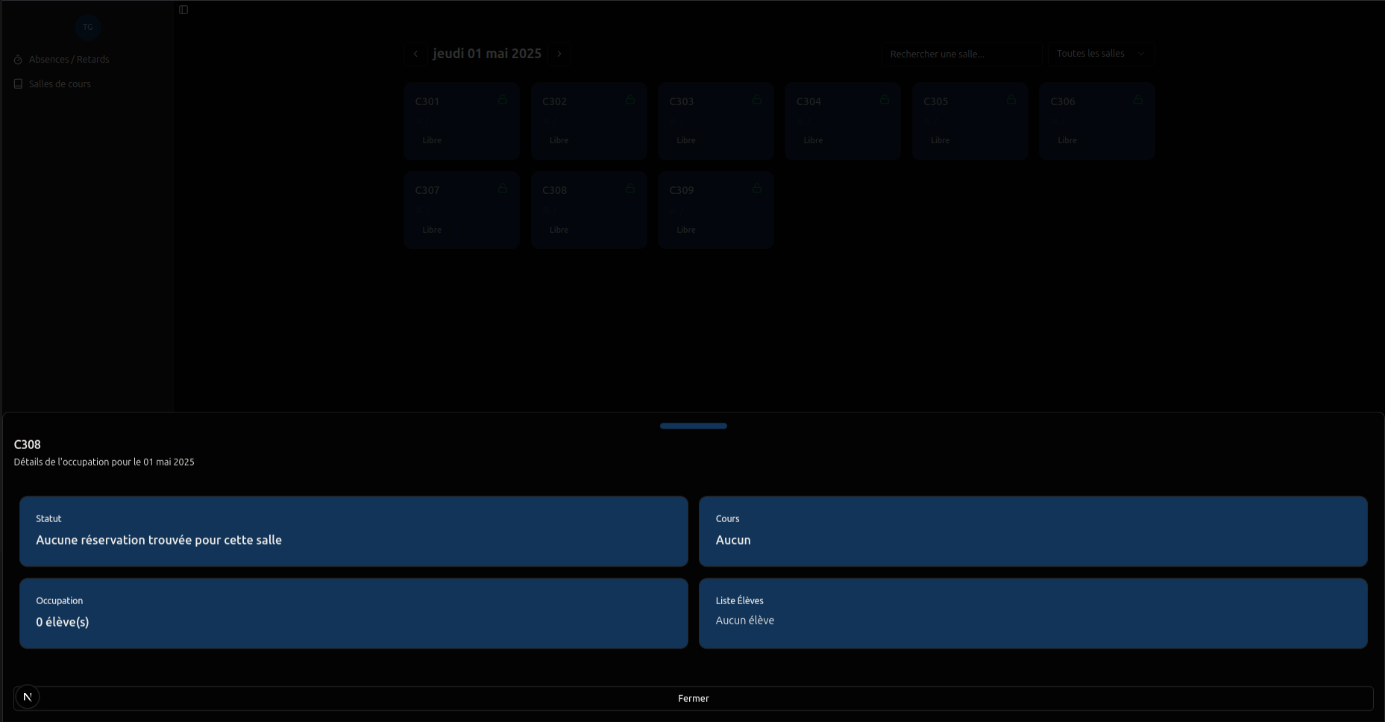


*Figure 13 Requête GET liste élève*

Une fois la partie absences/retards terminées, il fallait s’occuper des salles et plus précisément des informations des salles. Nous devons voir le nombre d’élèves présents, si la salle est occupée ou non, si elle est réservée ou non et le cours actuel ou futur qui s’y déroule ou déroulera. Toutes ces informations seront visibles pour le personnel ainsi que les élèves. Voici en figure 14 et 15 l’affichage coté utilisateur des salles ainsi que des informations sur celles-ci.

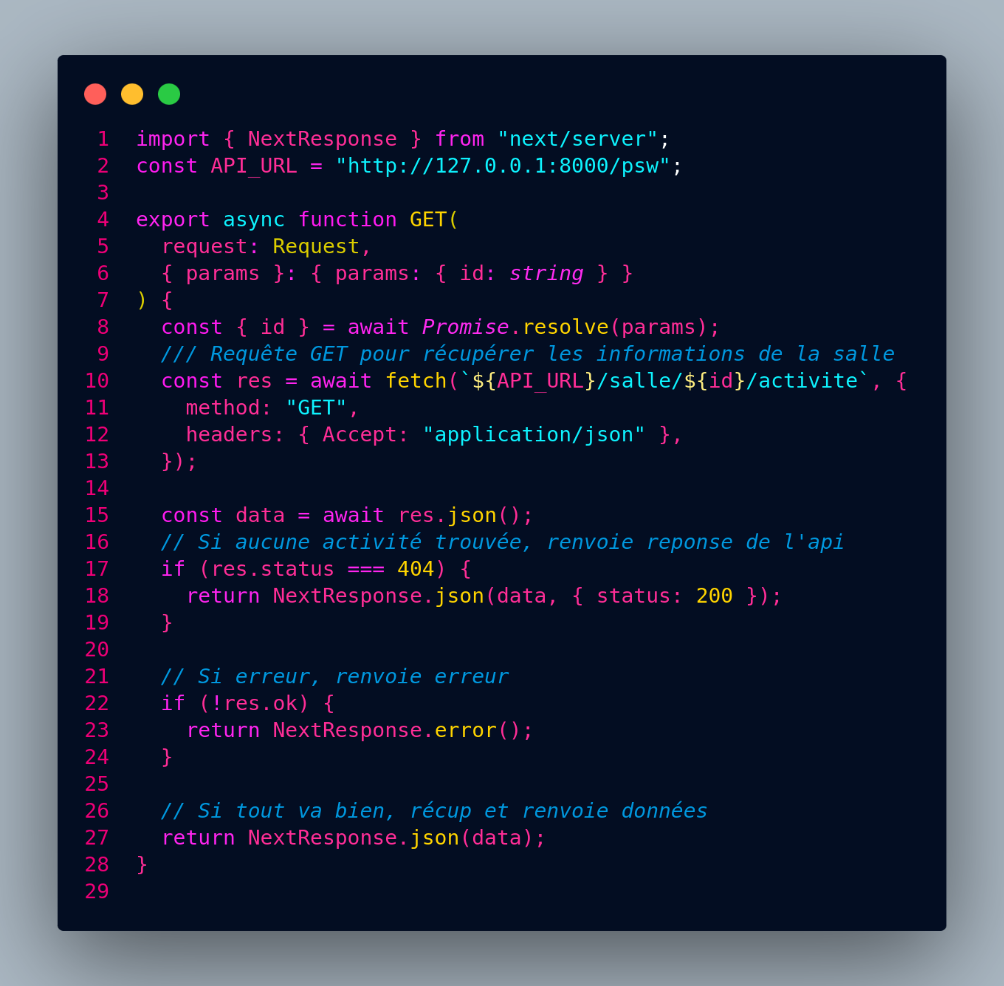


*Figure 14 Affichage des salles*



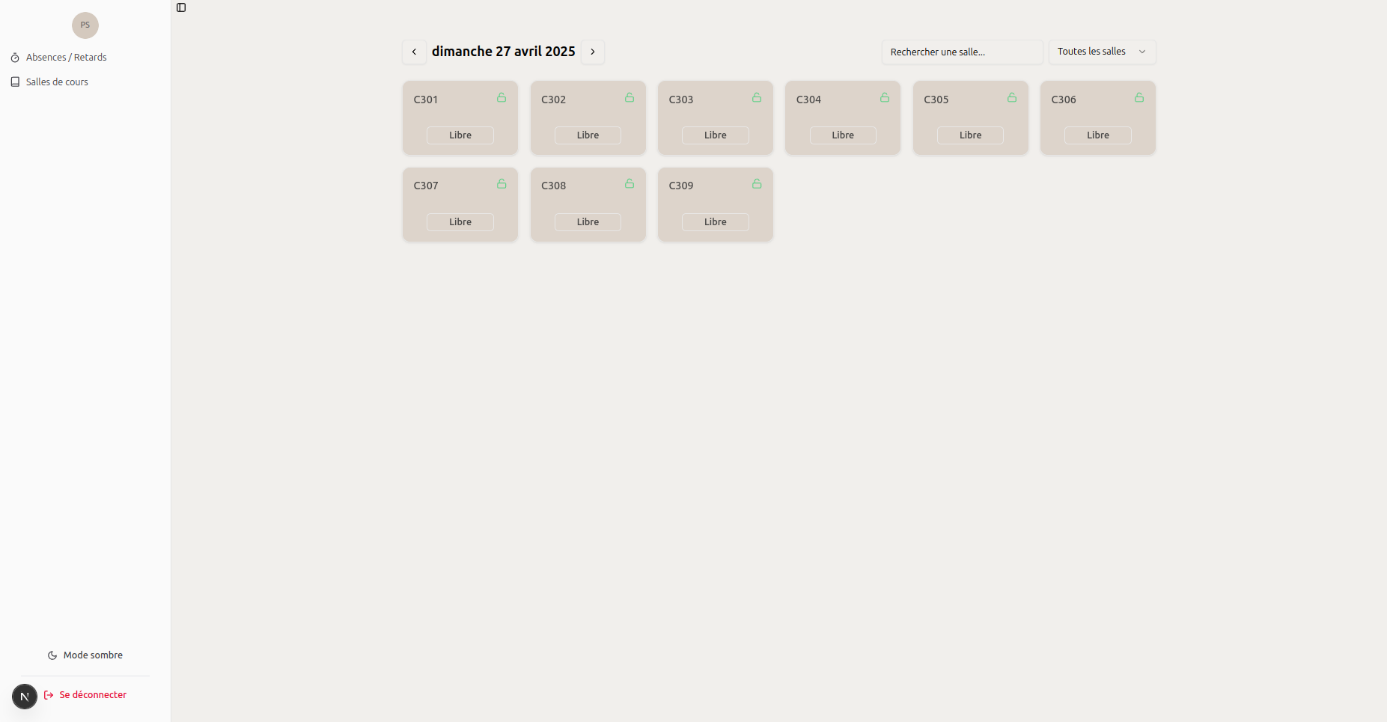
*Figure 15 Affichage des informations de salles*



*Figure 16 Requête GET salle*

*Figure 17 Requête GET activité salle*

En plus des cas d’utilisation terminés, un ajout non demandé a été réalisé : l’ajout d’un « mode clair/mode sombre ».



*Figure 18 Mode clair mode sombre*

## – Test

### 2.5.1 – Test unitaire

Voici un plan de test unitaire, il permet de tester la fonction AbscenceRetardsPage sur la version brute du PSW.

**Plan de test – Fonction AbsencesRetardsPage**

**1 - Identification du test**

**Nom** : Affichage élèves de ses absences et retards  
**Numéro** : TU1

**2 - Référence du module testé**

app/absences/eleve/page.tsx – Fonction AbsencesRetardsPage

**3 - Objectif du test**

S’assurer qu’un élève peut voir ses absences et retards :

* Retour correct des absences et retards
* Affichage correct des absences et retards

**4 - Procédure du test**

* **Initialisation** :
  + Un utilisateur
  + Des retards et absences créé en brut
* **Lancement** : clique sur Absences / Retards
* **Observation** : Affichage des absences et retards

**5 - Résultats attendus**

| **N° Test** | **Condition** | **Résultat attendu** | **Statut attendu** |
| --- | --- | --- | --- |
| 1 | Aucune absence ou retard | Aucune absence ou Aucun retard | Echec |
| 2 | Absence et retard présents | Liste des retards et absences (cours, horaire, durée, motif...) | Succès |

**6 - Moyens à mettre en œuvre**

* **Logiciels** : Oracle VirtualBox
* **Matériel** : Poste informatique

### 2.5.2 – Test d’intégration

Voici un plan de test unitaire, il permet de tester la fonction le login sur la version finale du PSW.

**Plan de test – Login**

**1 - Identification du test**

**Nom** : Login utilisateur   
**Numéro** : TI5

**2 - Référence du module testé**

/psw/login/

**3 - Objectif du test**

L’utilisateur se connecte à son compte personnel

* Envoie des identifiants utilisateurs
* Connecte l’utilisateur au PSW

**4 - Procédure du test**

* **Initialisation** :
  + Un utilisateur élève existe dans la BDD
* **Lancement** : L’utilisateur entre son identifiant dans la case « Identifiant », puis son mot de passe dans la case « Mot de passe » et pour finir clique sur « Se connecter »
* **Observation** : L’utilisateur se connecte au PSW

**5 - Résultats attendus**

| **N° Test** | **Condition** | **Résultat attendu** | **Statut attendu** |
| --- | --- | --- | --- |
| 1 | Aucun compte utilisateur | Impossible de se connecter | Echec |
| 2 | Erreur dans les identifiants | Identifiant ou mot de passe incorrect | Echec |
| 3 | Identifiant correct | L’utilisateur se connecte au PSW | Succès |

**6 - Moyens à mettre en œuvre**

* **Logiciels** : FastAPI, PostgreSQL, express, pm2, pgAdmin
* **Matériel** : Poste informatique connecté au réseau

# III- Conclusion

Pour finir, ce projet m’a permis de développer de nombreuses compétences essentielles, tant sur le plan technique que sur le plan humain. Travailler en équipe m’a appris l'importance de la communication, de l’adaptation aux autres et de l’entraide, des qualités indispensables à la réussite collective. Chaque membre ayant sa propre façon de travailler, il a été nécessaire de trouver un équilibre pour avancer ensemble dans la même direction.

Sur le plan technique, ce projet a été l’occasion pour moi de me familiariser avec le fonctionnement des API et des bases de données, qui constituaient le cœur du projet. J’ai également compris que la réalisation d’un projet ne se fait jamais en une seule tentative : rien n’est parfait du premier coup. Les améliorations, les ajustements et les évolutions font partie intégrante du processus de création. Cette prise de conscience s’est concrétisée à travers l’utilisation du versioning, qui permet de suivre et d’analyser l’évolution progressive du site PSW sur GitHub.

En conclusion, ce projet a été une expérience extrêmement enrichissante, tant sur le plan professionnel que personnel. Il m’a non seulement permis d’acquérir des compétences techniques solides, mais également de mieux comprendre l’importance de la collaboration, de la rigueur et de l’adaptabilité dans le monde de l’informatique.