



Lycée polyvalent St Eloi
9 Avenue Jules Isaac
13100 Aix-en-Provence

Rapport de projet

Rédacteurs :

LONGUET Benjamin (IR)
Zemal Sany (ER)
Abbas Keren (ER)

BTS CIEL 2

Créé le : 25/01/2005

Version en cours : 04

Professeur encadrant : Mme
Agostinelli

TABLE DES MATIERES

INTRODUCTION.....	3
Présentation du projet	3
1°/ Synoptique et contexte.....	3
2°/ Objectifs.....	4
3°/Solutions Techniques.....	5
2°/ Cahier des charges (diagramme des exigences).....	6
3°/ Implémentation physique (diagramme de déploiement).....	6
1°/ Fonctionnalités du système (diagramme de UC)	7
1°/ Tâches IR.....	8
2°/ Tâches ER.....	9
1°/ Fonctionnalités du système (UC + scénarioi)	10
Scénario Nominal.....	10
Modifier les Paramètres :.....	10
Gérer les Paramètres de l'IOT :.....	10
Extensions	10
Mise à jour de la Base de Données :.....	10
Modifier les Paramètres :.....	10
Scénario Nominal.....	11
Données non disponibles :.....	11
Paramètres invalides :.....	11
Connexion IOT perdue :.....	11
2°/ Planning prévisionnel.....	12
III./ Analyse personnelle Etudiant IR	13
dd	Erreur ! Signet non défini.
Conclusion.....	19

INTRODUCTION

Le projet de radiateurs connectés vise à optimiser la consommation électrique et améliorer le confort thermique des occupants du bâtiment SUD CIEL du Lycée Saint-Eloi à Aix-en-Provence. Ce projet s'inscrit dans une démarche de développement durable et d'efficacité énergétique. Le Lycée Saint-Eloi, situé à Aix-en-Provence, est engagé dans une démarche de développement durable et d'efficacité énergétique. Le bâtiment SUD CIEL, utilisé par les étudiants, présente des problèmes de surconsommation électrique et d'inconfort thermique dus à une utilisation anarchique des radiateurs électriques.

Présentation du projet

1°/ Contexte du projet

Comme évoqué précédemment, l'objectif du projet porte sur la conception, le développement et la réalisation d'un système embarqué capable de piloter intelligemment un radiateur connecté en fonction d'un emploi du temps reçu à distance. Ce système a vocation à être utilisé dans des environnements éducatifs ou professionnels afin d'optimiser le confort thermique tout en réduisant la consommation énergétique.

La communication entre le système et le serveur distant s'effectue par connexion Wi-Fi, en utilisant le protocole Web Socket pour la transmission des données en temps réel. Le serveur héberge un emploi du temps sous format CSV, qui est récupéré puis interprété par l'appareil pour définir les plages horaires de fonctionnement du radiateur.

Le système devra être capable de :

- Réceptionner et interpréter un fichier CSV contenant les horaires de présence.
- Allumer ou éteindre un radiateur électrique en fonction de ces données et en fonction d'un seuil de température.
- Conserver l'état du radiateur même en cas de mise en veille du microcontrôleur, grâce à un relais.
- Optimiser la consommation d'énergie en mettant l'ESP32 en mode veille profonde en dehors des périodes de communication.
- Communiquer avec un serveur distant via une connexion Internet, sans contrainte de réseau local.

2°/ Objectifs

Optimisation de la Consommation Électrique

- **Régulation automatique en fonction des besoins réels**
 - Mise en œuvre d'un système d'asservissement thermique basé sur des capteurs et sur l'analyse des périodes d'occupation, permettant de limiter le fonctionnement inutile des radiateurs.
- **Intégration de capteurs pour ajuster la puissance en temps réel**
 - Intégration de capteurs de température intérieurs et extérieurs afin de moduler la puissance délivrée en fonction des conditions réelles.
- **Suivi et analyse de la consommation pour réduire le gaspillage**
 - Enregistrement en base de données des courbes de consommation et de température, avec visualisation graphique, pour identifier les périodes de gaspillage et améliorer les stratégies d'économie d'énergie.

Amélioration du Confort Thermique

- **Température ajustable selon les préférences des occupants**
 - Programmation automatique selon les emplois du temps des cours (via Ecole Directe), avec possibilité d'anticipation de chauffage selon l'inertie thermique des salles.
- **Prise en compte des conditions extérieures pour une gestion optimisée**
 - Ajustement dynamique du pilotage selon les saisons intermédiaires (printemps/automne) pour maintenir un confort thermique constant.
- **Programmation intelligente et commande à distance**
 - Interface web ergonomique permettant un paramétrage en ligne des températures cibles, horaires de chauffage et suivi temps réel.

Sécurité et Fiabilité

- **Garantie d'une utilisation stable et durable**
 - Mise en veille automatique du système via ESP32 pour économiser l'énergie et garantir une longévité optimale de l'électronique embarquée.

3°/ Solutions Techniques

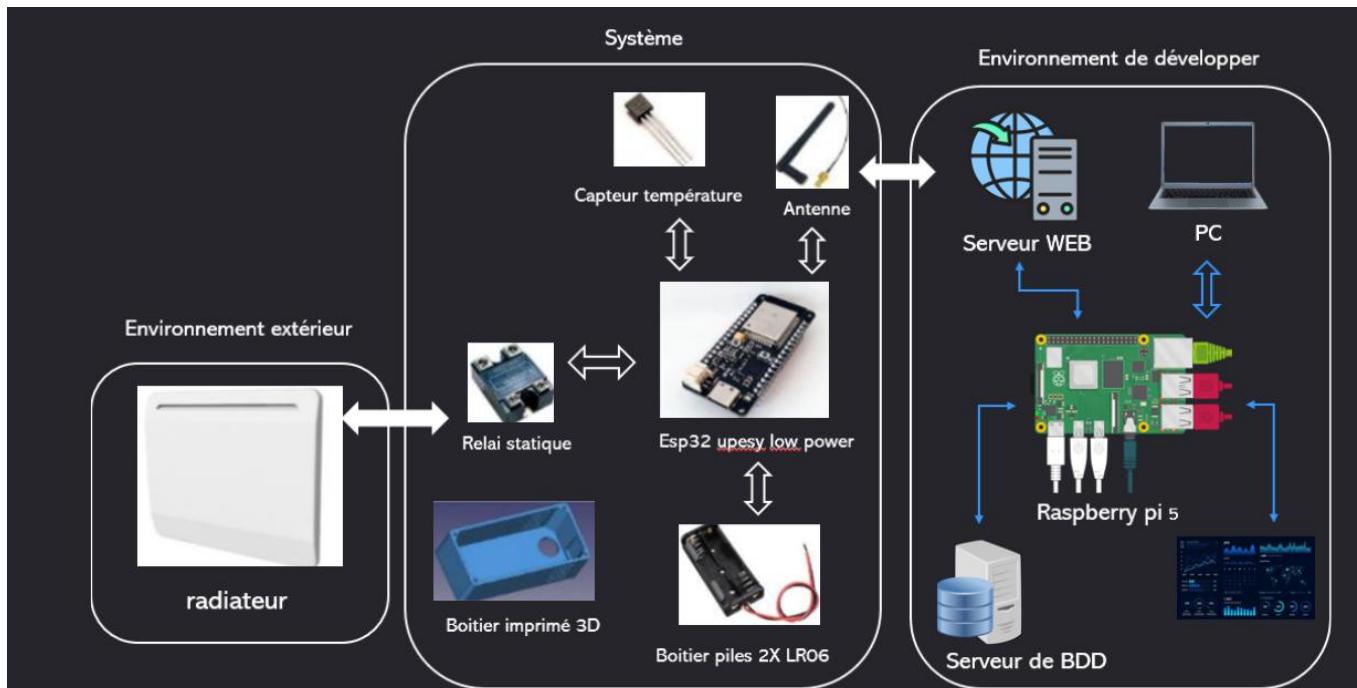
❖ Interfaçage des Modules

- Communication entre les capteurs de température, les actionneurs et l'interface utilisateur
 - Utilisation de capteurs DS18B20 pour la température, relais statiques pour l'activation des radiateurs, et moteurs si besoin de régulation fine.
- Utilisation de protocoles de communication adaptés (Wi-Fi, Bluetooth..)
 - Protocole Wi-Fi pour l'échange de données entre les capteurs, le module ESP32, la base de données et l'interface web. Prise en charge de commandes descendantes (paramètres) et montantes (mesures).

💻 Développement Logiciel

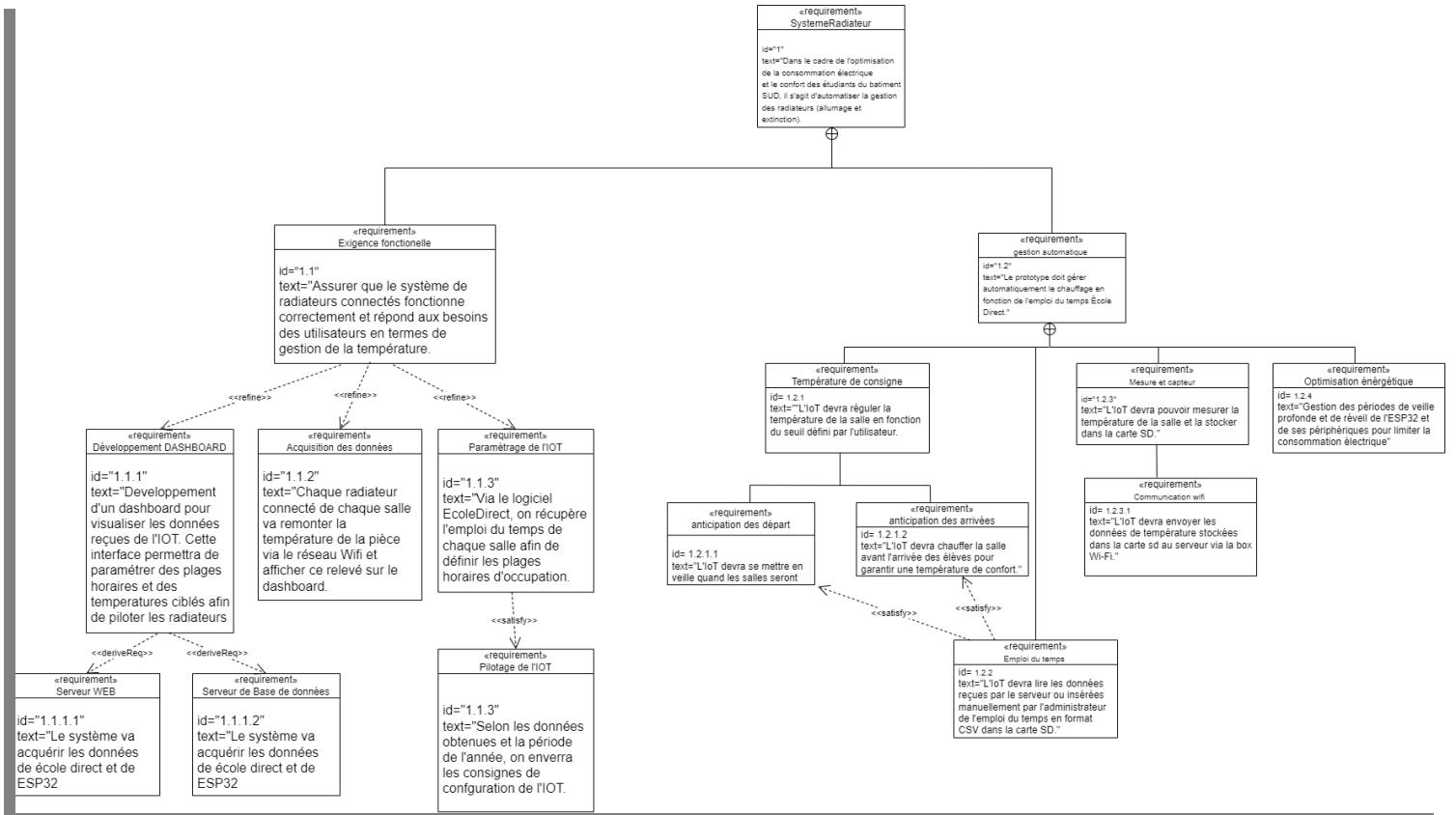
- Interface utilisateur intuitive (application mobile, écran tactile..)
 - Interface web responsive, accessible sur PC et mobile, affichant les données par salle, par période, et outils de paramétrage.
- Algorithmes d'apprentissage pour adapter le chauffage aux habitudes de l'utilisateur
 - Analyse des cycles de chauffe et habitudes d'utilisation pour proposer automatiquement des scénarios optimisés d'économie.

4°/ Synoptique du projet

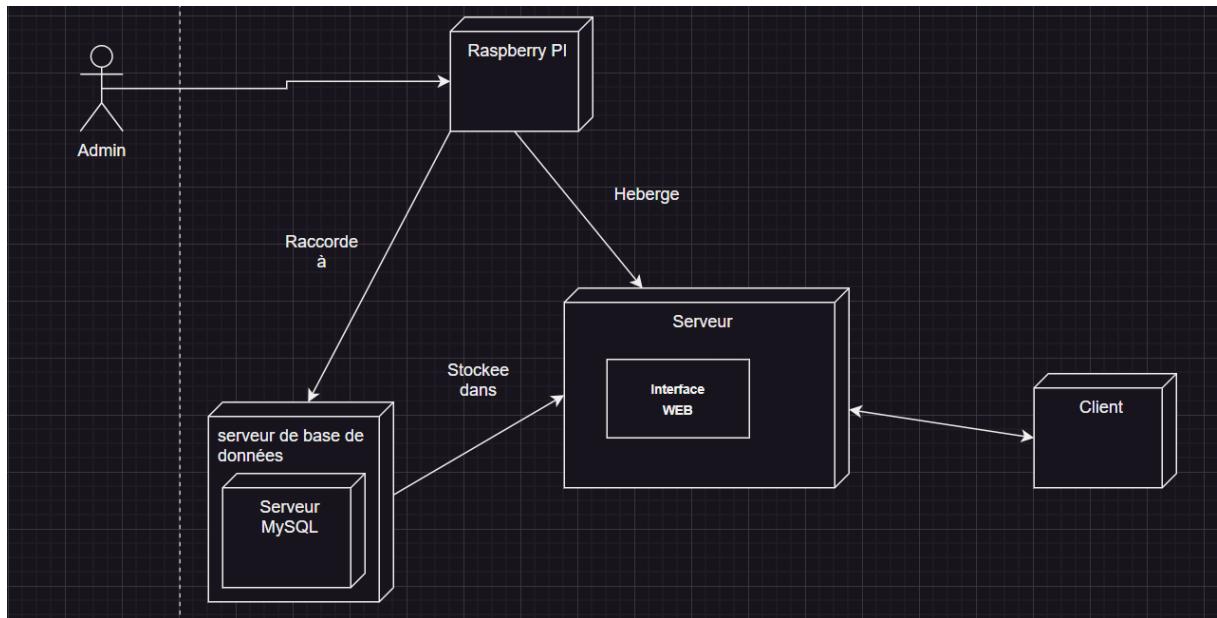


PRESENTATION DIAGRAMME UML

5°/ Cahier des charges (diagramme des exigences)



6°/ Implémentation physique (diagramme de déploiement)



7°/ Fonctionnalités du système (diagramme de UC)

Diagramme de USECASEER



Partage des tâches

8°/ Tâches IR

Développement de l'Interface Web

- Interface de visualisation des données : développement des différents écrans de présentation des données (échelles de temps).
- Interface de paramétrage : développement de l'interface de paramétrage des plages horaires, en fonction des vacances scolaires, de l'emploi du temps, et des températures cibles.

Développement de la Base de Données

- Stockage des données : développement de la base de données pour le stockage des données de température et de consommation électrique.

Protocole de Communication

- Protocole de communication dans le sens entrant : développement du protocole de communication pour la réception des mesures.
- Protocole de communication dans le sens sortant : développement du protocole de communication pour l'envoi des paramètres.

Gestion des Sources

- Outil de gestion des sources : Mise en place de l'outil de gestion des sources (GT) pour le logiciel web et le logiciel embarqué

Table	Action	Lignes	Type	Interclassement	Taille	Perte
Capteurs	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
Consignes	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
emploi_des_temps	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
Logs	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
Mesures	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
Radiateurs	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
Salles	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
Utilisateurs	 Parcourir  Structure  Rechercher  Insérer  Vider  Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
8 tables		65	InnoDB	utf8mb4_general_ci	208,0 kio	0 o
Somme						



9°/ Tâches ER

Interfacer le module température DS18B20 (câblage et logiciel)

- Interfacer le relai statique (câblage et logiciel).
- Fabriquer et intégrer le support (impression 3D).
- Gérer les phases de veille profonde et de réveil de l'esp32 et ses périphériques pour optimiser la consommation.
- Mesurer la consommation électrique du dispositif afin de proposer et dimensionner un mode d'alimentation pertinent (alimentation sur secteur, piles, batteries, super capa, module thermoelectrique..)
- Interfacer une box wifi

Protocole dans le sens entrant (paramétrage périodes de chauffe, mise à l'heure..)

- Réalisation des tests unitaires, puis développement du logiciel embarqué.
- Prototype PCB sur Kicad.
- Fabrication.
- PV/tests et recette.

III./ Analyse personnelle Etudiant IR

10°/ Fonctionnalités du système (UC + scénarii)

- Scénario Nominal

Consulter les Températures/Données des Radiateurs :

- L'utilisateur consulte les données via l'interface
- Le système affiche les données actuelles des radiateurs.
- **Modifier les Paramètres :**
 - L'administrateur modifie les paramètres via l'interface.
 - Le système enregistre les nouveaux paramètres et les applique immédiatement.
- **Gérer les Paramètres de l'IOT :**
 - L'administrateur gère les paramètres de l'IOT via l'interface.
 - Le système enregistre les nouveaux paramètres et les applique immédiatement.
- Extensions

Consulter les Températures/Données des Radiateurs :

- **Mise à jour de la Base de Données :**
 - L'administrateur choisit de mettre à jour la base de données.
 - Le système met à jour la base de données et affiche un message de confirmation.
- **Modifier les Paramètres :**
 - Gérer les Paramètres de l'IOT.
 - L'administrateur choisit de gérer les paramètres de l'IOT après avoir modifié les paramètres des radiateurs.
 - Le système affiche l'interface de gestion des paramètres de l'IOT.

- Scénario Nominal

Consulter les Températures/Données des Radiateurs :

- **Données non disponibles :**

- Si les données ne sont pas disponibles, le système affiche un message d'erreur indiquant que les données ne peuvent pas être consultées.

Modifier les Paramètres :

- **Paramètres invalides :**

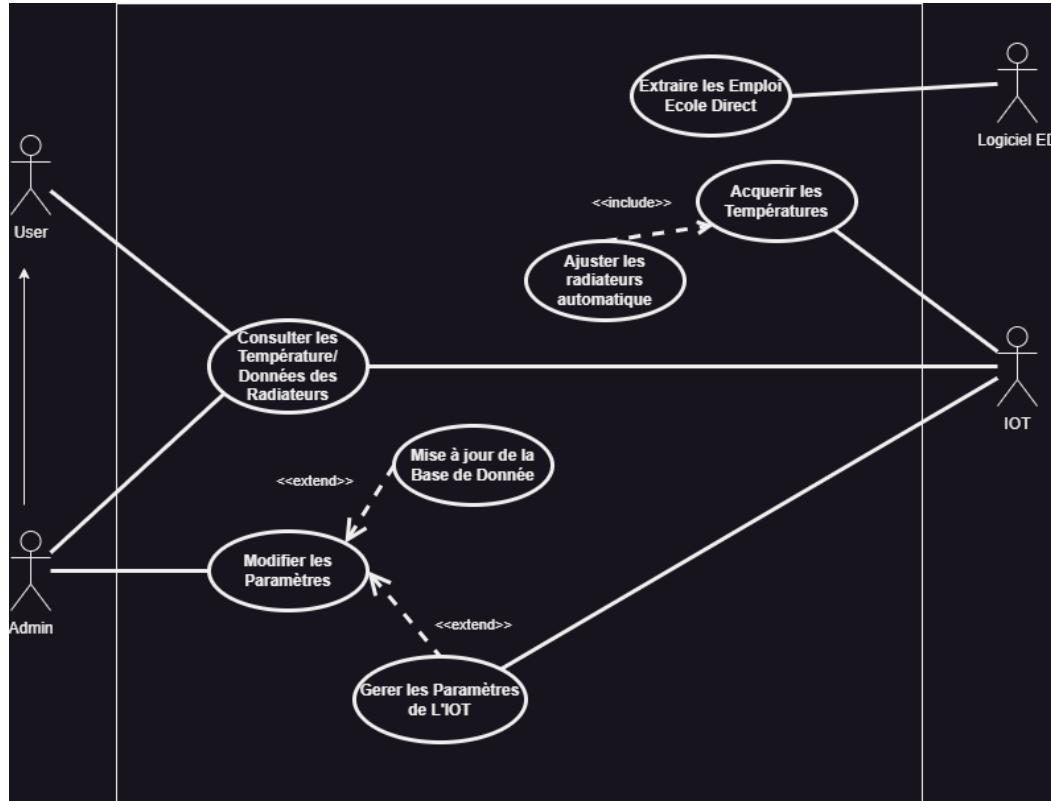
- Si les nouveaux paramètres sont invalides, le système affiche un message d'erreur indiquant que les paramètres ne peuvent pas être enregistrés.

Gérer les Paramètres de l'IOT :

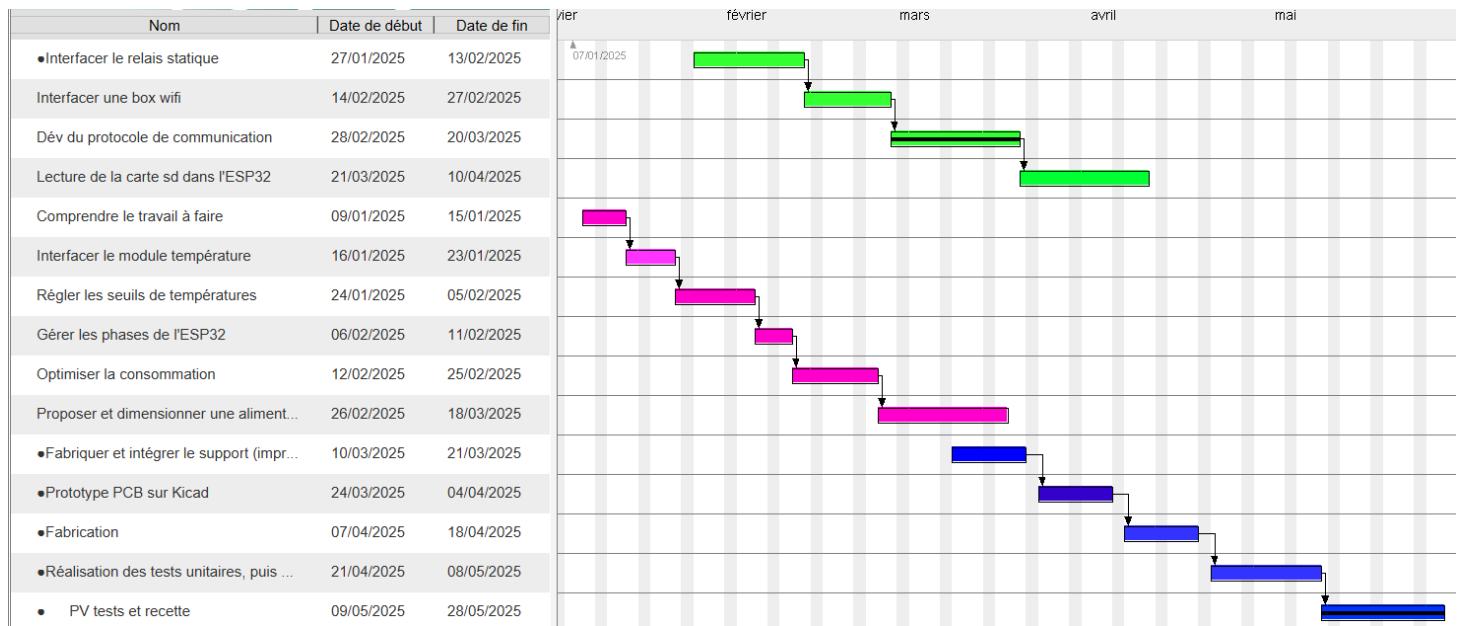
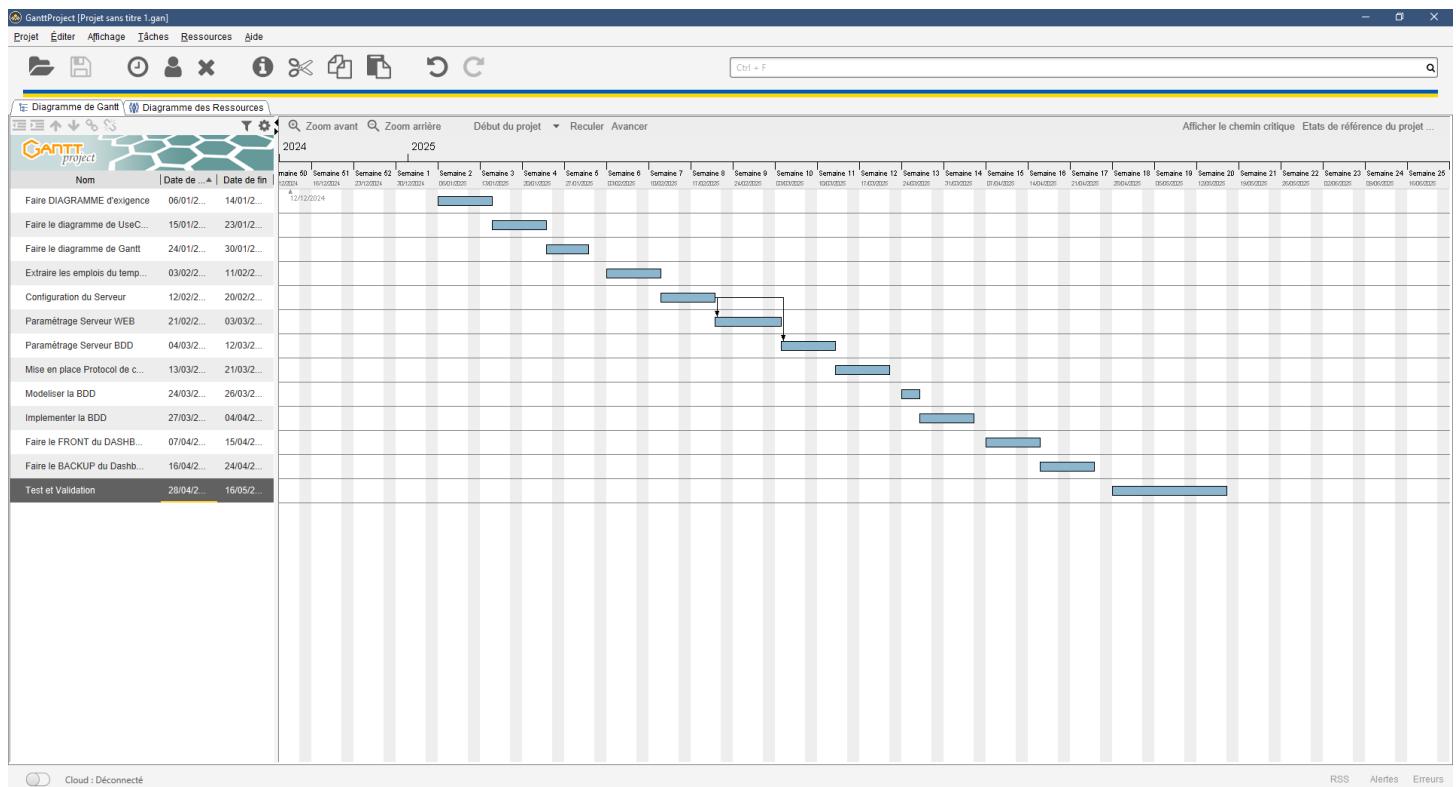
- **Connexion IOT perdue :**

- Si la connexion avec l'IOT est perdue, le système affiche un message d'erreur indiquant que la connexion a été perdue et que les paramètres ne peuvent pas être appliqués.

Diagramme de USECASEIR



11°. / Planning prévisionnel



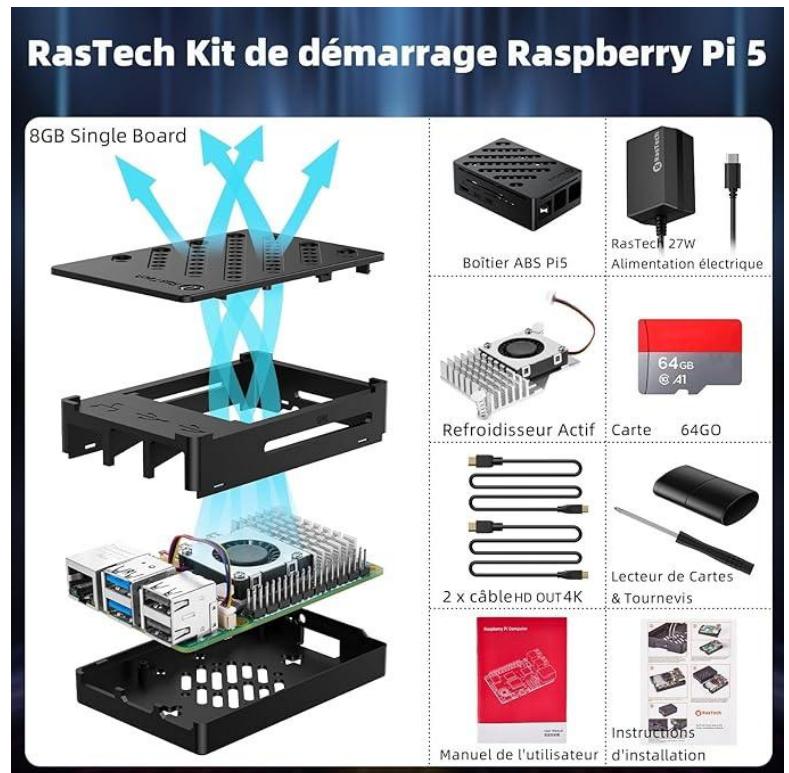
III./ Développement du projet

12/ Mise en place du Raspberry

Matériels Utilisé :



Raspberry Pi 5

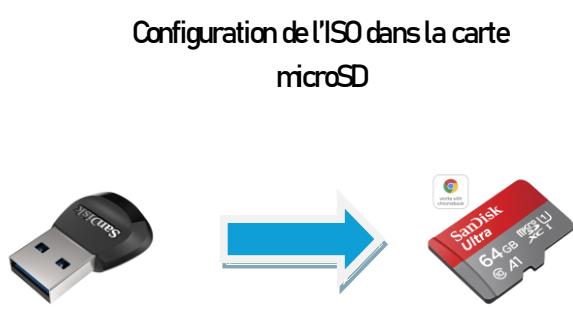


Matériel :

- Bâtier Pi5
- RasTech 27W Alimentation électrique (Adaptateur secteur de 27W)
- Refroidisseur Actif (Dissipateur thermique avec ventilateur)
- Carte 64GO (Carte microSD de 64 Go)
- 2 x Câble HD OUT 4K (Deux câbles HDMI pour sortie 4K)
- Lecteur de Cartes & Tournevis
- Manuel de l'utilisateur (Guide d'utilisation)

13°/ Configuration du Raspberry Pi :

- Téléchargement de l'ISO sur une carte SD: imager_1.8.5.exe



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)

To install on **Raspberry Pi OS**, type
`sudo apt install rpi-imager`
in a Terminal window.

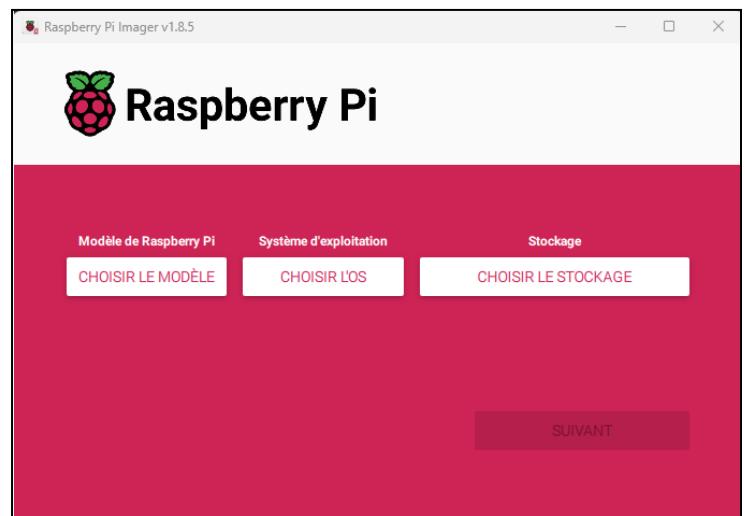
Téléchargement de l'ISO depuis mon PC portable

Lancement du logiciel :

- Choisir l'OS : sélectionner Raspberry Pi OS (64-bit).
- Choisir le support de stockage : votre carte microSD connecté en USB

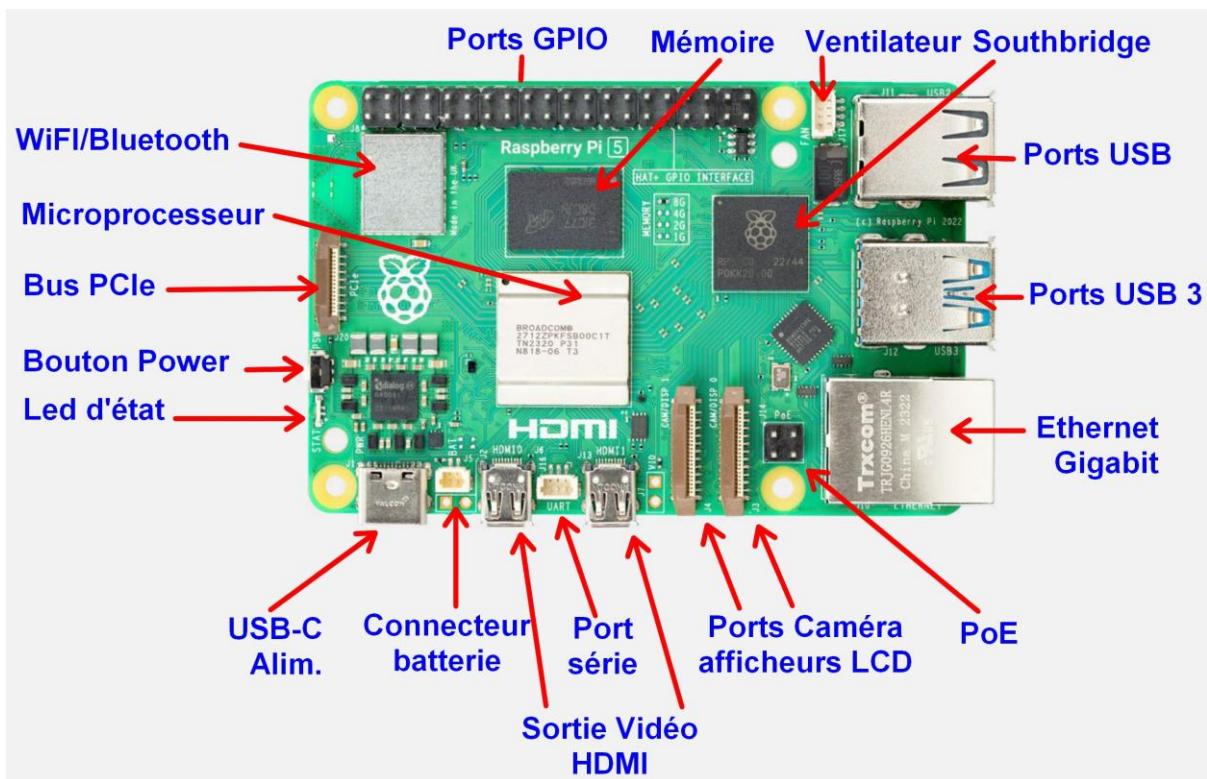
Cliquer sur l'icône ☰ (paramètres avancés) :

- Activer SSH
- Configurer W-Fi (SSID+mot de passe)
- Définir le nom d'hôte (**benlt**)
- Définir le mot de passe utilisateur : **123456789**



💡 Démarrage du Raspberry Pi

- Insérer la carte SD dans le Raspberry Pi 5 ou brancher le SSD
- Démarrer le Pi : il démarre automatiquement sur Raspberry Pi OS.
- Suivre l'assistant de configuration :
 - Sélection de la langue/clavier
 - Mise à jour du système
 - Connexion Wi-Fi (si pas déjà configurée)



Cette photo présente une vue annotée du Raspberry Pi 5, identifiant clairement ses composants et connecteurs principaux tels que le processeur, la mémoire, les ports GPIO, USB, HDMI, Ethernet, et les interfaces caméra

Connection via le Protocol SSH ou VNC

- Raspberry Pi 5 sous Raspberry Pi OS (64-bit de préférence)
- SSH ou VNC activé
- Connexion réseau (Wi-Fi ou Ethernet)
- Adresse IP locale ou nom d'hôte du Raspberry Pi (**192.168.32.104**)
- Logiciels recommandés:
 - Pour SSH PuTTY (Windows) ou Terminal (Linux)
 - Pour VNC: VNC Viewer

🔍 Trouver l'adresse IP du Raspberry Pi

- Aller dans CMD en mode Administrateur : Taper « Ipconfig », on trouve ainsi l'adresse IP de l'appareil.
Utilisez un scanner IP comme « **Angry IP Scanner** » ou connectez-vous à votre box pour voir les appareils connectés.

Mettre la plage d'adresse IP en fonction de l'appareil qui est connecté à la Box.

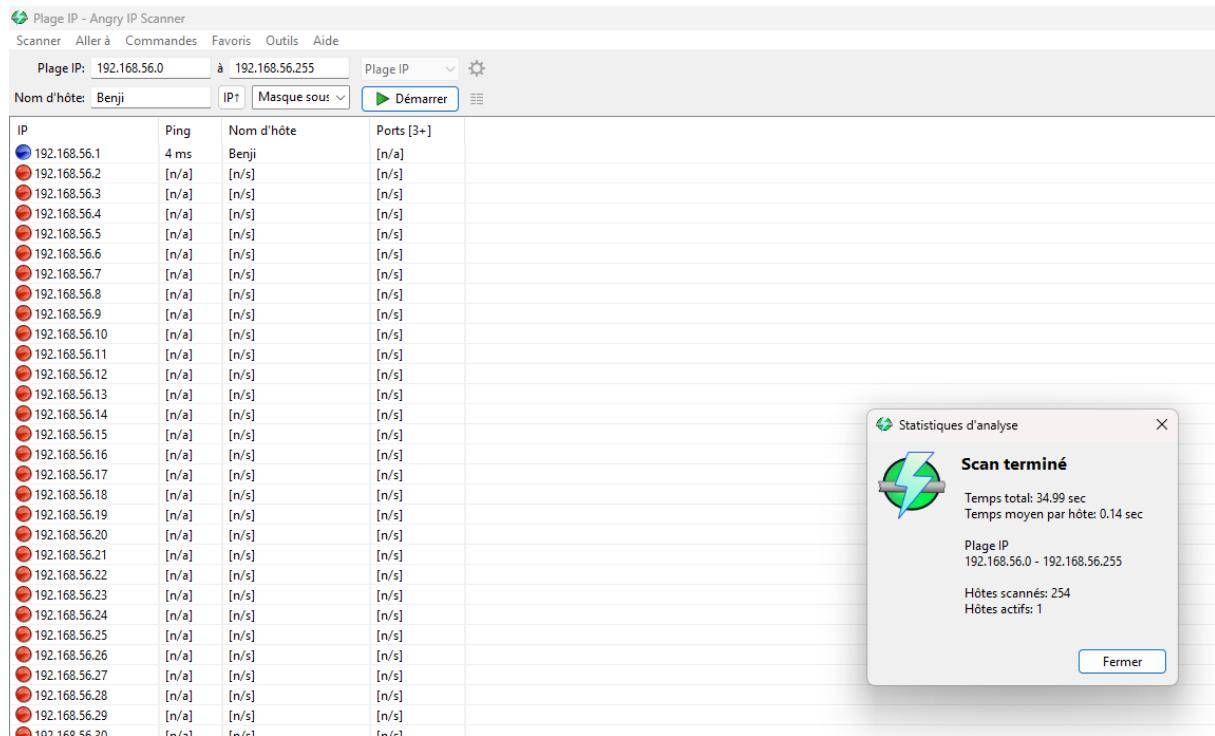


Illustration de recherche d'appareils via le logiciel Angry IP Scanner

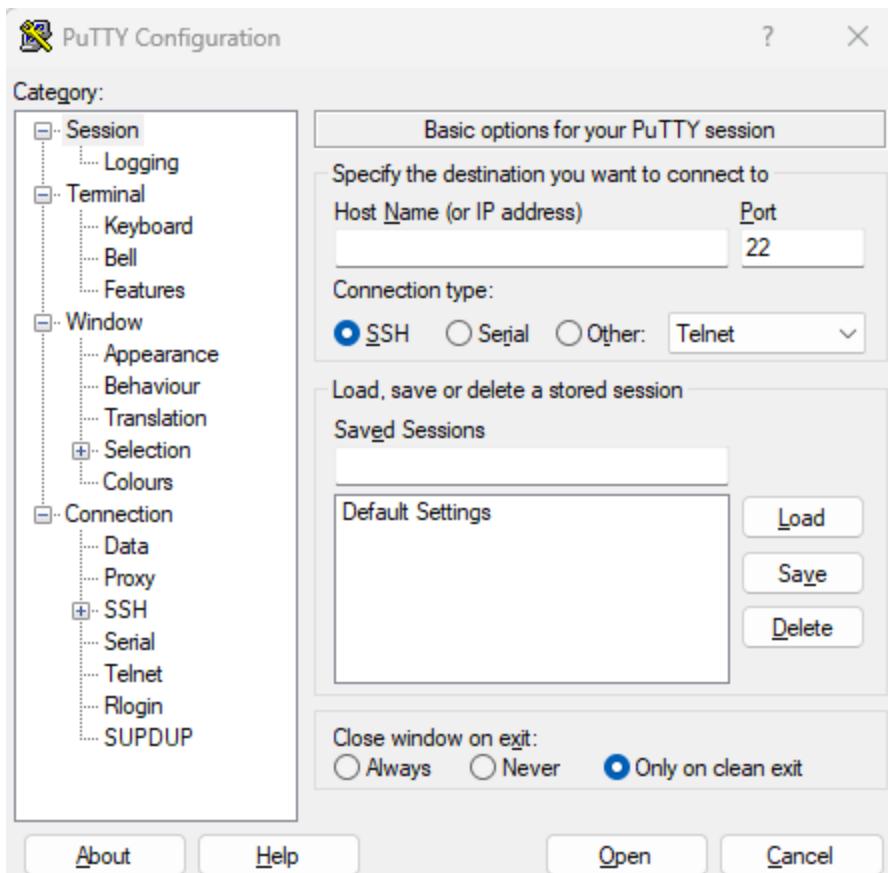
Connexion en SSH (terminal à distance)

- Je me suis servi de ces deux outils pour me connecter sur le Raspberry Pi

Méthode 1 :

Sur Windows avec PuTTY:

- Ouvrir PuTTY
- Adresse IP ou raspberrypi.local dans Host Name
- Port : 22
- Connexion → login : pi, mot de passe défini précédemment

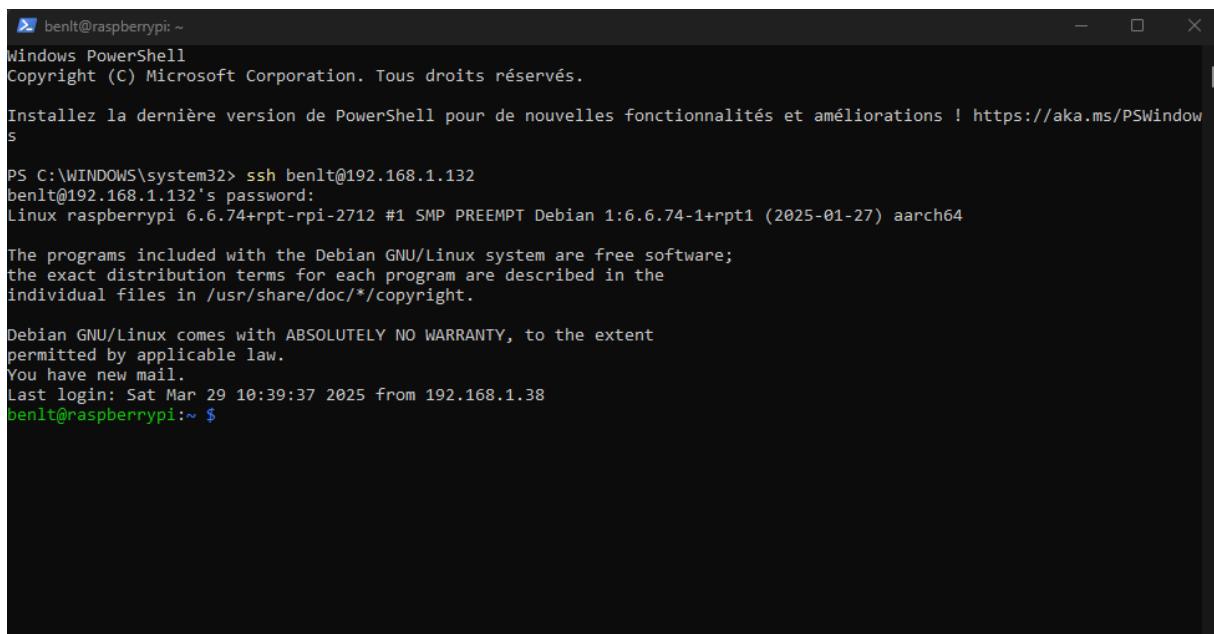


Méthode 2 :

Sur Windows avec cmd ou PowerShell :

```
1 ssh benlt@192.168.1.xxx  
2
```

14°/ Création du serveur WEB :



```
benlt@raspberrypi: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> ssh benlt@192.168.1.132
benlt@192.168.1.132's password:
Linux raspberrypi 6.6.74+rpi-2712 #1 SMP PREEMPT Debian 1:6.6.74-1+rpi1 (2025-01-27) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Sat Mar 29 10:39:37 2025 from 192.168.1.38
benlt@raspberrypi:~ $
```

Terminal dans lequel je vais exécuter toutes les commandes

- **Mise à jour du Raspberry**
 - **SUDO APT UPDATE**: Mise à jour des paquets disponibles
 - **SUDO APT UPGRADE**: Installe les mises à jour des paquets déjà installé
- **Installation d'Apache**
 - **SUDO APT INSTALL APACHE2**: Installation d'un serveur WEB open-source (Debian)
 - **SUDO APT INSTALL PHP LIBAPACHE2-MOD-PHP**: Installation de PHP dans le serveur WEB



15°/ Création du serveur BDD :

- **SUDO SERVICE APACHE2 RESTART** : Cette commande redémarre le serveur Apache sous Linux, en arrêtant puis en relançant le service
- **SUDO MYSQL --USER=ROOT** : Cette commande permet d'accéder à MySQL en tant qu'utilisateur root avec les priviléges administratifs.
- **DROP USER 'ROOT'@'LOCALHOST'** ; : Supprime l'utilisateur root de MySQL en local.
- **CREATE USER 'ROOT'@'LOCALHOST' IDENTIFIED BY 123456789** ; : Crée l'utilisateur root avec un mot de passe.
- **GRANT ALL PRIVILEGES ON TO 'ROOT'@'LOCALHOST'** ; : Accorde tous les priviléges à root sur toutes les bases.
- **SUDO APT INSTALL PHPMYADMIN**: Installe phpMyAdmin sur le système.  



Page d'accueil PhpMyAdmin

Table	Action	Lignes	Type	Interclassement	Taille	Péri
Capteurs		0	InnoDB	utf8mb4_general_ci	32,0 kio	
Consignes		0	InnoDB	utf8mb4_general_ci	32,0 kio	
emploi_du_temps		0	InnoDB	utf8mb4_general_ci	16,0 kio	
Logs		0	InnoDB	utf8mb4_general_ci	32,0 kio	
Mesures		0	InnoDB	utf8mb4_general_ci	32,0 kio	
Radiateurs		0	InnoDB	utf8mb4_general_ci	32,0 kio	
Salles		0	InnoDB	utf8mb4_general_ci	16,0 kio	
Utilisateurs		0	InnoDB	utf8mb4_general_ci	16,0 kio	
8 tables	Somme	0	InnoDB	utf8mb4_general_ci	208,0 kio	

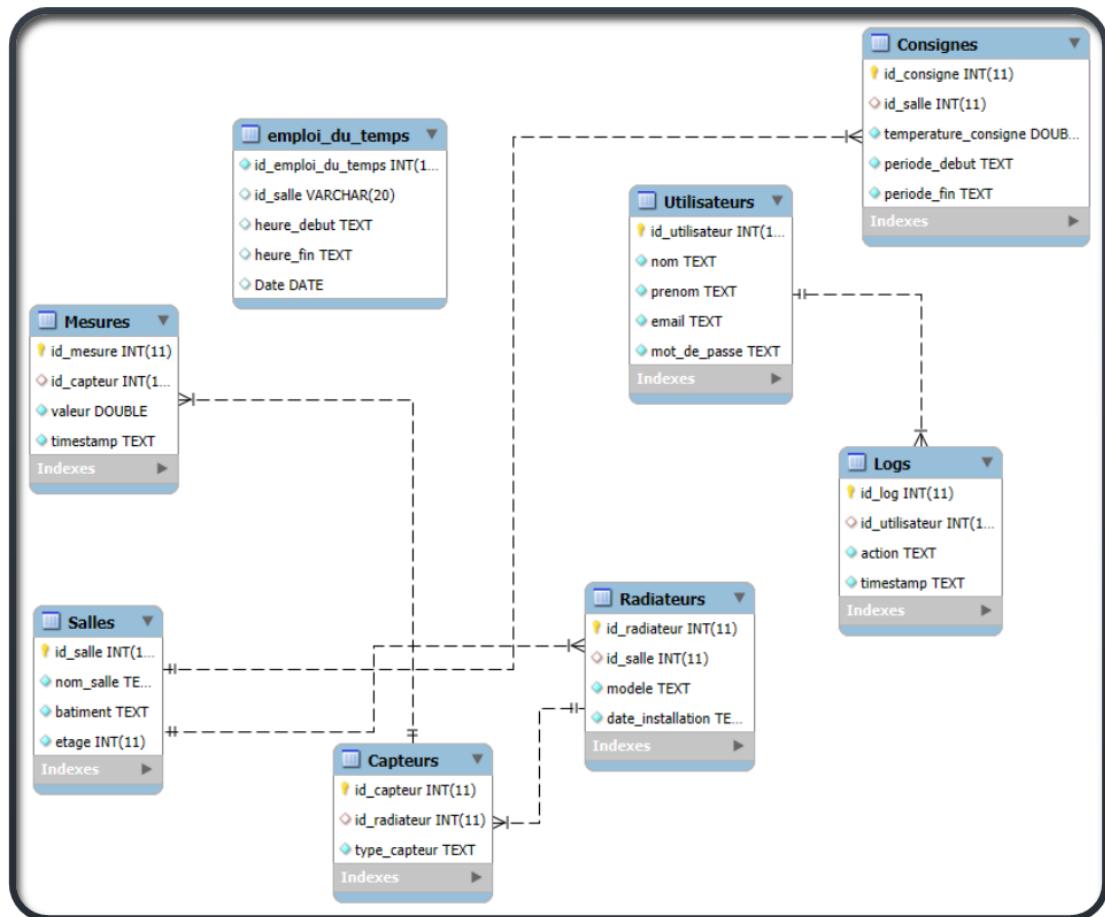
Illustration de la BDD

16°/ MODELISATION DE LA BASE DE DONNEE :

- Voici une modélisation des tables représentant la base de données PRO

Table de la BDD

- Capteurs
- Consignes
- Emploi_du_temps
- Logs
- Mesures
- Radiateurs
- Salles
- Utilisateurs



Création du serveur BDD :

- **SUDO SERVICE APACHE2 RESTART** : Cette commande redémarre le serveur Apache sous Linux, en arrêtant puis en relançant le service
- **SUDO MYSQL --USER=ROOT** : Cette commande permet d'accéder à MySQL en tant qu'utilisateur root avec les priviléges administratifs.
- **DROP USER'ROOT'@'LOCALHOST'** ; : Supprime l'utilisateur root de MySQL en local.
- **CREATE USER'ROOT'@'LOCALHOST' IDENTIFIED BY 123456789** ; : Crée l'utilisateur root avec un mot de passe.
- **GRANT ALL PRIVILEGES ON TO'ROOT'@'LOCALHOST'** ; : Accorde tous les priviléges à root sur toutes les bases.
- **SUDO APT INSTALL PHPMYADMN**: Installe phpMyAdmin sur le système.



Page d'accueil PhpMyAdmin

Illustration de la BDD

- Importation dans la BDD

Importation dans la base de données « PROJET »

Fichier à importer :

Le fichier peut être compressé (gzip, bz2, zip) ou non.
Le nom du fichier compressé doit se terminer par .[format].[compression]. Exemple : .sql.zip

Parcourir les fichiers : (taille maximale : 2 048Mo)

Choisir un fichier : Aucun fichier n'a été sélectionné.

Il est également possible de glisser-déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier :

utf-8

Importation partielle :

Permettre l'interruption de l'importation pour respecter la limite de temps définie dans PHP
Ceci peut faciliter l'importation des fichiers volumineux, mais peut aussi affecter la stabilité des transactions.

Ignorer le nombre de requêtes (pour SQL), à partir du début :

0

Autres options :

Activer la vérification des clés étrangères

Format :

SQL

Options spécifiques au format :

Mode de compatibilité SQL :
NONE

Ne pas utiliser AUTO_INCREMENT pour la valeur zéro

Importer

-Téléchargement de la modélisation en fichier SQL

- Importation du fichier SQL dans la Base de données

- Pour finir, cliquer sur importer

17°/ Écriture du Script de récupération

	Lun 24 Mar	Mar 25 Mar	Mer 26 Mar	Jeu 27 Mar	Ven 28 Mar	Sam 29 Mar	Dim 30 Mar
07:00							
08:00		08:10 - 11:45En SUD 08 ELECTRON.&COM TSCIEL_2_ER GUILLEMIN A.		08:10 - 09:55En SUD PHYSIQUE TSCIEL_2_ER MERABET H.	08:10 - 09:50En SUD SC. TECHNIQUES IND. UFA-CIEL1_ER GUILLEMIN A.		
09:00	09:00 - 10:55En SUD SC. TECHNIQUES IND. UFA-CIEL1_ER GUILLEMIN A.		09:00 - 13:25En SUD SC. TECHNIQUES IND. TSCIEL1_ER CAZALIX C.				
10:00				10:05 - 10:55En SUD SC. TECHNIQUES IND. UFA-CIEL1_ER			
11:00	10:55 - 1 PHYSI TSCIEL GUILLEMIN A.	10:55 - 1 ELECT TSCIEL GUILLEMIN A.					
12:00							
13:00	12:35 - 16:55En SUD 08 ELECTRON.&COM TSCIEL2_ER GUILLEMIN A.	12:35 - 16:55En SUD 08 SC. TECHNIQUES IND. TSCIEL1_ER CAZALIX C.		13:25 - 15:55En SUD 08 ELECTRON.&COM TSCIEL2_ER GUILLEMIN A.	12:35 - 1 ELECT STS GUILLEMIN A.	12:35 - 1 MATHE STS GUILLEMIN A.	
14:00			13:50 - 1 PHYSI TSCIEL MERABET H.	13:50 - 1 PHYSI TSCIEL MERABET H.			
15:00			15:05 - 1 PHYSI TSCIEL GUILLEMIN A.	15:05 - 1 SC. TECHN. IND. STS CIEL 1ère Année CAZALIX C.			
16:00			15:55 - 1 PHYSI TSCIEL MERABET H.				
17:00							

Téléchargement automatique du lien ICS à partir de l'emploi du temps sur Ecole Direct



INTÉGRER L'AGENDA DANS UNE APPLICATION TIERCE



Récupérer l'adresse de l'agenda au format iCal pour pouvoir y accéder à partir d'autres applications.

Copier l'url dans le presse-papier

- Mise en place du fichier ICS dans le répertoire cd /var/www/PRO du Raspberry Pi
 - 52485270596a63764e305978533278564e4574585245704d566e5252626c64784e5863.ics



Lien ICS copié sur le logiciel Ecole Direct

Script Python pour transférer le fichier ICS en fichier CSV :

```
import requests
import icalendar
import datetime
import csv
import pytz

# Configuration
ICAL_URL = "https://api.ecoledirecte.com/v3/ical/E/3436/4e5656725a32466d656d787a544552714d6a4a6f6457564959315247563278684d79394d65433933.ics"
ICAL_FILE = "emploi_du_temps.ics"
CSV_FILE = "occupation_salles.csv"
DATE_DEBUT = datetime.date(2025, 1, 1) # Début : Janvier
DATE_FIN = datetime.date(2025, 5, 31) # Fin : Mai
```

- Requests : pour télécharger le fichier iCal depuis une URL.
- Icalendar : pour lire et analyser le fichier iCal.
- Datetime : pour gérer les dates et heures.
- Csv : pour écrire les données dans un fichier CSV.
- Pytz : pour gérer les fuseaux horaires (conversion en heure de Paris).

- ICAL_URL : lien pour télécharger le fichier iCal.
- ICAL_FILE : nom du fichier où sera enregistré l'emploi du temps.
- CSV_FILE : nom du fichier CSV de sortie.
- DATE_DEBUT et DATE_FIN : période de filtrage des événements (de janvier à mai).

- Téléchargement du fichier iCal

```
def telecharger_emploi_du_temps():
    """Télécharge le fichier iCal depuis l'API École Directe"""
    print("⬇️ Téléchargement de l'emploi du temps...")
    response = requests.get(ICAL_URL)
```

Envoie une requête HTTP pour télécharger le fichier iCal.

- Envoie une requête HTTP pour télécharger le fichier iCal.

```
if response.status_code == 200:
    with open(ICAL_FILE, "wb") as f:
        f.write(response.content)
    print(f"✅ Emploi du temps enregistré sous '{ICAL_FILE}'")
    return True
else:
    print(f"❌ Erreur lors du téléchargement : {response.status_code}")
    return False
```

- Si le téléchargement réussit (status_code=200), le fichier est sauvegardé.
- Sinon, affiche une erreur.

Envoie une requête HTTP pour télécharger le fichier iCal.

```
def convertir_en_heure_locale(dt):
    """Convertit la date UTC en heure locale (Fuseau horaire de Paris)"""
    if isinstance(dt, datetime.datetime): # Vérifie si c'est bien un objet datetime
        paris_tz = pytz.timezone("Europe/Paris")
        if dt.tzinfo is None: # Si la date n'a pas de timezone, on la suppose UTC
            dt = pytz.utc.localize(dt)
        return dt.astimezone(paris_tz)
    return None
```

- Vérifie si l'objet `dt` est bien une date/heure.
- Ajoute le fuseau UTC si l'il n'existe pas.
- Convertit en heure de Paris.
- Extraction des horaires des salles

```
def extraire_horaires_salles(salles_cibles):
    """Extrait et trie les horaires des salles demandées entre janvier et mai"""
    occupation_salles = []
```

- Initialise une liste `occupation_salles` pour stocker les créneaux occupés.

```
try:
    with open(ICAL_FILE, "rb") as f:
        calendar = icalendar.Calendar.from_ical(f.read())
```

- Ouvre et lit le fichier iCal pour extraire les événements.

```
for event in calendar.walk("VEVENT"):
    start = event.get("DTSTART").dt # Heure de début
    end = event.get("DTEND").dt # Heure de fin
    salle = event.get("LOCATION")
```

- Parcourt chaque événement `VEVENT` du fichier.
- Récupère la date de début, la date de fin et la salle concernée.

```
if salle:
    salle = salle.upper()
```

- Convertit le nom de la salle en majuscules pour uniformiser.

```
if salle in salles_cibles and isinstance(start, datetime.datetime):
    date_cours = start.date()
    if DATE_DEBUT <= date_cours <= DATE_FIN:
```

- Vérifie si la salle fait partie de celles recherchées.
- Vérifie si la date est bien entre janvier et mai.

```
occupation_salles.append([
    salle,
    date_cours, # Garde l'objet datetime pour le tri
    convertir_en_heure_locale(start).strftime("%H:%M"),
    convertir_en_heure_locale(end).strftime("%H:%M")
])
```

- Ajoute la salle, la date et les horaires (convertis en heure locale) à la liste.

```
except Exception as e:
    print(f"X Erreur lors de l'extraction des données : {e}")
```

- Capture les erreurs et affiche un message si l'extraction échoue.

```
occupation_salles.sort(key=lambda x: (x[1], x[2]))
```

- Trie les événements d'abord par date, puis par heure de début.

```
for item in occupation_salles:
    item[1] = item[1].strftime("%d/%m/%Y")
```

- Formate la date pour qu'elle soit plus lisible (JJ/MM/AAAA).

```
def generer_csv(occupation_salles, salles_cibles):
    """Génère un fichier CSV avec les horaires des salles triées"""
    with open(CSV_FILE, mode="w", newline="", encoding="utf-8") as file:
        writer = csv.writer(file)
        writer.writerow(["Salle", "Date", "Heure de début", "Heure de fin"])
```

- Ouvre le fichier CSV en mode écriture.
- Ajoute la ligne d'en-tête avec les colonnes.

```
if not occupation_salles:  
    print("⚠️ Aucune donnée trouvée pour les salles sélectionnées.")  
    return
```

- Si aucune donnée n'est disponible, affiche un message d'avertissement et arrête la fonction.

```
for reservation in occupation_salles:  
    writer.writerow(reservation)  
  
print(f"☑ CSV enregistré sous '{CSV_FILE}'")
```

- Écrit chaque ligne dans le fichier CSV.
- Affiche un message de confirmation après la création du fichier.

```
if __name__ == "__main__":  
    if telecharger_emploi_du_temps():  
        ...
```

- Vérifie si le téléchargement du fichier iCal a réussi avant de continuer.

```
salles_input = input("💡 Entrez les noms des salles (séparés par des virgules) : ")  
salles_cibles = [s.strip().upper() for s in salles_input.split(",")]
```

- Demande à l'utilisateur de saisir les noms des salles voulues.
- Nettoie et convertit en majuscules.

```
occupation_salles = extraire_horaires_salles(salles_cibles)  
generer_csv(occupation_salles, salles_cibles)
```

- Extrait les horaires des salles sélectionnées.
- Génère le fichier CSV avec ces informations.

```
C:\Users\benlt\OneDrive\Bureau\PRO>python Fichier.py  
⬇ Téléchargement de l'emploi du temps...  
☑ Emploi du temps enregistré sous 'emploi_du_temps.ics'  
💡 Entrez les noms des salles (séparés par des virgules) : SUD 07  
☑ CSV enregistré sous 'occupation_salles.csv'
```

- Exécution de 'Fichier.py' dans l'invite de commande CMD

18°/ Voici une illustration du fichier CSV :

Salle	Date	Heure de debut	Heure de fin
SUD	07/01/2025	08:10:12:35	
SUD	07/09/2025	08:10:10:55	
SUD	07/09/2025	13:25:15:05	
SUD	07/13/2025	08:10:09:00	
SUD	07/13/2025	12:35:15:55	
SUD	07/14/2025	08:10:11:45	
SUD	07/16/2025	08:10:10:55	
SUD	07/16/2025	13:25:15:05	
SUD	07/20/2025	12:35:16:55	
SUD	07/21/2025	08:10:12:35	
SUD	07/23/2025	08:10:10:55	
SUD	07/23/2025	13:25:15:05	
SUD	07/27/2025	08:10:09:00	
SUD	07/27/2025	12:35:16:55	
SUD	07/28/2025	08:10:12:35	
SUD	07/30/2025	08:10:10:55	
SUD	07/30/2025	13:25:15:05	
SUD	07/03/2025	12:35:16:55	
SUD	07/04/2025	08:10:12:35	
SUD	07/24/2025	08:10:09:00	
SUD	07/24/2025	12:35:16:55	
SUD	07/25/2025	08:10:12:35	
SUD	07/27/2025	08:10:10:55	
SUD	07/27/2025	13:25:15:05	
SUD	07/03/2025	12:35:16:55	
SUD	07/04/2025	08:10:12:35	
SUD	07/06/2025	08:10:10:55	
SUD	07/06/2025	13:25:15:05	
SUD	07/10/2025	08:10:09:00	
SUD	07/10/2025	12:35:16:55	

- ILLUSTRATION DU FICHIER
CSV TRANSFERE DU
LOGICIEL ECOLEDIRECT
SUR MON PC GRACE A UN
SCRIPT PYTHON
- MISE EN AVANT DES
CHAMPS DE LA TABLE
EMPLOI_DU_TEMPS :
 - SALLE
 - DATE
 - HEURE_DEBUT
 - HEURE_FIN

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id_emploi du temps	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 id_salle	varchar(20)	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/>	3 heure_debut	text	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/>	4 heure_fin	text	utf8mb4_general_ci		Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/>	5 Date	date			Oui	NULL			Modifier Supprimer Plus

Voici la Table Emploi_du_temps dans la base de Données

Résumé

- ✓ Télécharge un emploi du temps au format iCal.
- ✓ Extrait les horaires des salles sélectionnées.
- ✓ Convertit les heures en fuseau horaire de Paris.
- ✓ Trie les événements par date et heure de début.
- ✓ Enregistre les résultats dans un fichier CSV.

19°/ Fichier CSV dans la Base de données :

- Activation des erreurs PHP

```
error_reporting(E_ALL);
ini_set('display_errors', 1);
```

- Error_reporting(E_ALL); : Active l'affichage de tous les types d'erreurs.
- Ini_set('display_errors', 1); : Permet d'afficher ces erreurs directement sur la page (utile en mode développement).

- Connexion à la base de données

```
// Connexion à la base de données
$host = "192.168.1.132";
$user = "root";
$pass = "123456789";
$dbname = "PROJET";

$conn = new mysqli($host, $user, $pass, $dbname);
if ($conn->connect_error) {
    die("Échec de la connexion : " . $conn->connect_error);
}
```

Définit les informations de connexion à la base de données :

- Adresse IP du serveur MySQL → 192.168.1.132
- Nom d'utilisateur → root
- Mot de passe → 123456789
- Nom de la base de données → PROJET

- New MySQL(\$host, \$user, \$pass, \$dbname); : Crée une connexion MySQL.
- If (\$conn->connect_error) : Vérifie si la connexion a échoué.
- die("Échec de la connexion : ". \$conn->connect_error); : Stoppe l'exécution du script et affiche l'erreur si la connexion échoue.

- Vérification de l'existence du fichier CSV

```
$file_name = "occupation_salles.csv";
if (!file_exists($file_name)) {
    die("Le fichier CSV est introuvable !");
}
```

- Définit le nom du fichier CSV à lire.
- File_exists(\$file_name) : Vérifie si le fichier existe
- die("Le fichier CSV est introuvable!"); : Arrête l'exécution du script si le fichier est introuvable.

- Ouverture et lecture du fichier CSV

```
if (($handle = fopen($file_name, "r")) !== FALSE) {
    fgetcsv($handle); // Ignorer la première ligne (en-têtes)
```

- `fopen($file_name, "r")` : Ouvre le fichier CSV en mode lecture (r).
- `!== FALSE` : Vérifie si le fichier a bien été ouvert.
- `fgetcsv($handle)` : Lit la première ligne du fichier (les en-têtes) et l'ignore.

- Lecture et insertion des données

```
while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
```

- `fgetcsv($handle, 1000, ",")` : Lit une ligne du fichier CSV et la stocke sous forme de tableau `$data`.
- `1000` : Définit la taille maximale de la ligne (1000 caractères).
- `,` : Indique que les valeurs sont séparées par des virgules.
- La boucle while parcourt chaque ligne du fichier CSV jusqu'à la fin.
- Sécurisation des données

```
$id_salle = $conn->real_escape_string($data[0]);
$heure_debut = $conn->real_escape_string($data[1]);
$heure_fin = $conn->real_escape_string($data[2]);
$date = $conn->real_escape_string($data[3]);
```

- `Real_escape_string()` : Sécurise les données avant de les insérer dans la base de données (évite les injections SQL).
- `$data[0]` → ID de la salle
- `$data[1]` → Heure de début
- `$data[2]` → Heure de fin
- `$data[3]` → Date

```
if (!preg_match("/^\d{4}-\d{2}-\d{2}$/", $date)) {
    echo "Erreur : format de date incorrect ($date)<br>";
    continue;
}
```

- `Preg_match("/^\d{4}-\d{2}-\d{2}$/", $date)` : Vérifie si la date est au format YYYY-MM-DD.
- Si le format est incorrect, affiche une erreur et saute cette ligne (`continue`).

- Insertion des données dans la base

```
$sql = "INSERT INTO emploi_du_temps (id_salle, heure_debut, heure_fin, Date)
        VALUES ('$id_salle', '$heure_debut', '$heure_fin', '$date')";
```

- Crédation de la requête SQL pour insérer les données dans la table emploi_du_temps.
- Insère les valeurs \$id_salle, \$heure_debut, \$heure_fin et \$date.
- Exécution de la requête SQL

```
if ($conn->query($sql) === TRUE) {
    echo "✓ Données insérées pour $id_salle ($date)<br>";
} else {
    echo "✗ Erreur d'insertion : " . $conn->error . "<br>";
}
```

- If (\$conn->query(\$sql) ==TRUE) : Exécute la requête et vérifie si l'insertion a réussi.
- Si oui, affiche un message de confirmation ✓.
- Sinon, affiche un message d'erreur ✗ avec conn->error (pour identifier le problème).
- Fermeture du fichier et fin de l'importation

```
fclose($handle);
echo "<br>✗ Importation terminée !";
} else {
    echo "✗ Erreur lors de l'ouverture du fichier.";
```

- Fclose(\$handle); : Ferme le fichier CSV après la lecture.
- Affiche un message final pour signaler la fin du processus.
- Fermeture de la connexion à la base de données

```
$conn->close();
?>
```

- Ferme la connexion MySQL proprement.

	← T →		id_emploi_du_temps	id_salle	heure_debut	heure_fin	Date	
<input type="checkbox"/>	Éditer	Copier	Supprimer	1	SUD 07	08:10	12:35	2025-01-07
<input type="checkbox"/>	Éditer	Copier	Supprimer	2	SUD 07	08:10	10:55	2025-01-09
<input type="checkbox"/>	Éditer	Copier	Supprimer	3	SUD 07	13:25	15:05	2025-01-09
<input type="checkbox"/>	Éditer	Copier	Supprimer	4	SUD 07	08:10	09:00	2025-01-13
<input type="checkbox"/>	Éditer	Copier	Supprimer	5	SUD 07	12:35	15:55	2025-01-13
<input type="checkbox"/>	Éditer	Copier	Supprimer	6	SUD 07	08:10	11:45	2025-01-14
<input type="checkbox"/>	Éditer	Copier	Supprimer	7	SUD 07	08:10	10:55	2025-01-16
<input type="checkbox"/>	Éditer	Copier	Supprimer	8	SUD 07	13:25	15:05	2025-01-16
<input type="checkbox"/>	Éditer	Copier	Supprimer	9	SUD 07	12:35	16:55	2025-01-20
<input type="checkbox"/>	Éditer	Copier	Supprimer	10	SUD 07	08:10	12:35	2025-01-21
<input type="checkbox"/>	Éditer	Copier	Supprimer	11	SUD 07	08:10	10:55	2025-01-23
<input type="checkbox"/>	Éditer	Copier	Supprimer	12	SUD 07	13:25	15:05	2025-01-23
<input type="checkbox"/>	Éditer	Copier	Supprimer	13	SUD 07	08:10	09:00	2025-01-27
<input type="checkbox"/>	Éditer	Copier	Supprimer	14	SUD 07	12:35	16:55	2025-01-27
<input type="checkbox"/>	Éditer	Copier	Supprimer	15	SUD 07	08:10	12:35	2025-01-28
<input type="checkbox"/>	Éditer	Copier	Supprimer	16	SUD 07	08:10	10:55	2025-01-30
<input type="checkbox"/>	Éditer	Copier	Supprimer	17	SUD 07	13:25	15:05	2025-01-30
<input type="checkbox"/>	Éditer	Copier	Supprimer	18	SUD 07	12:35	16:55	2025-02-03
<input type="checkbox"/>	Éditer	Copier	Supprimer	19	SUD 07	08:10	12:35	2025-02-04
<input type="checkbox"/>	Éditer	Copier	Supprimer	20	SUD 07	08:10	09:00	2025-02-24
<input type="checkbox"/>	Éditer	Copier	Supprimer	21	SUD 07	12:35	16:55	2025-02-24
<input type="checkbox"/>	Éditer	Copier	Supprimer	22	SUD 07	08:10	12:35	2025-02-25
<input type="checkbox"/>	Éditer	Copier	Supprimer	23	SUD 07	08:10	10:55	2025-02-27
<input type="checkbox"/>	Éditer	Copier	Supprimer	24	SUD 07	13:25	15:05	2025-02-27
<input type="checkbox"/>	Éditer	Copier	Supprimer	25	SUD 07	12:35	16:55	2025-03-03

Résultat final du transfert de l'emploi de temps de ma Salle SUD07 sur la Base de données dans la table 'Emploi_du_temps'

Résumé

- ✓ Active l'affichage des erreurs pour le débogage.
- ✓ Se connecte à une base de données MySQL.
- ✓ Vérifie l'existence du fichier CSV avant de le traiter.
 - ✓ Lit et extrait les données du fichier CSV.
 - ✓ Vérifie et sécurise les valeurs avant insertion.
 - ✓ Vérifie le format de la date (YYYY-MM-DD).
 - ✓ Insère les données dans la base de données.
- ✓ Affiche des messages de confirmation ou d'erreur.
- ✓ Ferme le fichier CSV et la connexion MySQL après l'importation

Transfert automatique de l'emploi du temps EcoleDirect sur la Base de données

- [Installer Cron sur Linux \(Debian, Ubuntu\)](#) :

Sur Ubuntu ou Debian, installez Cron avec :

- `Sudo apt update && Sudo apt install cron -y`

Puis démarrez et activez le service :

- `Sudo systemctl start cron`
- `Sudo systemctl enable cron`

Vérifiez si il fonctionne bien :

- `Systemctl status cron`

Accéder au répertoire du serveur WEB:

- `Cd /var/www/PRO`

- [Ajouter une tâche Cron](#)

Ouvrez le fichier Crontab pour ajouter une tâche automatique :

- `Crontab -e`

Ajout de la ligne :

- `*/5 * /usr/bin/flock -n /tmp/PROJET.lock /usr/bin/php /var/www/PRO/PROJET.php >> /var/log/import_csv.log 2>&1 : Tous les jours`

Accéder au fichier PROJET.php :

- `Sudo nano PROJET.php`

```

?>php
$host = "192.168.1.100";
$user = "root";
$password = "123456789";
$database = "PROJET";

// Connexion à la base de données
$conn = new mysqli($host, $user, $password, $database);
if ($conn->connect_error) {
    die("Échec de connexion : " . $conn->connect_error);
}

// S'assurer que l'encodage est bien UTF-8
$conn->set_charset("utf8");

// Chemin du fichier CSV
$csvFile = "/var/www/PROJET/occupation_salles.csv";

// Vérifier si le fichier existe
if (!file_exists($csvFile)) {
    die("Erreur : Le fichier occupation_salles.csv n'existe pas.");
}

// Vider la table avant d'insérer de nouvelles données
$conn->query("TRUNCATE TABLE emploi_du_temps");

// Ouvrir le fichier CSV
if (($handle = fopen($csvFile, "r")) !== FALSE) {
    fgetcsv($handle); // Ignorer la première ligne (en-tête)

    while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
        if (isset($data[0], $data[1], $data[2], $data[3])) {
            $id_salle = $conn->real_escape_string($data[0]);
            $date = date('Y-m-d', strtotime(str_replace('/', '-', $data[1])));
            $heure_debut = $conn->real_escape_string($data[2]);
            $heure_fin = $conn->real_escape_string($data[3]);

            // Insérer les données en évitant les doublons
            $sql = "INSERT INTO emploi_du_temps (id_salle, heure_debut, heure_fin, Date)
                    VALUES ('$id_salle', '$heure_debut', '$heure_fin', '$date')";

            if (!$conn->query($sql)) {
                file_put_contents("/var/log/import_csv.log", "Erreur : " . $conn->error . " - Ligne : " . implode(",", $data) . "\n", FILE_APPEND);
            }
        } else {
            file_put_contents("/var/log/import_csv.log", "Erreur : Données incomplètes dans le fichier CSV\n", FILE_APPEND);
        }
    }
    fclose($handle);
    echo "Importation terminée avec succès !";
} else {
    die("Erreur d'ouverture du fichier.");
}

$conn->close();
?>

```

Programme du fichier PROJET.php

20°/ Protocol de communication :

- **Préquis : Ce qu'il faut installer**

-  **Sur le Raspberry Pi (Serveur)**

Installer Node.js (si non installé)

```
sudo apt update  
sudo apt install nodejs npm -y
```

Installer WebSocket pour Node.js :

```
npm install ws
```

Créer le serveur WebSocket (Sur le Raspberry Pi) :

Ce script écoute les connexions WebSocket et envoie un fichier à la demande :

```
const WebSocket = require('ws');
const fs = require('fs');
const os = require('os');

// Fonction pour récupérer l'adresse IP locale
function getLocalIp() {
    const interfaces = os.networkInterfaces();
    for (const iface of Object.values(interfaces)) {
        for (const config of iface) {
            if (config.family === 'IPv4' && !config.internal) {
                return config.address;
            }
        }
    }
    return 'localhost';
}

// Détection de l'IP actuelle
const localIp = getLocalIp();
const port = 8080;

const wss = new WebSocket.Server({ host: '0.0.0.0', port });

wss.on('connection', (ws) => {
    console.log("Client connecté");

    ws.on('message', (message) => {
        if (message.toString() === "SEND_FILE") {
            console.log("Demande reçue : envoi du fichier...");
            const filePath = '/home/pi/occupation_salles.csv';

            if (fs.existsSync(filePath)) {
                const fileData = fs.readFileSync(filePath);
                ws.send(fileData);
                console.log("Fichier envoyé !");
            } else {
                console.log("ERREUR : fichier introuvable !");
                ws.send("ERROR: File not found");
            }
        }
    });
});
```

Résumé du fonctionnement:

- Démarrage du serveur: Il écoute sur ws://IP_Raspberry:8080.
- Connection d'un client: Un message s'affiche lorsqu'un client se connecte.
- Demande de fichier :
 - Si le client envoie "SEND_FILE", le serveur vérifie l'existence du fichier /home/pi/occupation_salles.csv.
 - Si le fichier existe → Il est envoyé au client.
 - Sinon → Le serveur renvoie "ERROR File not found".
- Déconnexion du client: Un message s'affiche lorsque le client se déconnecte.

💡 Utilité : Permet d'envoyer un fichier via WebSocket d'un Raspberry Pi vers un client. 🚀

```
ws.on('close', () => {
    console.log("Client déconnecté");
});

console.log(`Serveur WebSocket en écoute sur ws://${localIp}:${port} et ws://localhost:${port}`);
```

Sur le Client (PC, autre Raspberry, etc.)

Installer Node.js (si non installé)

 Télécharger Node.js selon ton OS

Installer WebSocket pour Node.js

```
npm install ws
```

Après la mise en place du code dans le fichier du client, il n'a pas qu'à lancer le serveur avec :

```
node serveur.js
```

Explication du Protocole

 Flux de communication :

1. Le client se connecte au serveur WebSocket.
2. Le client envoie le message "SEND_FILE".
3. Le serveur vérifie l'existence du fichier :
 - o  Si oui → Il envoie le fichier en binaire.
 - o  Si non → Il envoie "ERROR File not found".
4. Le client reçoit le fichier et l'enregistre.

 Avantages de WebSocket pour le transfert de fichier

-  Connexion persistante : Pas besoin d'ouvrir et fermer une connexion à chaque requête.
-  Faible latence : Contrairement au HTTP, WebSocket permet un transfert direct sans surcharge de requêtes répétées.
-  Facilité d'implémentation : Code simple en Node.js.

CONCLUSION

- ❖ Le projet de radiateurs connectés a permis d'optimiser la consommation électrique et d'améliorer le confort thermique des occupants du bâtiment SUD CIEL du Lycée Saint-Eloi.
- ❖ **CE QUI MANQUE...**
 - ❖ Développer l'interface web
 - ❖ Développer le protocole de communication dans le sens sortant (paramétrage)
 - ❖ Mise en place de l'outil de gestion des sources (GIT pour logiciel web et logiciel embarqué)
 - ❖ Désigner les différents écrans de présentation des données (échelles de temps)