Projet IoT BUT 3
Parcours web

MétéoConnect
Station Météo Personnalisée

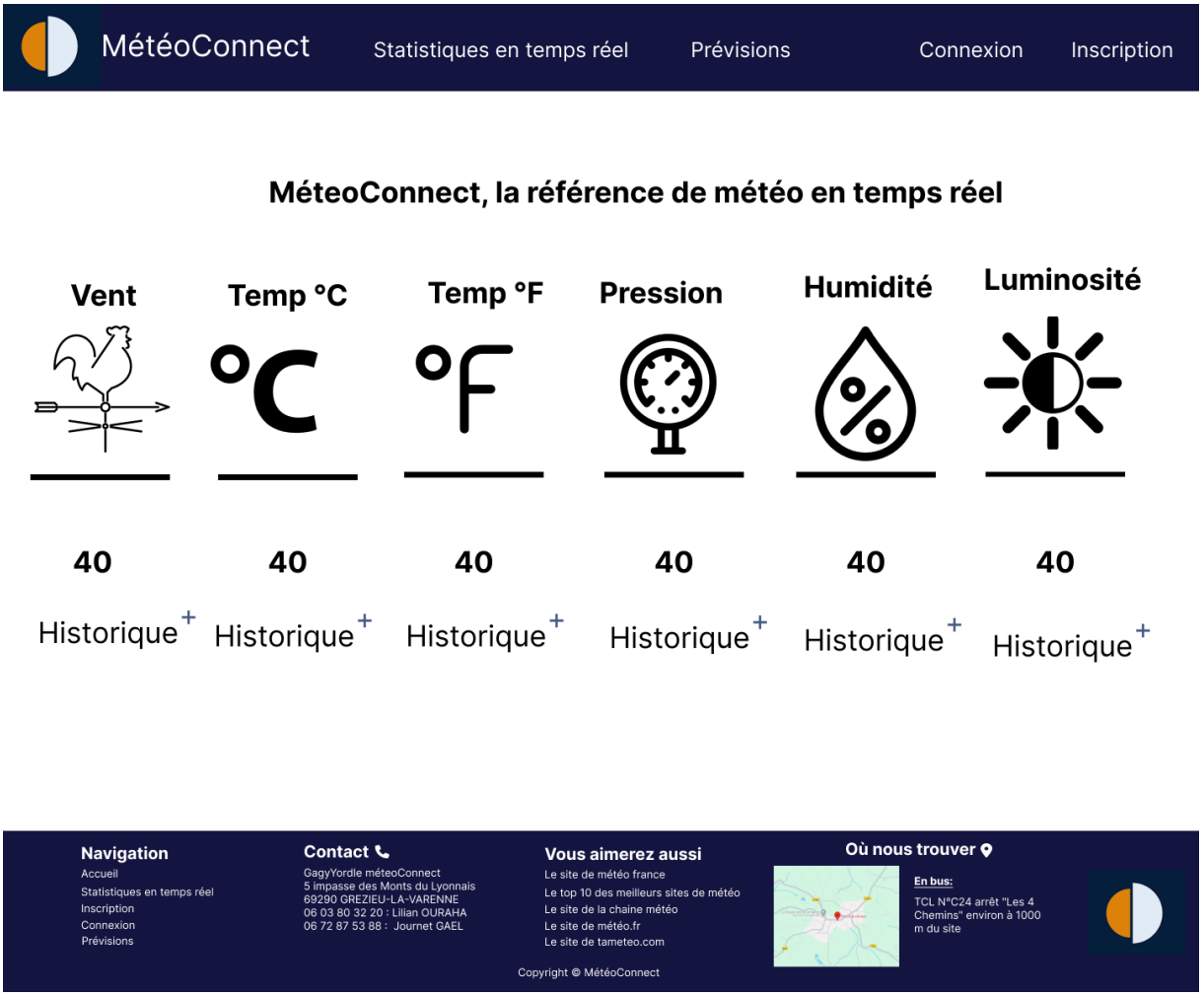
Ouraha Lilian

Projet

Ce projet combine des capteurs de température, d'humidité, de baromètre, de luminosité et de vent avec une interface web conviviale pour fournir des données météorologiques locales en temps réel, des graphiques de tendances et des conseils personnalisés. Les utilisateurs peuvent consulter la température en degrés Celsius ou Fahrenheit, le taux d'humidité, la pression atmosphérique, la luminosité et la direction du vent. Ils peuvent également définir des seuils d'alarme pour être avertis lorsque ces seuils sont atteints, par exemple en cas de température supérieure à 30 degrés Celsius. Le projet permet également à l'administrateur de fixer des seuils d'alarme, d'ajouter des informations sur les prévisions des prochains jours et d'être averti en cas de franchissement de ces seuils.

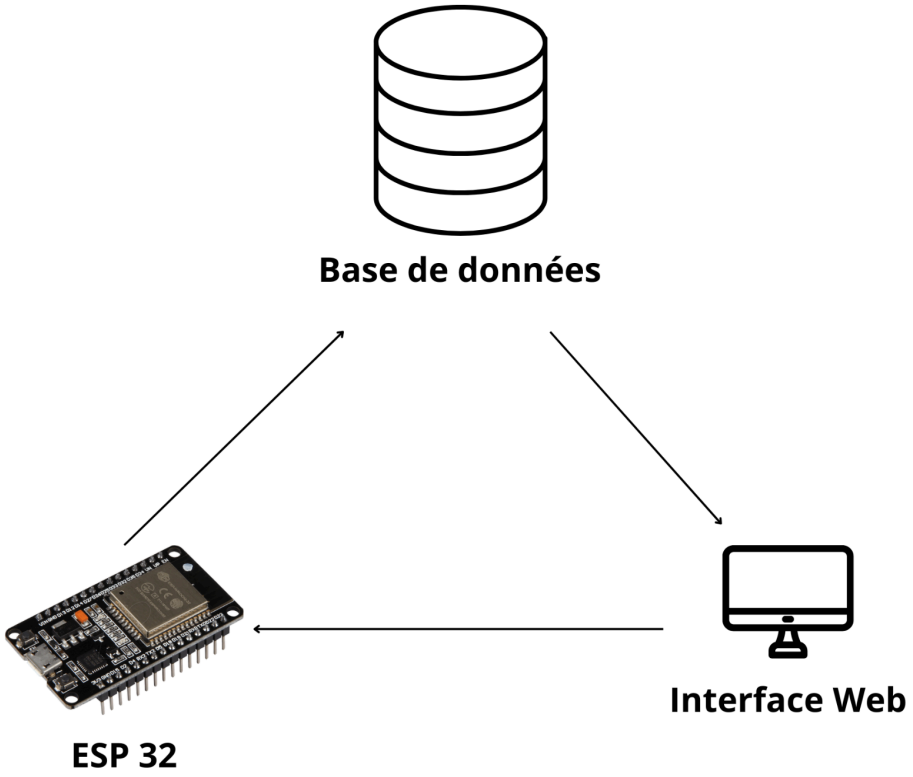
Maquette

.



Matériel & Technologies utilisées

- Un ESP 32
- Une girouette
- Un capteur de luminosité (Grove - Light sensor)
- Un Capteur de température, d'humidité et de pression (BME680)
- Un buzzer
- Une led RGB (Chainable RGB led)
- L'IDE Arduino
- L'IDE Visual Studio Code
- Base de données MySQL avec l'outil visuel MySQL Workbench
- Symfony



Description des routes

1. Éteindre l'Alarme
Méthode HTTP: GET
URL: /eteindreAlarme
Description: Envoie une requête à l'ESP32 pour éteindre l'alarme. Retourne une réponse indiquant si l'alarme a été éteinte avec succès (code 200) ou non (code 500).

2. Ajouter une Alarme
Méthode HTTP: POST
URL: /alarmes/ajouter
Description: Récupère les données reçues en POST pour créer une alarme pour l'utilisateur connecté et stocke cette alarme en base de données. Retourne un message de confirmation indiquant si l'ajout s'est bien passé ou non.

3. Récupérer la Dernière Mesure
Méthode HTTP: GET
URL: /mesures
Description: Permet de récupérer la dernière mesure ajoutée pour l'afficher dans une page.

4. Récupérer les Prévisions
Méthode HTTP: GET
URL: /mesures/previsions
Description: Récupère les prévisions pour le lendemain en s'appuyant sur les données de la veille. Les prévisions incluent les températures du matin, de l'après-midi et du soir.

5. Récupérer Toutes les Mesures
Méthode HTTP: GET
URL: /mesures/all
Description: Permet de récupérer l'ensemble des mesures pour en faire des statistiques.

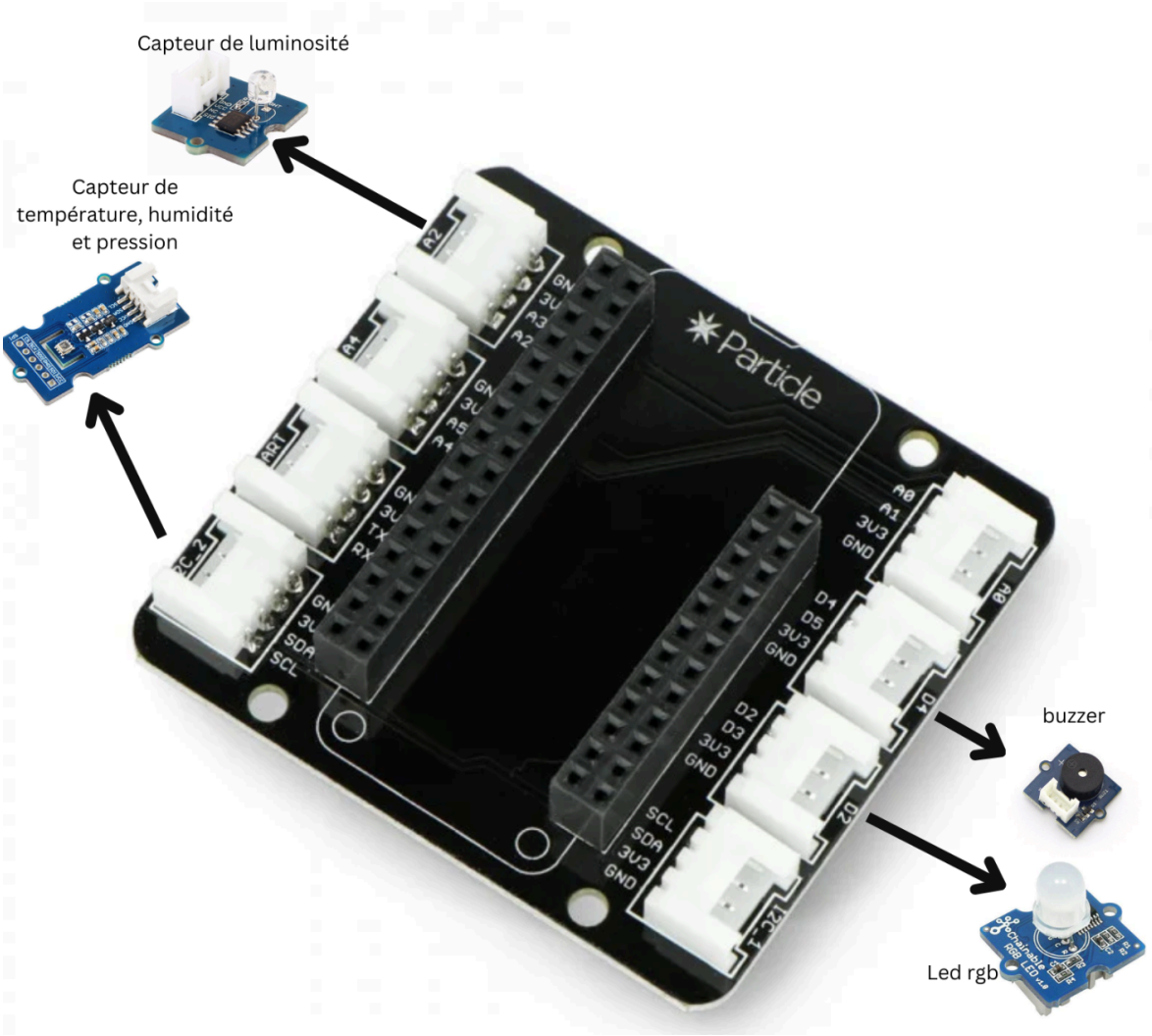
6. Récupérer l'Historique des Mesures par Date
Méthode HTTP: GET
URL: /mesures/historique/{date}
Description: Permet de récupérer les mesures d'une date donnée.

7. Ajouter une Mesure
Méthode HTTP: POST
URL: /mesures/ajout
Description: Permet d'ajouter une mesure avec les données fournies en POST (au format JSON). Selon les alarmes liées à l'utilisateur connecté, peut renvoyer une éventuelle requête /allumerAlarme à l'ESP32.

8. Authentification
Méthode HTTP: POST
URL: /login
Description: Route d'authentification pour les utilisateurs.

9. Création de Compte
Méthode HTTP: POST
URL: /register
Description: Route de création de compte pour les nouveaux utilisateurs.

Plan de montage du prototype



Lien du code source

gitlab:
https://iutbg-gitlab.iutbourg.univ-lyon1.fr/but3-wot-23-24/ouraha-journet/meteoconnect-rendus.git
github:
https://github.com/GagyZer/Meteo_connect

Communication entre les objets

librairies utilisées:

seeed_bme680.h: Utilisée pour interagir avec le capteur de qualité de l'air BME680 de Seeed Studio.

ChainableLED.h: Utilisée pour allumer et éteindre la led lors des alarmes

WiFi.h: Permet de se connecter au réseau Wi-Fi.

HTTPClient.h: Utilisée pour effectuer des requêtes HTTP, pour communiquer avec le serveur d'API
Dans mon cas, il sert à envoyer les données (température, luminosité…) au format POST.

ESPAsyncWebServer.h: Une bibliothèque pour la création de serveurs web asynchrones sur des plates-formes ESP32.

Langages & framework utilisés:

PHP: Le langage de programmation principale de notre application

JavaScript: Parfois utilisé pour des requêtes AJAX, lorsque l'on désactive l'alarme par exemple

-

Symfony: Framework qui nous a permis d'implémenter le modèle MVC dans notre application, également très utile pour la gestion des données

Bootstrap: Framework framework aidant au développement front-end de l'application

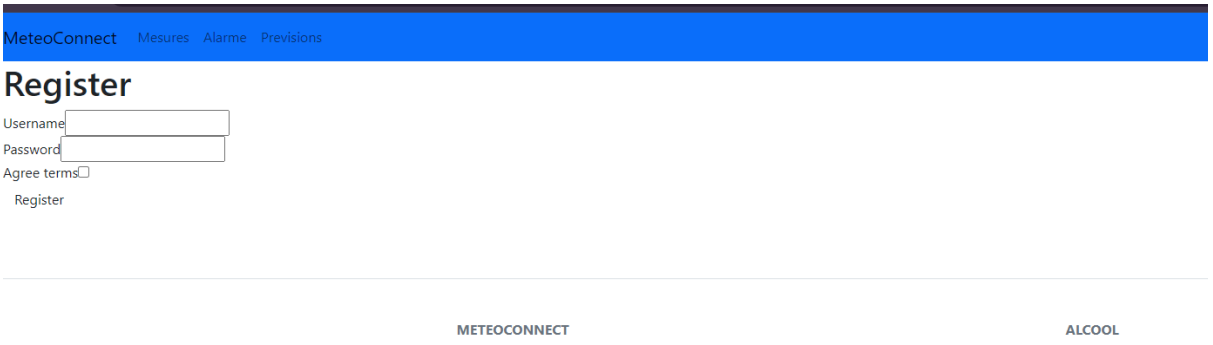
Fonctionnement de la communication :

Notre station météo communique donc avec l'objet de manière asynchrone, l'objet récupére à intervalle régulier (toute les 5 minutes en temps normal et toute les minutes lors des tests) les différentes valeurs à l'aide de ses capteurs (luminosité, pression, température et humidité) puis les renvoie au serveur web à l'adresse
`/mesures/ajout` au format JSON avec une méthode POST grâce à la librairie `HTTPClient`.

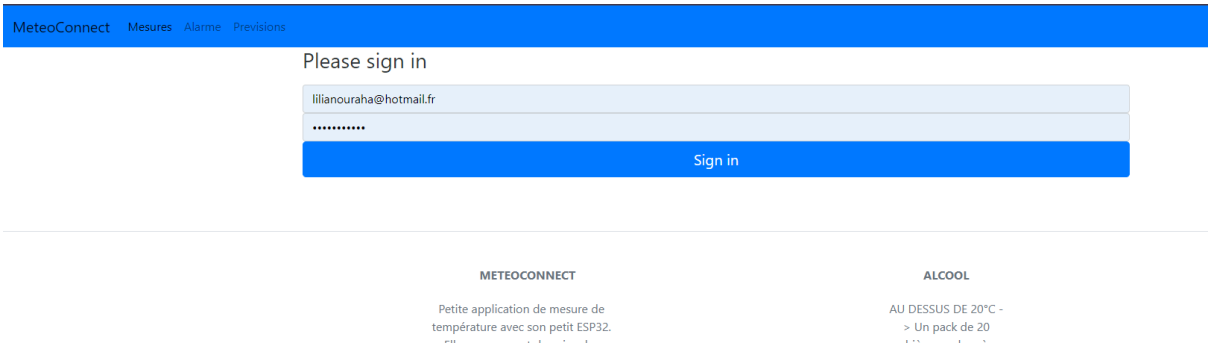
Le serveur web quant à lui ajoute ces données à la base de données puis il les analyses pour vérifier si une alarme est fixée pour une de ces valeurs **et** pour l'utilisateur connecté, par exemple si la valeur ajoutée à une température de 23°C et que l'utilisateur a une alarme pour une température supérieure à 22,5°C alors il enverra une requête get sur la route http://192.168.37.68/allumerAlarme que l'ESP32 écoute en permanence grâce à la librairie ESPAsyncWebServer.h (sur le port 80, celui de l'API) et va alors allumer l'alarme.

Pour éteindre l'alarme, l'utilisateur doit se rendre dans la page Alarme et cliquer sur le bouton "désactiver l'alarme" qui effectue une requête AJAX vers la route /eteindreAlarme qui fait une requête get sur la route http://192.168.37.68/eteindreAlarme que l'ESP32 écoute également en permanence et qui va alors éteindre l'alarme.

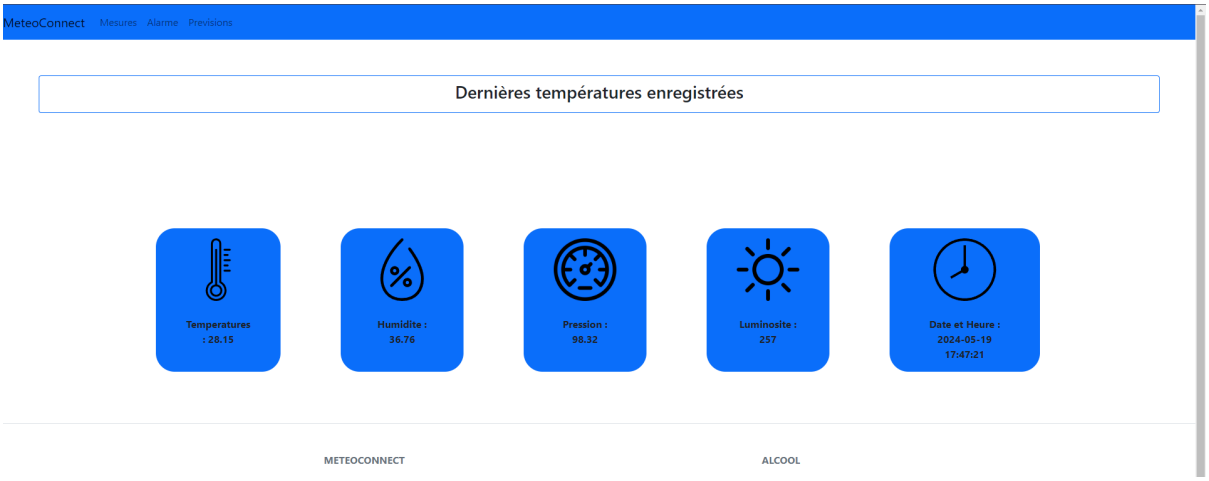
Application finale



Page d'inscription avec un formulaire classique(l'username doit être une adresse email)

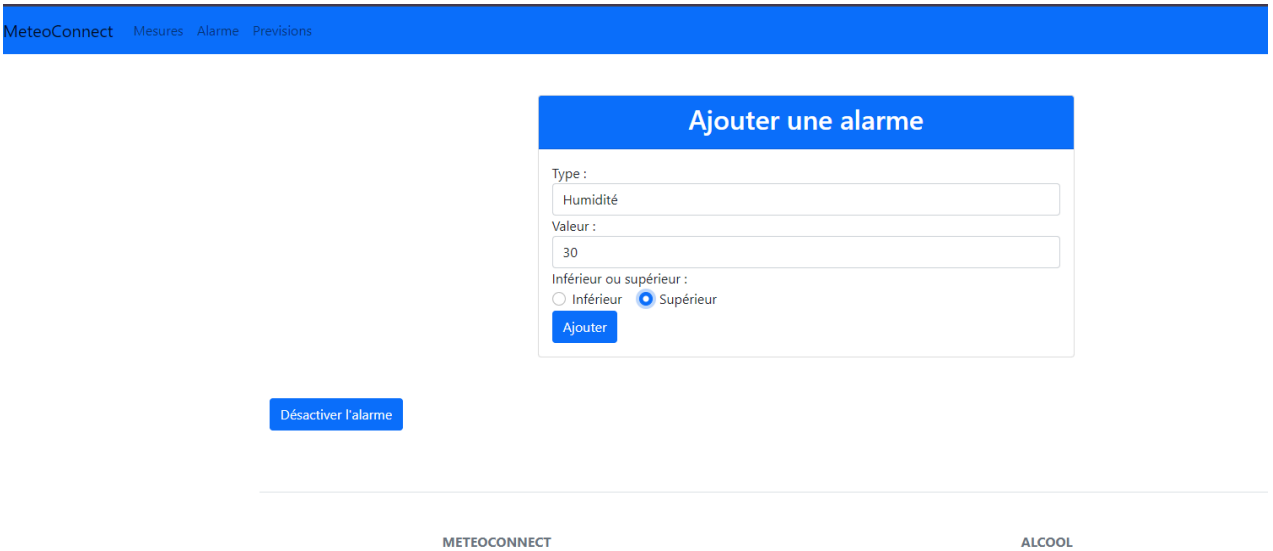


Page de connexion avec un formulaire classique également



Page principale avec la dernière mesure enregistrée (cette page est accessible même si l'utilisateur n'est pas connecté)

Page des prévisions du lendemain matin, après midi et soir, ces valeurs sont calculées en fonction des valeurs de la veille.



Page pour ajouter une alarme, le formulaire contient le champ type qui est un menu déroulant contenant: humidité, pression, température et luminosité, il faut en choisir un parmi

les 4 puis une valeur entière ou décimale et enfin choisir une option entre supérieure ou inférieure

Il y a également le bouton pour désactiver l'alarme en cas de besoins.

Cette page est uniquement accessible pour les utilisateurs connectés.