Documentation Technique Générale Projet V1.1

résumé

Ce document s'adresse aux membres du Groupe BLACMI , et autres Apprenants de SIMPLON ADENFIPA et des COACH et à toute personne qui voudrait prendre connaissance des caractéristiques techniques de cette documentation.
Le projet se présente comme suit:
Il s'agit de mettre en place un système qui permet de détecter la température ambiante et l'humidité ensuite afficher ses informations via une page web. Le système contiendra un microcontrôleur Arduino, un capteur de température et de l'humidité, un ventilateur qui tourne si la température est supérieure à 35°C sinon elle reste éteinte et un buzzer qui

sonne aussi si la température est supérieure à 35°C . Le ventilateur pourra être allumé manuellement par une télécommande.
La collecte de la température et de l'humidité se fera trois fois par jour à des heures précises:

- 8h 00
- 12h00
- 19h00

A chaque heure de la journée, les données collectées seront stockées dans une base de données.
Côté web, il faudra afficher les informations suivantes:

- La température et l'humidité pour chaque heure donnée;
- La température et l'humidité moyenne de la journée associées à une image qui dépend de la valeur de l'humidité. Par exemple, si l'humidité est supérieure à une valeur x afficher une image donnée.

La documentation va comprendre trois grand parties:

- La partie Frontend qui nous permet d'afficher graphiquement les informations recueillies au niveau de notre Arduino.
- La partie Serveur qui nous permet de stocker les informations recueillis aussi au niveau de notre arduino
- et enfin une Câblage bien finie qui représente les charges demandées par les projets accompagner de son programme.

Sommaire

I-Choix des technologies

a- Choix de l'éditeur de text(côté frontend et coté Arduino)

b-Outils de travails de collaboratifs et de Gestion de projets

c- Choix des langages

d-choix de la base de donnée

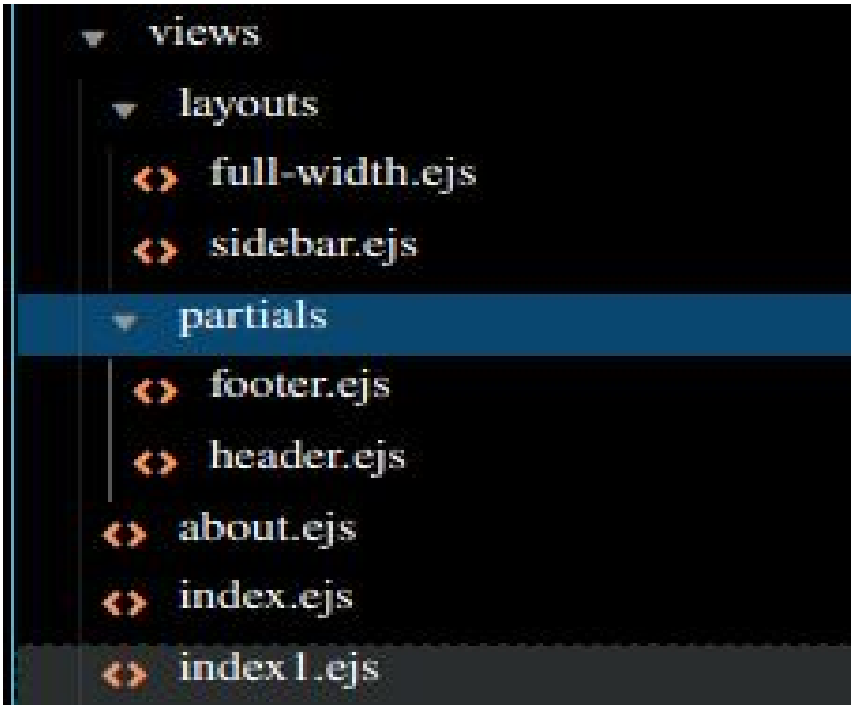
Fonctionnement du Projet

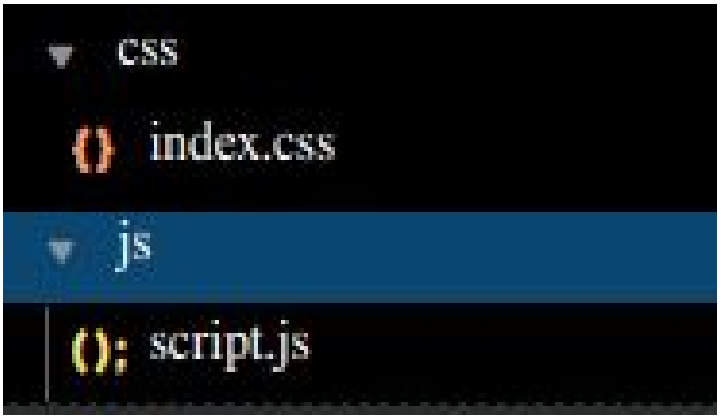
II Partie Backend

-index.js

III-Partie Frontend

Structure



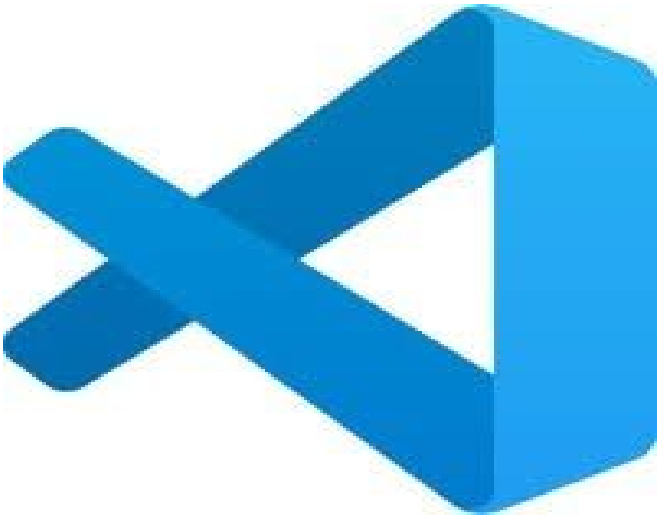


Partie Câblage

I-Choix des technologies

a- Choix de l'éditeur de text(côté frontend et coté Arduino)

Pour le choix de l'éditeur de texte nous avons choisis VISUAL STUDIO CODE qui est très facile à utiliser avec ses différentes extensions.



Pour l'éditeur arduino nous avons pris le soin de prendre la logicielle Standard d' Arduino pour le code.

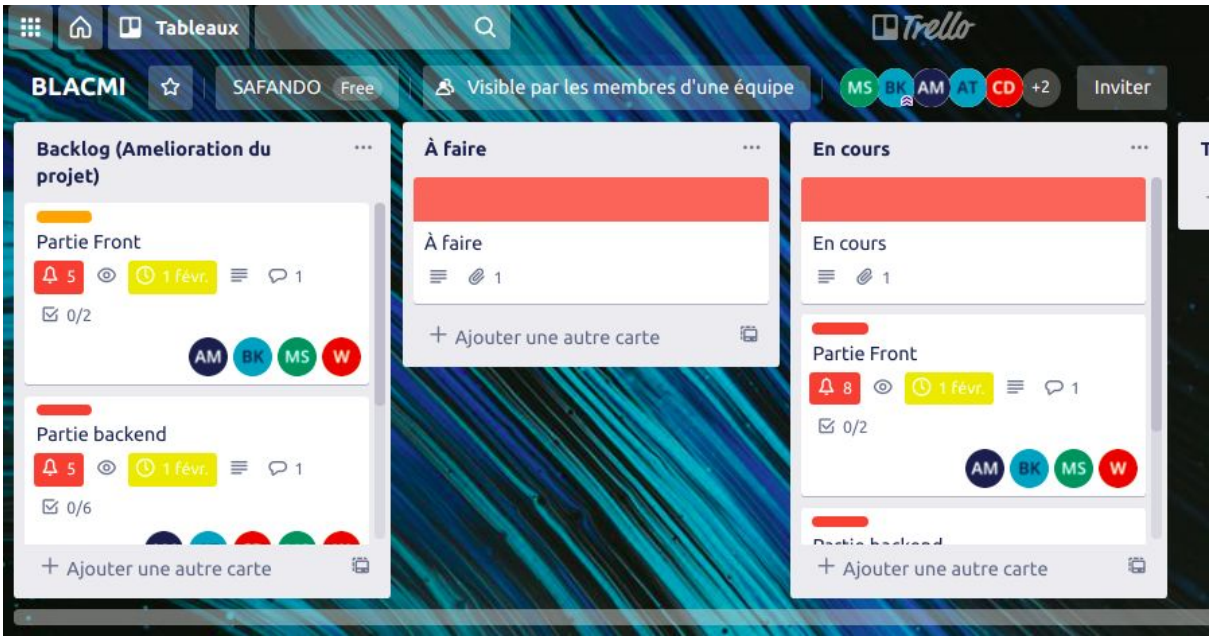
b-Outils de travails de collaboratifs et de Gestion de projets

-Pour les outils collaboratifs

Nous avons pris SLACK pour L'échanges des informations des documents et liens concernant le projet.Et d'autre part nous avons aussi créé un Groupe WhatsApp pour que l'information se transmette beaucoup plus rapidement.
Et enfin pour rendre fluide notre code nous l'avons déposé sur Github ou tous les membres de l'équipe seront capables de faire des modifications en l'améliorant.

-Pour les outils de Gestion de Projet

Nous avons pris Trello qui repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches.



Ainsi pour mieux gérer notre projet nous avons choisis de travailler avec la méthode agile Scrum qui nous permet de rendre à temps et très efficacement le projet à savoir d'être bien informé et de nous faciliter les réorientations opportunes.

Cette démarche participative active est un atout fondamental. Elle garantit pour le client le juste équilibre entre l'investissement prévu et le produit finalement livré. L'étude du prototype permet l'évaluation des fonctionnalités réalisées, et facilite la réflexion commune sur l'opportunité de futurs développements. D'autre part, l'étroite intimité entre les clients utilisateurs et les prestataires développeurs facilite l'appropriation future de l'outil.

Ainsi sa particularité est ce que l'on appelle Sprint C'est la phase essentielle de développement du produit. Limitée dans le temps, sans dépasser un mois pour autant, elle a pour but de réaliser "quelque chose" de présentable à un client. La méthode Scrum procède par itération. C'est en effet la bonne méthode pour, théoriquement, toujours rester en phase avec le besoin défini et les attentes du client.

c- Choix des langages

Pour l'interface Web nous allons utiliser Le Template EJS est un langage de création de modèles simple qui vous permet de générer un balisage HTML avec du JavaScript brut depuis Node. js.

Donc nous avec Ejs nous pouvons en déduire que nous avons simultanément utiliser toutes les langages nécessaire du coté front et backend reste à savoir , de définir certains termes déjà mentionnés précédemment.

Nous utilisons le terme Template c'est quoi Template:

Un **Template**, connu également sous le terme de modèle, *layout* ou Gabarit en français, représente l'ensemble des éléments graphiques de l'agencement des colonnes, passant par le choix des couleurs jusqu'à l'établissement de la structure des différents éléments enveloppant un site Internet, abstraction faite de son contenu.

Un Template possède l'avantage de disposer préalablement des éléments statiques d'une page web. Le Template ou gabarit contient notamment:

- Un pré-positionnement des zones constituant la page : les en têtes, l'identité visuelle, les colonnes et le footer formant ainsi un zonage du site.
- La localisation des menus de navigation.
- Les caractères, formes et couleurs du contenu textuel, titres et bien sûr des liens.
- Le contenu visuel tel que les images ornementales : les icônes, fonds…

Nous avons aussi parlé de node.js , bon Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.Node.js étant du JavaScript, on retrouve les bases de l'asynchrone et de l'événementiel. On dispose également du

gestionnaire de paquet officiel de Node.js : npm. Npm fonctionne dans un terminal et permet d'ajouter des dépendances à notre projet.
Et comme vous le voyez, nous utilisons principalement du JavaScript dans notre projet car ce dernier présente un certain avantage d'être manipulable facilement côté client et serveur sans se déplacer tout le temps de langes à autres.

d-choix de la base de donnée

Pour le choix de la base de données nous avons fait une comparaison entre les bases de données relationnelles et non relationnelles et nous avons jugé nécessaire de travailler avec les bases de données relationnelles notamment avec MONGODB étant beaucoup plus fluide avec node JS. Du coup nous vous présentons un tableau de comparaison entre les bases de données SQL et Non SQL.
Mais nous pouvons en ajouter que Les bases de données SQL sont des bases de données basées sur des tables tandis que les bases de données NoSQL sont des bases de données basées sur des paires clé-valeur, des bases de données graphiques, etc. Cela signifie que les bases de données SQL représentent des données sous la forme de tables composées de n nombre de lignes de données, tandis que les bases de données NoSQL sont la collection de paires clé-valeur, de documents, de bases de données graphiques, etc. qui ne possèdent pas de définitions de schéma standard. Pour plus de précisions nous avons mis ce lien à la disposition.

https://waytolearnx.com/2019/03/difference-entre-sql-et-nosql.html#:~:text=Diff%C3%A9rence%20cl%C3%A9%20entre%20SQL%2

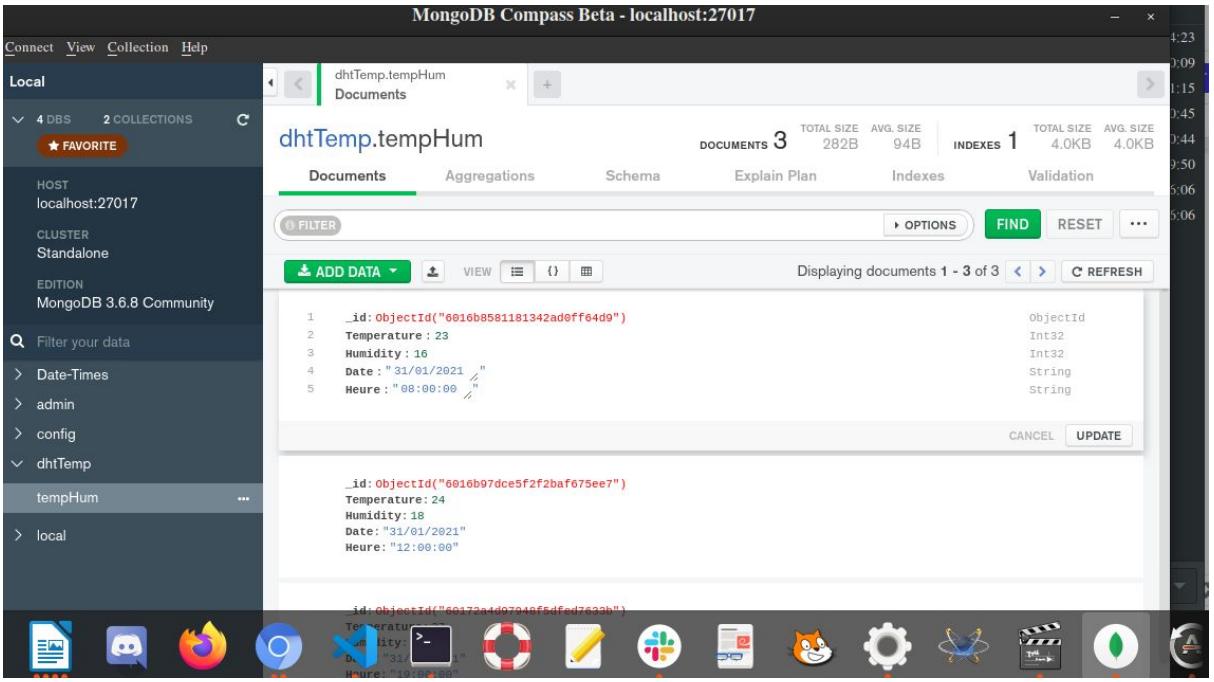
0et%20NoSQL&text=Cela%20signifie%20que%20les%20bases,bases%20de%20donn%C3%A9es%20graphiques%2C%20etc.

MySQL	MongoDB
INSERT	
INSERT INTO account (`A/c number`, `first name`, `last name`) VALUES ('12345746352', 'Mark', 'Jacobs');	db.account.insert({ A/c number: "12345746352", first name: "Mark", last name: "Jacobs" });
UPDATE	
UPDATE account SET contact number = 9426227364 WHERE A/c number = '12345746352'	db.account.update({ A/c number: '12345746352' }, { $set: {contact number: 9426227364 } });
DELETE	
DELETE FROM account WHERE e-mail address = 'jv1994@gmail.com';	db.account.remove({ "E-mail address": "jv1994@gmail.com" });

comme vous le voyez de cette image pour l'insertion de donnes dans notre collections nous avons téléchargé la version graphique de Mongodb
qui est Mongo Compass.
On télécharge le fichier Zip de Mongo compass version bêta avec ce lien
https://www.mongodb.com/try/download/compass?tck=docs_compass

-on se positionne dans le dossier de téléchargement
nous exécutons ce commande
sudo dpkg -i mongodb-compass-beta_1.25.0_beta.1_amd64.deb

après exécution et affectations de notre collections nous avons obtenu ceci



II Partie Backend

`-index.js`

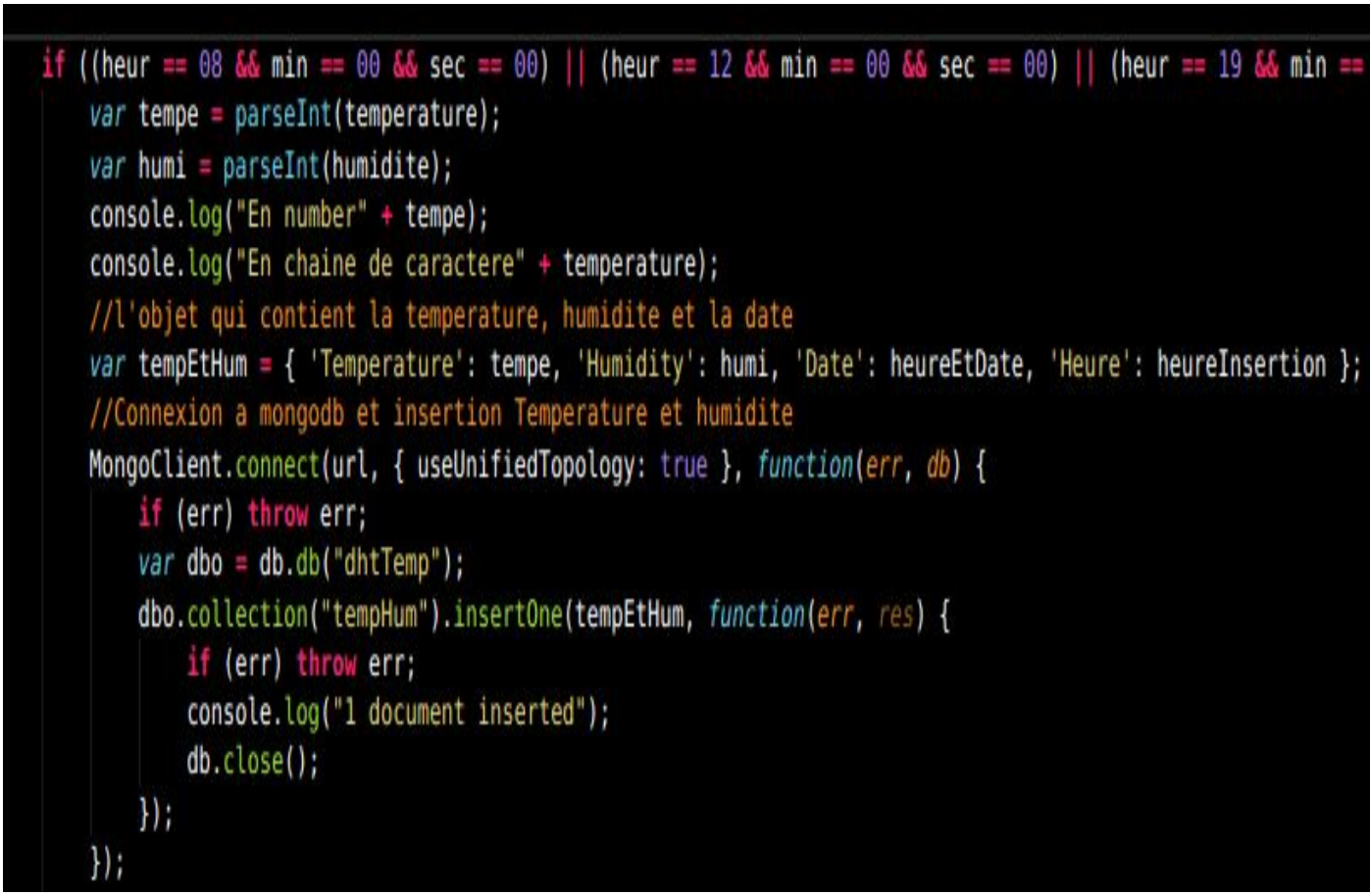
Ce fichier représente notre serveur c'est à l'intérieur que nous faisons appel à toutes les librairies nécessaires à savoir les librairies websocket le module expresse, les librairies socket.io et les modules pour les bases de données Mongodb pour le bon fonctionnement de notre projet.

C’est également dans ce fichier que nous interagissons avec la base de données Mongodb pour le stockage des informations telles que: la température, l'humidité et les images.

- Pour stocker la température et l’humidité venant de la carte arduino nous avons créé une fonction **`parser.on`** `dans laquelle on contrôle l’heur qu’il fait.`

`Pour cela on s’est limité sur les instructions du cahier de charge pour le contrôle de l’heur si l'heur du système correspond a l’heur demandé on enregistre la température et l'humidité qu’il fait.`

-



```
if ((heur == 08 && min == 00 && sec == 00) || (heur == 12 && min == 00 && sec == 00) || (heur == 19 && min ==
    var tempe = parseInt(temperature);
    var humi = parseInt(humidite);
    console.log("En number" + tempe);
    console.log("En chaine de caractere" + temperature);
    //l'objet qui contient la temperature, humidite et la date
    var tempEtHum = { 'Temperature': tempe, 'Humidity': humi, 'Date': heureEtDate, 'Heure': heureInsertion };
    //Connexion a mongodb et insertion Temperature et humidite
    MongoClient.connect(url, { useUnifiedTopology: true }, function(err, db) {
        if (err) throw err;
        var dbo = db.db("dhtTemp");
        dbo.collection("tempHum").insertOne(tempEtHum, function(err, res) {
            if (err) throw err;
            console.log("1 document inserted");
            db.close();
        });
    });
```

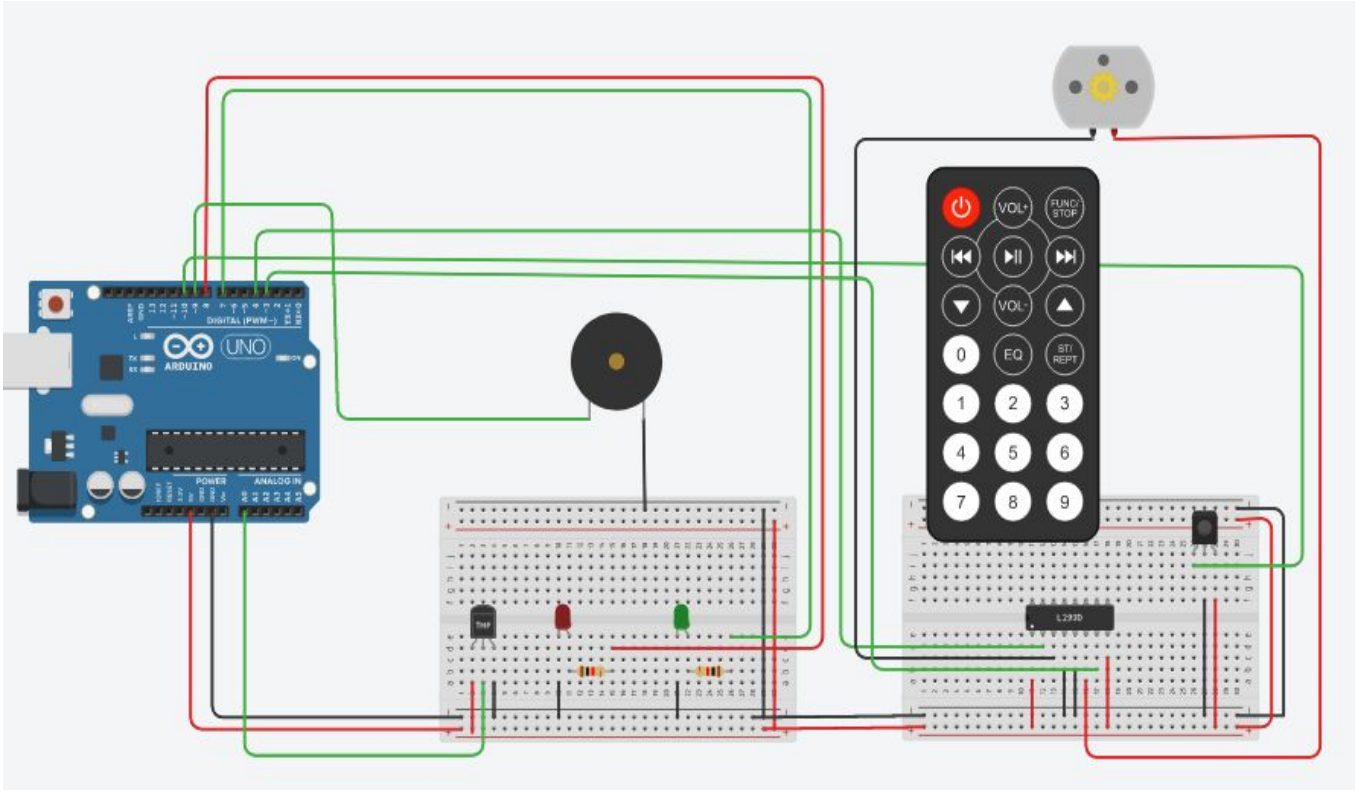
-Dans `app.get('', (req, res) => {`

C’est ici que nous avons définies les requêtes qui nous permettent d’attaquer la base de données pour récupérer la température, l’humidité et leur moyenne.
Dans cette même partie on n’a creer un tableau d’objet appelé objet ou nous avons stocké toutes les données retournées dans la base de donnée, puis cet objet lui même retourné vers le vue principale a savoir index.ejs.
Dans ce fichier index.js nous avons également implémenté des fonctions qui nous ont servie pour le stockage des images dans la base de donnée,tout en communiquant avec la vue, index 1.ejs

III-Partie Frontend

-index.js : Ce fichier représente notre Vue principale, c'est ici que nous affichons les données venant de la base de données. Pour cela nous avons utilisé la méthode -forEach() qui nous a servi à parcourir l’objet qui a été propulsé par index.js dans ce fichier nous avons créé un tableau pour afficher ces données.
-full-width.ejs : C’est le fichier ou nous incluons le Header, le footer et le body de toutes les vues.
-footer.ejs : Il contient toutes les informations du pied de page.
-header.ejs : Il contient les images qui varient en fonction de l'humidité , c'est également sur ce fichier qu’on va afficher l’humidité et la température en temps réel.
-index1.ejs : Il contient le formulaire pour l’enregistrement des images.

Partie Câblage



- La Led rouge est branchée sur le pin 8 de la carte Arduino.
- La led verte est branchée sur le pin 7 de la carte arduino.
- Le capteur de température est branché sur le pin 2 de la carte Arduino.
- Le capteur infrarouge est branché sur le pin 10 de la carte Arduino.
- Le ventilateur est branché sur le circuit intégré entre le pin 3 et le pin 6 et communique avec la carte arduino entre le pin 3 et 4.
- Le ventilateur est commandé par la télécommande infrarouge.