



République Du Sénégal
Un Peuple- Un But- Une Foi
L'Excellence est ma constance, l'éthique ma vertu.

P3 ADEFNIPA SIMPLON SENEGAL
GROUPE 5

PROJET : YAKAR

THÈME : Système de Surveillance et de Contrôle de la
Température et de l'Humidité en Temps Réel pour un Magasin de
Stockage de produit.

Réalisé par :
Moustapha DIENG
Mamadou lamine NIASSY
Ahmadou Bamba
Abou Abdou

Année 2023-2024

RÉSUMÉ

La gestion optimale de la température et de l'humidité est cruciale pour assurer la qualité des produits stockés dans les entreprises de transformation locale. Dans ce cadre, la structure "YAKAR" souhaite mettre en place un système innovant permettant de contrôler et de surveiller ces paramètres environnementaux dans son magasin de stockage. Le projet consiste à concevoir une solution web qui permettra de suivre en temps réel la température et l'humidité, d'effectuer des actions automatiques (activation du ventilateur, alerte sonore, etc.) en fonction des seuils définis, et d'offrir des fonctionnalités de contrôle à distance via une interface intuitive.

Le système devra collecter les données de température et d'humidité à des moments précis de la journée (10h00, 14h00, 17h00), stocker ces informations dans une base de données, et les afficher en temps réel sur un écran LCD ainsi que sur une interface web accessible par les utilisateurs. De plus, des actions automatiques seront mises en place : si la température dépasse 27°C, le système devra activer un ventilateur, émettre une alerte sonore et afficher un signal rouge, tandis que si la température est inférieure ou égale à 27°C, un signal vert sera affiché et le ventilateur restera éteint.

Le projet comprend également des fonctionnalités de gestion des utilisateurs pour l'administrateur, telles que la création, modification et suppression de comptes, ainsi que la gestion des rôles. Les administrateurs auront également accès à un tableau de bord pour contrôler manuellement certains paramètres du système, comme l'activation du ventilateur.

L'objectif principal est de mettre en place une solution complète et fiable permettant de contrôler à distance les conditions de stockage, tout en assurant une expérience utilisateur fluide et intuitive.

Dans ce cadre, nous allons aborder la conception technique du système, du câblage des composants électroniques à la création de l'application web, en passant par le développement de la base de données. Ce projet sera réalisé selon une méthodologie agile et collaborative, avec des livrables spécifiques à chaque phase de développement, dont la maquette de l'application, le code source, ainsi que la documentation complète du projet.

Ce projet représente une opportunité d'acquérir des compétences techniques variées, allant du câblage électronique à la création d'une application web complète, en passant par la gestion de bases de données et le travail en équipe dans un environnement collaboratif.

SOMMAIRE

I. CHOIX DES TECHNOLOGIES.....	3
II. UTILES DE DÉVELOPPEMENT.....	5
1. Visual Studio Code (VS Code).....	5
4. Angular (Front-End).....	7
III. PRODUCTION DE L'APPLICATION.....	9
1. Arduino Uno (avec Keypad et Relais pour le ventilateur).....	11
Composants connectés à l'Arduino Uno :	11
Explication :	11
2. Arduino Mega (avec DHT11 et écran LCD).....	11
Composants connectés à l'Arduino Mega :	11
Explication :	11
Communication entre les deux cartes (Uno et Mega).....	12
Communication Série (UART).....	12
2. Schéma Récapitulatif.....	12
IV. FONCTIONNEMENT DE L'APPLICATION.....	13
a. Authentification par Email et Mot de Passe.....	13
b. Authentification par Code Secret.....	14
Cas d'utilisation typique :	14
a. Surveillance en Temps Réel.....	15
b. Consultation des Données Historiques.....	15
c. Calcul des Moyennes.....	16
d. Gestion des Utilisateurs.....	16
a. Surveillance en Temps Réel.....	17
b. Consultation des Données Historiques.....	17
c. Affichage des Informations en Temps Réel.....	17
V. CONCLUSION.....	18

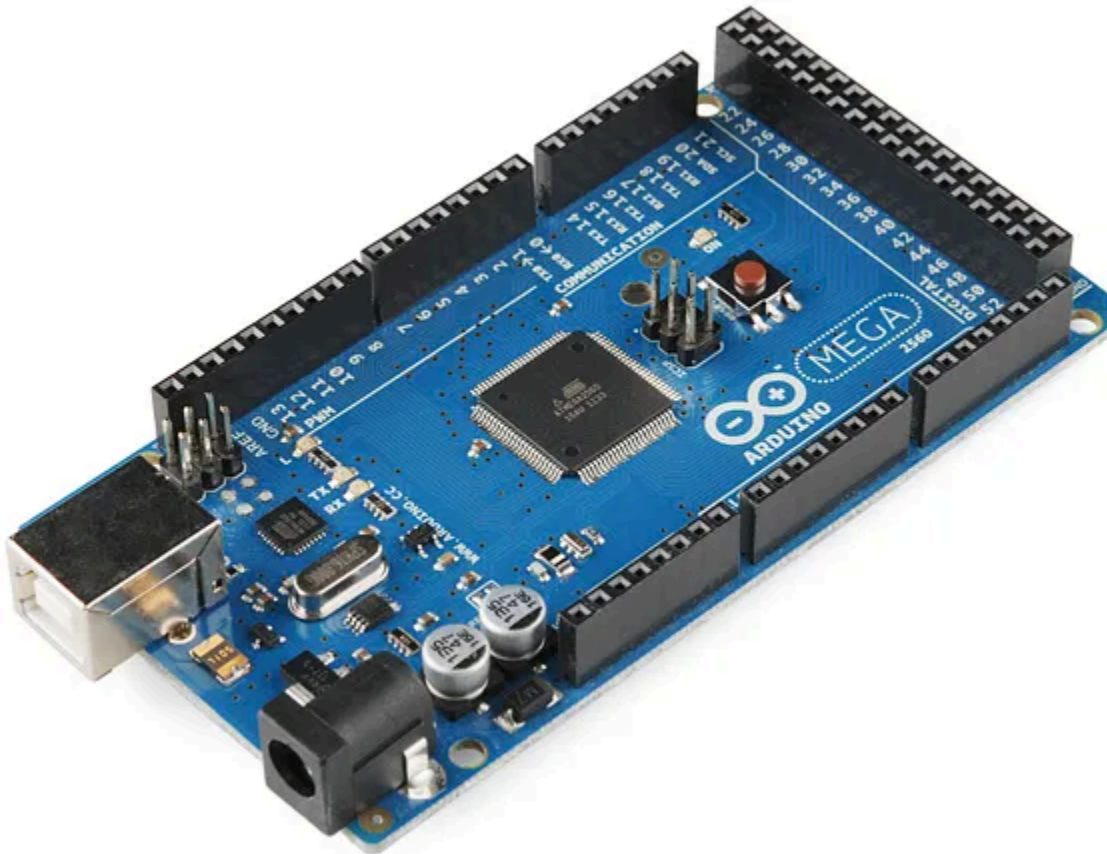
I. CHOIX DES TECHNOLOGIES

Dans le cadre du projet de gestion de la température et de l'humidité pour la structure "YAKAR", plusieurs technologies ont été sélectionnées pour assurer l'efficacité, la fiabilité et la scalabilité de la solution. Ces choix se basent sur les besoins spécifiques du projet, qui incluent la collecte de données en temps réel, l'automatisation des actions, le contrôle à distance, ainsi que la gestion d'une interface web conviviale pour l'utilisateur. Voici un détail des technologies choisies pour chaque composant du système :

1. Composants électroniques et microcontrôleur

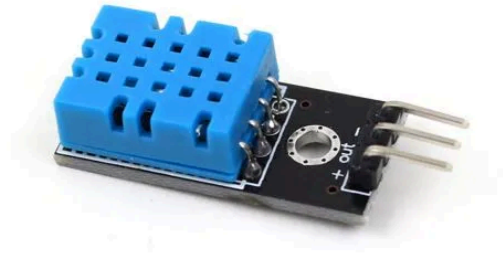
a. Arduino Mega

Une option plus simple et plus économique. Utilisé principalement pour la collecte de données des capteurs, mais nécessitant l'ajout d'un autre appareil pour gérer la communication web. Il est parfait si la priorité est la simplicité et la faible consommation énergétique.



b. Capteurs de Température et d'Humidité

DHT11 : Moins précis et avec une gamme de mesure plus restreinte, il pourrait être une option moins coûteuse, mais le DHT22 est souvent préféré pour des projets nécessitant des mesures plus fiables.



c. Actionneurs et dispositifs

- ❖ **Ventilateur** : Un ventilateur 12V peut être contrôlé via un relais sur Arduino pour activer/désactiver le ventilateur en fonction de la température.
- ❖ **Buzzer** : Utilisé pour alerter l'utilisateur lorsque la température dépasse un seuil critique. Il peut être directement contrôlé via un port.
- ❖ **LEDs (Signal rouge et vert)** : Des LEDs seront utilisées pour afficher l'état du système (température au-dessus ou en dessous de 27°C), et pourront être pilotées via l'Arduino.
- ❖ Un **keypad 4x4** est une matrice de 16 touches qui permet de saisir des commandes sous forme de chiffres ou de lettres. Il est couramment utilisé dans les projets électroniques nécessitant des saisies de données simples comme des codes PIN, des sélections de menus, etc.
- ❖ L'intégration d'un **écran LCD I2C** dans notre projet permet d'afficher en temps réel des informations relatives à la température et à l'humidité dans le magasin de stockage, directement sur le dispositif sans avoir besoin d'interface web. C'est particulièrement utile pour un affichage local rapide et pour le suivi en temps réel des conditions environnementales.
- ❖ Une **breadboard** (ou planche à pain) est un outil essentiel pour le prototypage rapide et l'assemblage de circuits électroniques sans avoir besoin de souder les composants. Elle permet de tester facilement différentes configurations de câblage et de composants avant de passer à une version plus permanente. Dans le cadre de votre projet, la breadboard sera utilisée pour connecter les composants électroniques tels que le **capteur de température et d'humidité**, l'**écran LCD I2C**, le **keypad**, le **ventilateur**, et d'autres composants nécessaires.
- ❖ Un **relai** est essentiel lorsque vous souhaitez contrôler des appareils qui nécessitent plus de courant ou une tension plus élevée que ce que votre microcontrôleur (comme un **Raspberry Pi** ou un **Arduino**) peut gérer directement. Par exemple, un ventilateur

fonctionne généralement avec une tension de 12V ou 220V, alors que les microcontrôleurs utilisent des signaux à faible tension (3.3V ou 5V). Le **relai** sert d'interrupteur pour contrôler ces appareils externes en toute sécurité en utilisant un signal de commande faible.

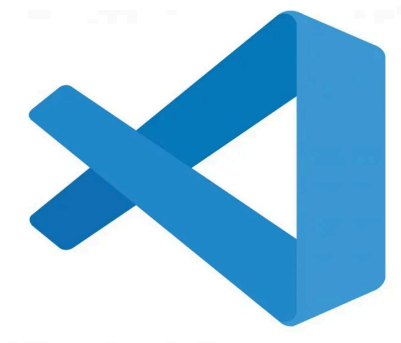


II. UTILES DE DÉVELOPPEMENT

Dans le cadre de votre projet, vous avez choisi d'utiliser certains outils de développement spécifiques qui facilitent la gestion du code, la base de données et l'interface utilisateur. Voici un aperçu des outils que vous avez utilisés, leur rôle et pourquoi ils sont adaptés à votre projet.

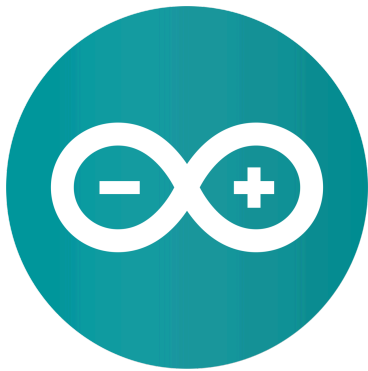
1. Visual Studio Code (VS Code)

Visual Studio Code (ou **VS Code**) est un éditeur de code source léger mais puissant développé par Microsoft. Il est particulièrement adapté au développement d'applications web et est largement utilisé dans les projets JavaScript, Node.js, et pour le développement backend en général.



2. Arduino

Dans le cadre de ce projet, l'utilisation de **l'Arduino** est essentielle pour interfacer les capteurs (comme ceux de température et d'humidité), le contrôle du ventilateur (via relais), et la gestion des signaux d'alerte (buzzer, LED) avec le système global. Arduino est un microcontrôleur open-source, largement utilisé dans les projets électroniques pour son accessibilité et sa capacité à interagir avec divers capteurs et actionneurs.



3. MongoDB et MongoDB Compass

MongoDB est une base de données NoSQL largement utilisée dans le développement d'applications modernes, en particulier pour des applications nécessitant de gérer des données non structurées ou semi-structurées. MongoDB est basé sur des documents JSON-like, ce qui permet une grande flexibilité.

MongoDB Compass est l'interface graphique officielle de MongoDB, permettant aux développeurs de visualiser, interroger et gérer facilement leurs bases de données MongoDB.



4. Angular (Front-End)

Angular est un framework JavaScript développé par Google pour la création d'applications web dynamiques et riches. Il est particulièrement adapté aux applications à page unique (SPA - Single Page Applications), où les interactions avec l'utilisateur se produisent sans rechargement complet de la page.



5. Node.js (Back-End)

Node.js est un environnement d'exécution JavaScript côté serveur, basé sur le moteur V8 de Google Chrome. Il est particulièrement adapté pour les applications web en temps réel, comme dans votre cas où les données de température et d'humidité sont collectées et affichées régulièrement.



6. Bootstrap

Bootstrap est un framework CSS populaire et open-source, développé par Twitter, qui facilite la création de sites web réactifs et modernes. Il fournit une série de composants et de styles préconçus, ce qui permet de gagner du temps dans la conception de l'interface utilisateur et d'assurer une cohérence visuelle. L'utilisation de Bootstrap dans votre projet permet d'améliorer l'interface en la rendant esthétique, fonctionnelle et adaptée à tous les types d'appareils.



7. WebSocket et Socket.IO

L'utilisation de **WebSocket** et de **Socket.IO** dans votre projet est essentielle pour gérer les échanges de données en temps réel entre le front-end (Angular) et le back-end (Node.js). En

particulier, pour un projet qui gère des données telles que la température et l'humidité, l'intégration d'une communication bidirectionnelle en temps réel est indispensable pour garantir que les utilisateurs reçoivent immédiatement les mises à jour lorsque la température change ou lorsque d'autres événements se produisent dans le système.

III. PRODUCTION DE L'APPLICATION

1. Backend(Création d'une API avec Node.js)

Dans le cadre de ce projet, la création d'une **API RESTful** avec Node.js permet de gérer toutes les opérations backend liées à la gestion des utilisateurs, des données de température et d'humidité, ainsi que l'interaction avec le système en temps réel. Vous avez développé votre propre API pour gérer les connexions, les opérations **CRUD** (Create, Read, Update, Delete) pour les données des capteurs, et gérer les déconnexions des utilisateurs.

Voici une vue d'ensemble de la façon dont le back-end est structuré et comment les différentes fonctionnalités sont implémentées dans l'API avec Node.js et Express.

a. structure de l'API

L'API backend de votre application est construite à l'aide de Node.js et de Express.js, un framework léger et flexible qui facilite la gestion des requêtes HTTP. L'API se divise en plusieurs parties principales, que nous allons détailler ci-dessous.

- Gestion des Utilisateurs : Création de comptes, authentification (connexion et déconnexion), gestion des rôles et des accès.
- Gestion des Données de Température et d'Humidité : Création, récupération, mise à jour et suppression des données collectées à partir des capteurs (CRUD).
- Authentification et Sécurisation : Mise en place de l'authentification via JWT (JSON Web Tokens) pour protéger certaines routes et garantir que seul un utilisateur authentifié puisse accéder aux fonctionnalités critiques.
- WebSocket pour la communication en temps réel : Intégration de Socket.IO pour envoyer les mises à jour des données de température et d'humidité en temps réel.

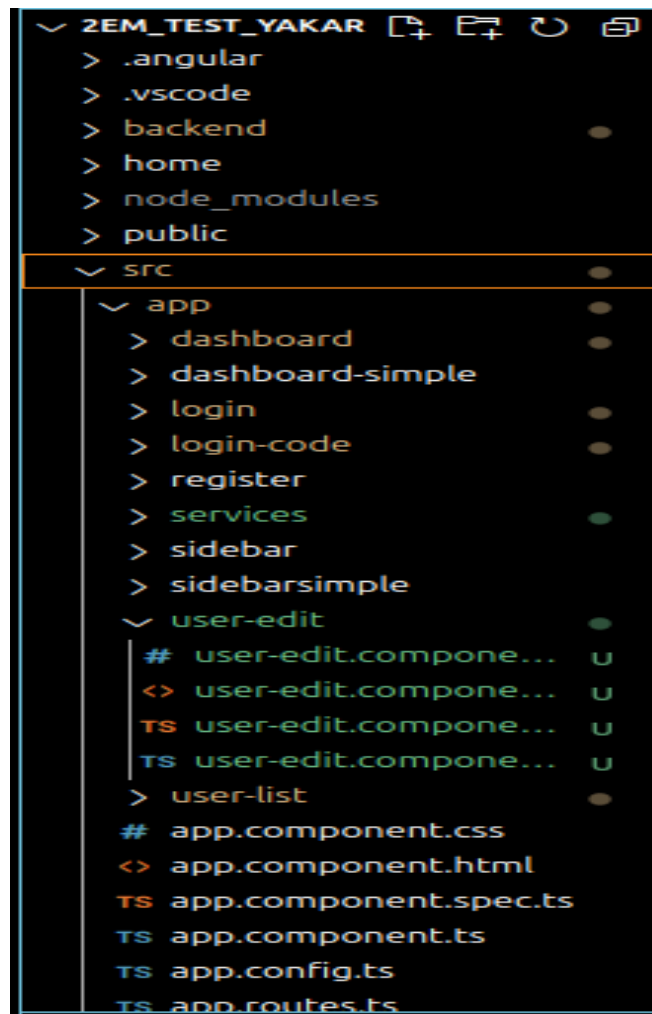
Voici la documentation de notre API:
<https://documenter.getpostman.com/view/39624135/2sAYBREDGU>

2. Frontend

Le frontend du projet utilise Angular, un framework puissant pour le développement d'applications web dynamiques et réactives. Angular facilite la création de composants modulaires, l'intégration avec des API RESTful, et la gestion des données en temps réel via WebSockets, ce qui est essentiel pour le suivi en temps réel de la température et de l'humidité dans votre application.

Voici une vue d'ensemble de l'architecture et des principales fonctionnalités mises en œuvre dans l'application frontend avec Angular.

a. Structure



3. Câblage

Dans ce projet, le câblage des composants électroniques a été un aspect crucial pour garantir le bon fonctionnement du système de gestion de la température et de l'humidité du magasin. Le système utilise plusieurs composants tels qu'un microcontrôleur Arduino, des capteurs de température et d'humidité, un écran LCD I2C, un keypad, un buzzer, un ventilateur avec relais, et d'autres composants pour le contrôle et l'affichage des données.

Le câblage total est effectué à l'aide de 27 fils pour relier tous les composants au microcontrôleur (Arduino), permettant de collecter, afficher et contrôler les données en temps réel.

1. Arduino Uno (avec Keypad et Relais pour le ventilateur)

L'Arduino Uno est utilisé pour contrôler le **keypad** et le **relai pour le ventilateur**, qui nécessitent plusieurs entrées et sorties numériques.

Composants connectés à l'Arduino Uno :

- **Keypad (Clavier matriciel 4x4)** : Utilisé pour la saisie des informations (ex : code secret ou réglage).
- **Relais pour le ventilateur** : Utilisé pour contrôler l'alimentation du ventilateur.

Explication :

- **Keypad** : Les pins du clavier matriciel (8 fils) sont reliés aux pins **PIN 2 à PIN 9** de l'Arduino Uno. C'est un câblage standard pour un keypad 4x4.
- **Relais** : Le relais est connecté à la broche **D11** de l'Arduino Uno pour le contrôle du ventilateur. Assurez-vous de connecter correctement les bornes de puissance du relais aux broches **VCC**.

2. Arduino Mega (avec DHT11 et écran LCD)

L'Arduino Mega sera utilisé pour les composants **DHT11** (capteur de température et d'humidité) et l'**écran LCD I2C**, qui nécessitent moins de pins numériques, mais bénéficient d'un plus grand nombre de broches disponibles sur l'Arduino Mega.

Composants connectés à l'Arduino Mega :

- **DHT11 (Température et Humidité)** : Ce capteur envoie des données à l'Arduino.

Explication :

- **DHT11** : Le capteur de température et d'humidité est connecté avec **VCC** à **5V**, **GND** à **GND**, et **Signal (Data)** à la broche **PIN 2** de l'Arduino Mega.

Communication entre les deux cartes (Uno et Mega)

Comme nous utilisons **deux cartes Arduino**, nous devons établir une communication entre elles pour échanger des données telles que la température et l'humidité mesurées par le DHT11 (sur l'Arduino Mega) tout en contrôlant le ventilateur via le relais sur l'Arduino Uno.

Pour cela, nous pouvons utiliser la **communication série** (UART) pour connecter les deux cartes.

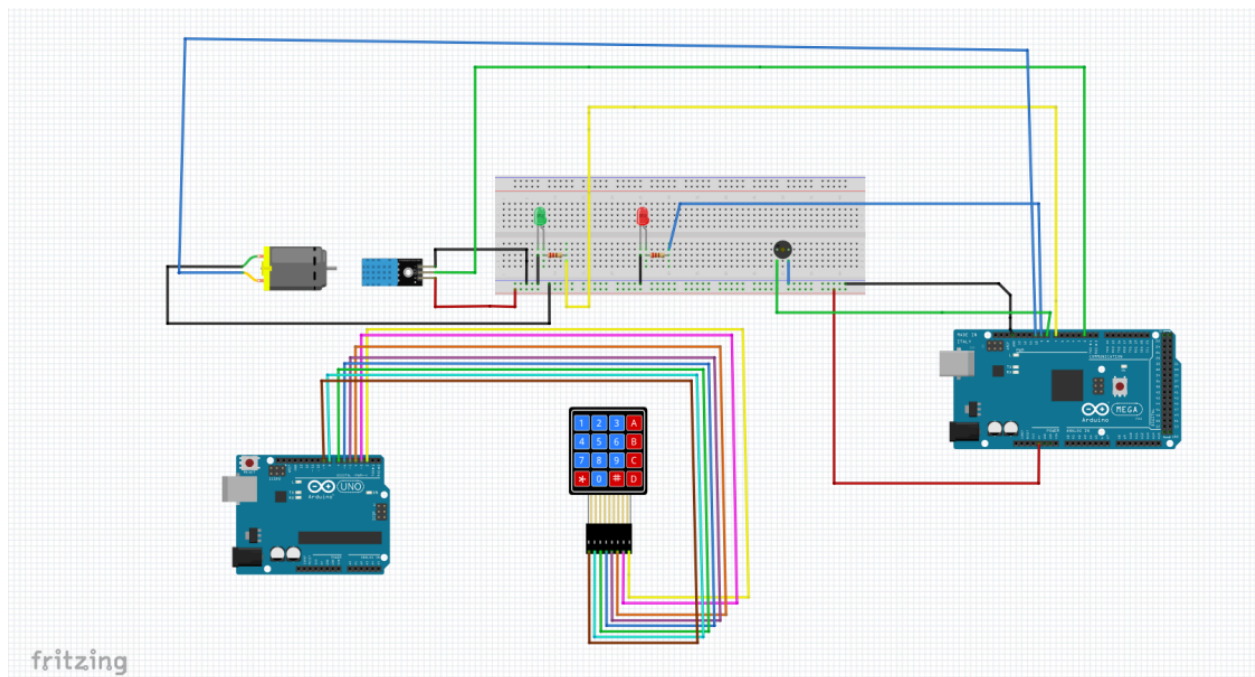
Communication Série (UART)

L'Arduino Uno et l'Arduino Mega peuvent échanger des informations via la communication **série** en utilisant les ports **TX** et **RX**.

- **TX** de l'Arduino Mega (broche 1) → **RX** de l'Arduino Uno (broche 0)
- **RX** de l'Arduino Mega (broche 0) → **TX** de l'Arduino Uno (broche 1)

Vous pourrez ainsi envoyer des données de température et d'humidité du Mega vers l'Uno. L'Uno pourra ensuite utiliser ces données pour contrôler le ventilateur et afficher les informations sur l'écran.

2. Schéma Récapitulatif



IV. FONCTIONNEMENT DE L'APPLICATION

1. Authentification

L'authentification est un élément clé pour sécuriser l'accès à l'application et garantir que seules les personnes autorisées (utilisateurs ou administrateurs) peuvent accéder aux données sensibles ou contrôler le système (comme la gestion de la température, l'humidité, et le ventilateur). Dans ce projet, l'authentification est implémentée en utilisant un système basé sur des **JSON Web Tokens (JWT)**.

a. Authentification par Email et Mot de Passe

C'est la méthode la plus courante et la plus sécurisée pour authentifier un utilisateur dans une application. Dans ce cas :

- **Email** : L'utilisateur doit entrer son adresse email pour s'identifier.
- **Mot de Passe** : Un mot de passe est utilisé pour valider l'identité de l'utilisateur. Ce mot de passe est généralement haché côté serveur pour garantir la sécurité des données.

L'authentification avec **email et mot de passe** repose sur le principe classique d'enregistrement et de connexion via une combinaison d'identifiants et de mot de passe.

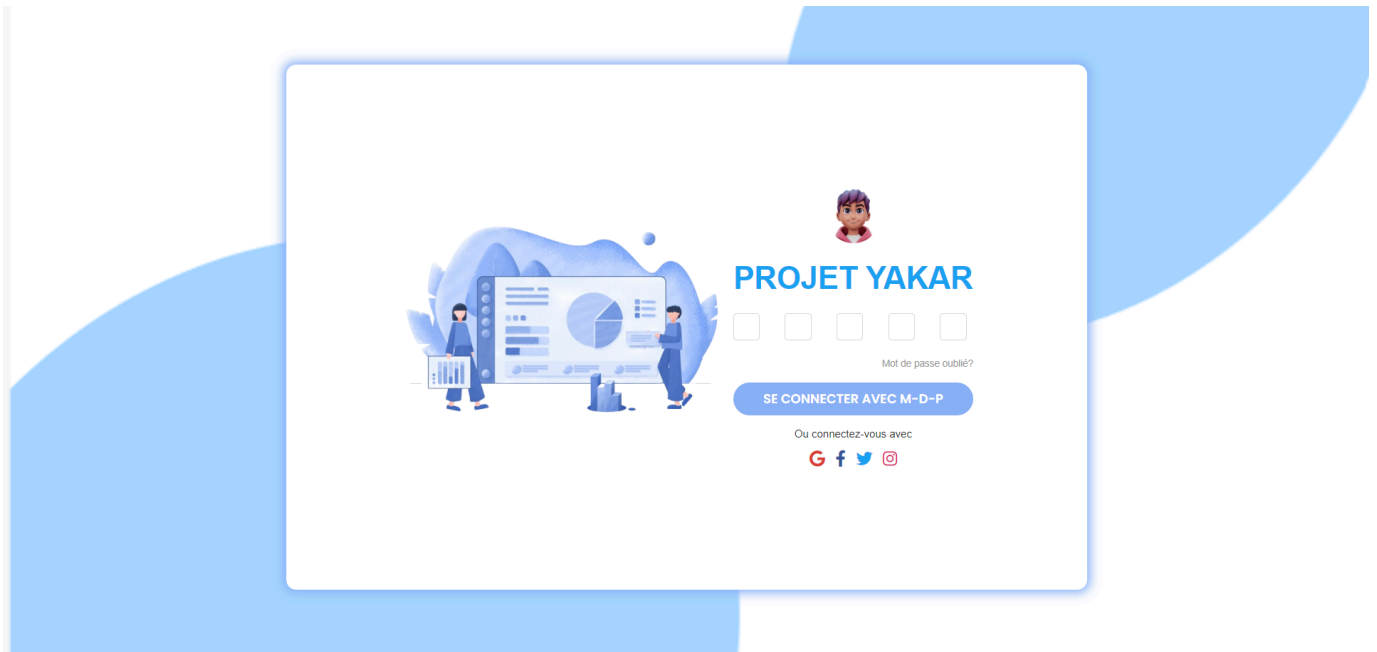


b. Authentification par Code Secret

Une autre option est d'utiliser un **code secret** (un PIN, un mot de passe à usage unique, ou un autre type de code) pour l'authentification. Ce code est statique (fixe), généré automatiquement lors de l'inscription.

Cas d'utilisation typique :

- **PIN ou Code Secret Statique** : L'utilisateur reçoit un code secret lors de son inscription (ou via un autre moyen comme un SMS ou un email) et doit entrer ce code pour se connecter.



2. Tableau de bord admin

L'interface administrateur de l'application offre une série de fonctionnalités permettant de gérer et surveiller à distance les conditions environnementales du magasin, ainsi que les utilisateurs. Voici un résumé des principales fonctionnalités proposées dans le template administrateur :

a. Surveillance en Temps Réel

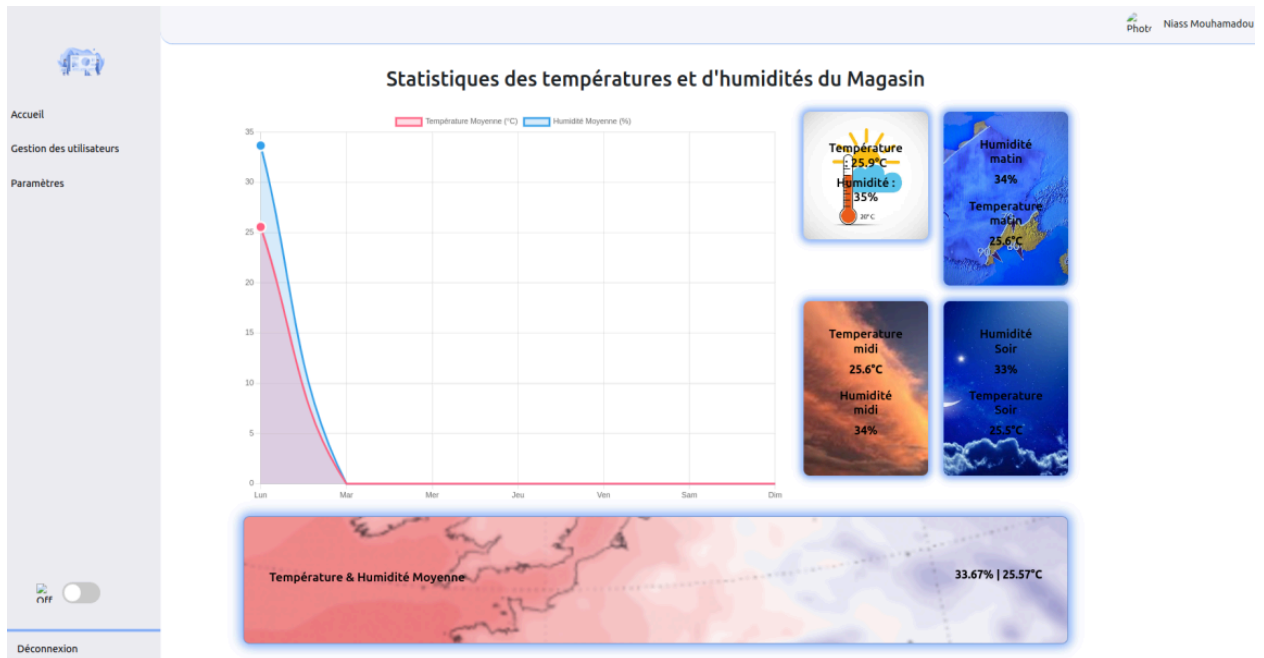
- Affichage de la Température et de l'Humidité Actuelles : Les valeurs de température et d'humidité sont mises à jour en temps réel, permettant à l'administrateur de suivre les conditions du magasin en continu.

b. Consultation des Données Historiques

- Affichage des Données à des Heures Spécifiques (10h, 14h, 17h) : L'administrateur peut consulter les relevés de température et d'humidité collectés à des moments spécifiques de la journée, à savoir 10h, 14h et 17h.

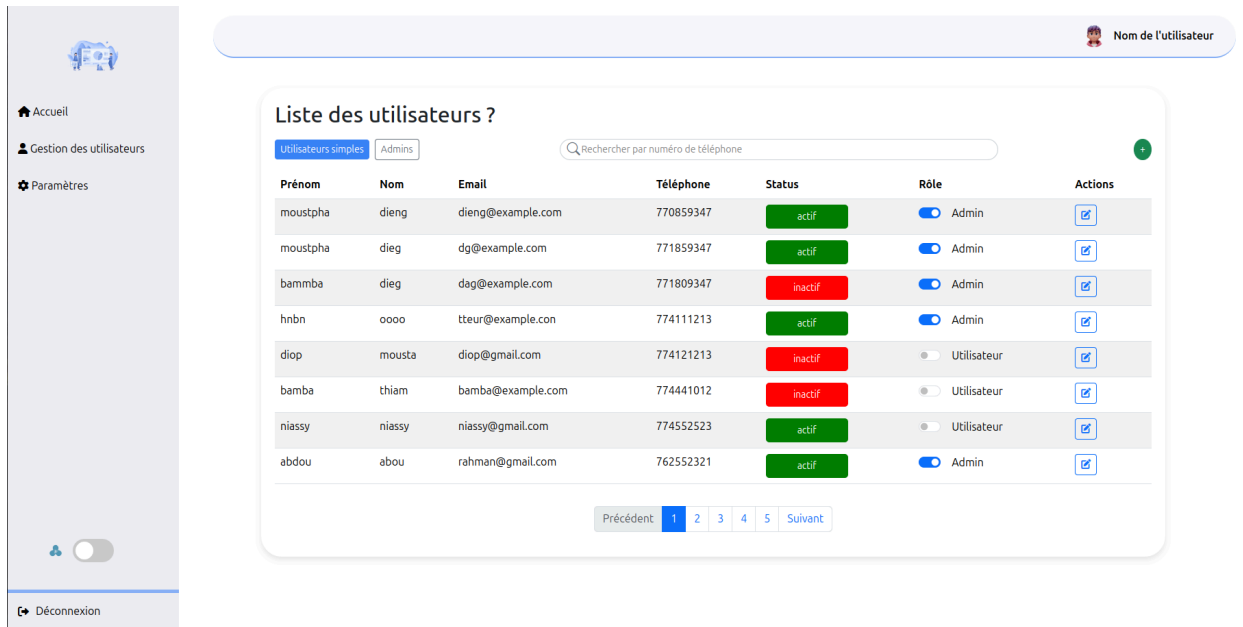
c. Calcul des Moyennes

- **Moyenne de la Journée** : Si les données des trois heures spécifiques (10h, 14h, 17h) sont présentes, l'application calcule et affiche la moyenne de la température et de l'humidité pour la journée.
- **Moyenne de la Semaine** : L'administrateur peut consulter la moyenne des températures et des humidités collectées sur la semaine, calculée sur les mêmes horaires (10h, 14h, 17h) pour chaque jour.



d. Gestion des Utilisateurs

- **Création d'Utilisateurs** : L'administrateur peut ajouter de nouveaux utilisateurs en renseignant leur email, mot de passe et rôle (utilisateur ou administrateur).
- **Modification des Utilisateurs** : Il est possible de modifier les informations d'un utilisateur existant (email, mot de passe, rôle).
- **Suppression d'Utilisateurs** : L'administrateur peut supprimer un utilisateur du système.
- **Gestion des Rôles** : L'administrateur peut attribuer un rôle d'administrateur ou d'utilisateur simple à chaque compte.



2. Tableau de bord Utilisateur Simple

L'interface utilisateur simple permet d'accéder à certaines fonctionnalités de l'application pour suivre l'état des conditions environnementales du magasin (température et humidité), mais avec un accès limité par rapport à l'administrateur. Voici un résumé des principales fonctionnalités proposées pour un utilisateur simple :

a. Surveillance en Temps Réel

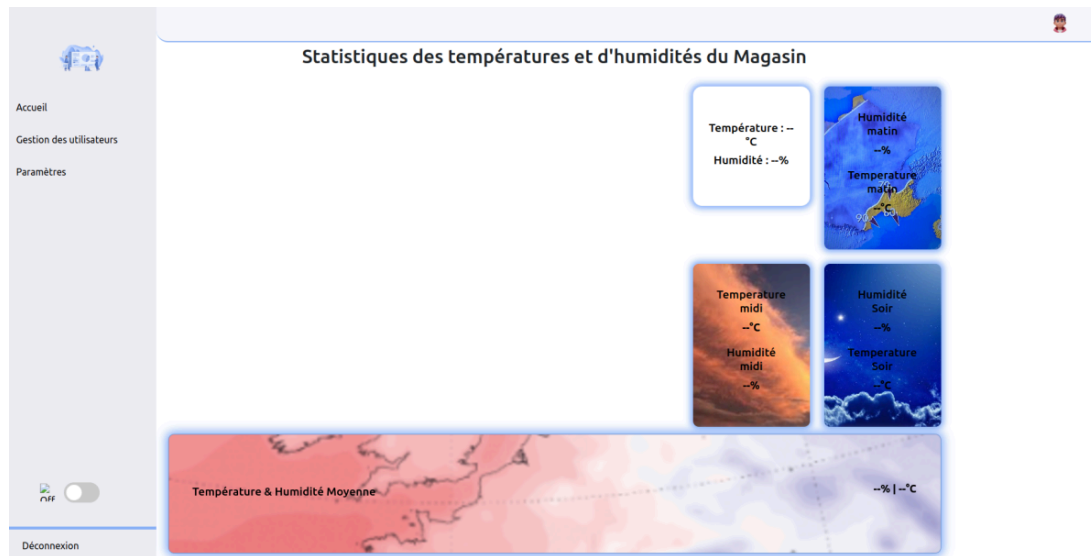
- **Affichage de la Température et de l'Humidité Actuelles :** L'utilisateur peut consulter les valeurs de température et d'humidité en temps réel, mises à jour régulièrement pour suivre les conditions du magasin en continu.

b. Consultation des Données Historiques

- **Données à des Heures Spécifiques (10h, 14h, 17h) :** L'utilisateur peut accéder aux relevés de température et d'humidité collectés à des heures spécifiques dans la journée, à savoir 10h, 14h et 17h. Ces données sont affichées sous forme de graphiques ou de tableaux.

c. Affichage des Informations en Temps Réel

- **Vue en Temps Réel de la Température et de l'Humidité :** L'utilisateur peut consulter l'état actuel des conditions dans le magasin, ce qui lui permet de suivre l'évolution en temps réel des données de température et d'humidité.



V. CONCLUSION

Le projet a permis de mettre en œuvre une solution complète de contrôle de la température et de l'humidité via une interface web, avec un système d'alertes automatisées et une gestion des utilisateurs. Le développement a intégré des technologies modernes tant pour le hardware (Arduino, capteurs, relais) que pour le software (Node.js, Angular, MongoDB, WebSocket).

Le système offre une surveillance en temps réel et à des moments précis de la journée, permettant de gérer efficacement l'environnement du magasin. Grâce à une interface bien conçue, l'utilisateur peut facilement suivre l'évolution des données et interagir avec le système, que ce soit en tant qu'utilisateur simple ou administrateur.

Le projet a également permis de renforcer les compétences en matière de câblage électronique, de développement web, de gestion de bases de données et de travail collaboratif en équipe, tout en appliquant les principes de la méthode agile pour assurer un développement progressif et bien structuré.

TABLEAU DE MATIÈRE

I. CHOIX DES TECHNOLOGIES

II. UTILES DE DÉVELOPPEMENT

1. Visual Studio Code (VS Code)

4. Angular (Front-End)

III. PRODUCTION DE L'APPLICATION

Détails du câblage :

IV. FONCTIONNEMENT DE L'APPLICATION

a. Authentification par Email et Mot de Passe

b. Authentification par Code Secret

Cas d'utilisation typique :

a. Surveillance en Temps Réel

b. Consultation des Données Historiques

c. Calcul des Moyennes

d. Gestion des Utilisateurs

a. Surveillance en Temps Réel

b. Consultation des Données Historiques

c. Affichage des Informations en Temps Réel

V. CONCLUSION