

# **Cahier des Charges : Application de Monitoring de Température et d'Humidité**

## **#### \*\*1. Introduction\*\***

Ce projet vise à développer une application web de monitoring de température et d'humidité en utilisant le framework Django pour le backend et n'importe quel framework pour la partie frontend (JavaScript, React, Angular, etc.). L'objectif est de surveiller les variations de température dans un frigo positif (+2 à +8 degrés Celsius) à l'aide d'un capteur DHT11 connecté à une carte ESP8266.

## **#### \*\*2. Objectifs du projet\*\***

- Développer une application IoT capable de collecter des données de température et d'humidité via un capteur DHT11 toutes les 20 minutes.
- Transmettre ces données via WiFi vers un serveur local ou cloud.
- Afficher les données de manière visuelle sur une application web développée avec Django et un framework frontend de votre choix.
- Mettre en place un système d'alerte en cas de température hors de la plage normale.

## **#### \*\*3. Fonctionnalités attendues\*\***

### **##### \*\*3.1. Acquisition des données\*\***

- Le capteur DHT11 connecté à l'ESP8266 doit collecter les données de température et d'humidité toutes les 20 minutes.
- Transmission des données via WiFi vers un serveur Django (local ou cloud).

### **##### \*\*3.2. Application Web Backend\*\***

- Utiliser Django pour créer une API REST permettant de recevoir et stocker les données envoyées par le capteur.
- Stockage des données dans une base de données (SQLite, PostgreSQL, etc.).

### **##### \*\*3.3. Application Web Frontend\*\***

- Utiliser un framework frontend de votre choix (JavaScript, React, Angular, etc.) pour afficher les données de manière visuelle.
- Afficher les graphiques de l'évolution de la température et de l'humidité sur différentes périodes.

- Développer une interface utilisateur permettant de visualiser les alertes et les messages.

#### ##### \*\*3.4. Système d'alerte\*\*

- Définir les seuils de température :
  - \*\*Température normale\*\* : 2 à 8 degrés Celsius.
  - \*\*Température critique\*\* : 0 à 2 ou 8 à 10 degrés Celsius.
  - \*\*Température sévère\*\* : en dehors de ces plages.
- L'application doit envoyer une alarme :
  - Dès qu'une température n'est pas normale, un message est envoyé à l'opérateur 1.
  - Après 4 alarmes consécutives, le message est également envoyé à l'opérateur 2.
  - Après 7 alarmes consécutives, un message est également envoyé à l'opérateur 3.
- Si la température redevient normale, le processus d'alerte recommence.

#### #### \*\*4. Technologies à utiliser\*\*

- \*\*Backend\*\* : Django, Django REST Framework, Python.
- \*\*Frontend\*\* : Choix libre (JavaScript, React, Angular, Vue.js, etc.).
- \*\*Base de données\*\* : SQLite, PostgreSQL, MySQL, etc.
- \*\*IoT Hardware\*\* : ESP8266, capteur DHT11.
- \*\*Hébergement\*\* : Déploiement de l'application web sur PythonAnywhere.

#### ##### \*\*5. Livrables attendus\*\*

- Code source complet de l'application.
- Documentation du code et du projet.
- Manuel d'utilisation pour l'opérateur.
- Présentation de la solution et démonstration du système fonctionnel.

#### ##### \*\*6. Critères d'évaluation\*\*

- Fonctionnalité : Le système doit fonctionner correctement, avec un flux de données du capteur à l'application web.
- Qualité du code : Respect des bonnes pratiques de développement en Python et Django.
- Design de l'interface : Interface utilisateur intuitive et bien conçue.

- Réactivité : Application web capable de gérer les changements de température en temps réel.
- Respect des délais : Le projet doit être terminé et livré dans le temps imparti.

#### #### \*\*7. Délais et échéances\*\*

- Phase d'étude : 1 semaine
- Phase de développement Backend : 2 semaines
- Phase de développement Frontend : 2 semaines
- Intégration et tests : 1 semaine
- Déploiement et finalisation : 1 semaine

#### #### \*\*8. Contraintes et recommandations\*\*

- Respecter les normes de sécurité pour la transmission des données.
- Utiliser des bibliothèques fiables et bien documentées.
- Penser à l'évolutivité de l'application pour intégrer d'autres capteurs ou fonctionnalités à l'avenir.