-- Ceci est un document explicatif pour un système de logement **éco-responsable**.

### ****Installation des bibliothèques nécessaires pour garantir un environnement de fonctionnement correct :****

pip install fastapi uvicorn jinja2 requests pydantic sqlite3

### ****Présentation :****

Il s'agit d'un **Eco-Home Management System**, conçu pour gérer et surveiller les données des maisons intelligentes, y compris la consommation d'énergie, l'état des capteurs, les informations de facturation et les prévisions météorologiques.

### ****Fonctionnalités principales :****

#### ****Page d'accueil :****

Fournit un message de bienvenue et présente les fonctionnalités du site.

#### ****Gestion des logements (/api/logements) :****

Affiche les données de tous les logements, y compris les pièces et les capteurs associés.

Fournit des données au format JSON pour faciliter l'affichage côté client.

#### ****Gestion de la consommation énergétique (/api/consumption et /api/savings) :****

**/api/consumption** : Affiche les données de consommation énergétique pour chaque logement et ses pièces.

**/api/savings** : Calcule la consommation totale d'électricité pour chaque logement et renvoie les informations de facturation correspondantes.

#### ****Surveillance de l'état des capteurs (/api/sensors\_status et /api/sensors/configuration) :****

**/api/sensors\_status** : Affiche l'état visible des capteurs et les données de mesure récentes.

**/api/sensors/configuration** : Gère la visibilité des capteurs.

#### ****Gestion des données de mesure des capteurs (/api/sensors/insert\_measure)****

Fournit une API pour insérer les données de mesure des capteurs (par exemple, température et humidité).

#### ****Prévisions météorologiques (/weather/5days)****

Utilise **WeatherAPI** pour fournir les prévisions météorologiques des 5 prochains jours, y compris les températures maximales et minimales ainsi que les conditions météorologiques.

#### ****Support des pages front-end****

Chaque fonctionnalité principale dispose d'une page front-end correspondante, rendue avec les modèles **Jinja2**.  
Exemples de pages :

* **Page d'accueil** : home.html
* **Page de visualisation de la consommation d'énergie** : consumption.html
* **Page de statut des capteurs** : sensors\_status.html
* **Page des factures et des économies d'énergie** : savings.html
* **Page de configuration des capteurs** : configuration.html
* **Page d'insertion des données des capteurs** : insert.html
* **Page d'affichage DHT11** : temperature.html
* **Page des prévisions météorologiques** : weather.html

### ****Instructions d'exécution :****

Ouvrez le dossier **eco\_2** avec **VScode**.

Trouvez le fichier **main.py**, dans lequel le code pour exécuter le serveur a déjà été ajouté :

import uvicorn

uvicorn.run(app, host="0.0.0.0", port=8000)

#### ****Description pour le DHT11****

**Connexion matérielle**

Ouvrez le fichier **http\_html.ino**.Connectez l'**ESP8266** et le capteur **DHT11** selon le schéma indiqué.



**Exécution du programme**

Téléversez le programme sur l'ESP8266 et exécutez-le.

Lancez le serveur.

**Affichage des données**

Vous pourrez voir sur le site web un graphique montrant l'évolution de la température et de l'humidité au fil du temps.