**Statie meteo – Proiect IoT Anghel David-Tudor**

**Capitolul 1: Introducere**

**Descriere generala a proiectului:**

Acest proiect are ca scop dezvoltarea unui sistem IoT care permite monitorizarea si analiza temperaturii utilizand un microcontroler ESP32 simulat in mediul Wokwi. Serverul ESP32 functioneaza ca un server HTTP capabil sa proceseze cereri de la o aplicatie web si sa returneze date despre temperatura dintr-o locatie specificata. Aplicatia web permite utilizatorilor sa introduca coordonate geografice sau sa le selecteze direct de pe o harta interactiva, pentru a obtine informatii despre conditiile meteo, prin intermediul API-ului Open Meteo.

Sistemul este gandit pentru a functiona exclusiv in simulare, fara componente hardware fizice, dar include flexibilitate pentru a fi extins in viitor cu senzori fizici (ex: senzori de temperatura si umiditate) si actuatori (ventilatoare sau LED-uri).

**Obiectivele proiectului:**

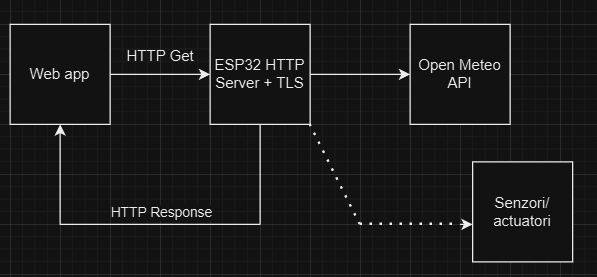
1. Dezvoltarea unui sistem IoT simulat care sa monitorizeze date de temperatura pentru locatii specificate.
2. Integrarea unei harti interactive in aplicatia web pentru a facilita selectia coordonatelor geografice.
3. Crearea unui server ESP32 simulat in Wokwi, capabil sa gestioneze cereri HTTP si sa returneze date meteo preluate prin API-ul Open Meteo.
4. Asigurarea unui flux de date sigur si eficient prin utilizarea protocoalelor HTTPS.
5. Pregatirea arhitecturii pentru extinderea cu senzori si actuatori in simulari viitoare.

**Utilizarea senzorilor si actuatorilor:**

* Senzori (viitor):
  + In simulare, se vor putea adauga senzori de temperatura (ex: DHT22) pentru a colecta date meteo locale.
* Actuatori (viitor):
  + Simulare de ventilatoare sau LED-uri pentru notificari vizuale sau notificari pentru telegram, sonore sau pentru controlul mediului (in serii sau alte aplicatii).

**Capitolul 2: Arhitectura**

**Diagrama topologiei retelei**



**Descrierea componentelor**

1. Aplicatia Web cu harta interactiva:
   * Ruleaza pe un server local (ex: Python HTTP server) si permite utilizatorilor sa selecteze coordonatele geografice printr-o interfata grafica interactiva.
   * Utilizeaza biblioteca Leaflet.js pentru a afisa si gestiona harta interactiva.
   * Transmite coordonatele catre serverul ESP32 prin cereri HTTPS GET.
   * Afiseaza rezultatele in format usor de interpretat (JSON, mesaje text).
2. ESP32 Server HTTP (simulat in Wokwi):
   * Functioneaza in mediul de simulare Wokwi, permitand testarea completa fara hardware fizic.
   * Ruleaza un server HTTP care primeste cereri cu coordonate geografice de la aplicatia web.
   * Preia date despre temperatura de la API-ul Open Meteo.
   * Returneaza raspunsuri JSON cu informatii despre temperatura curenta.
   * Utilizeaza protocoale sigure (HTTPS) pentru criptarea datelor.
3. API-ul Open Meteo:
   * Serviciu web utilizat pentru obtinerea datelor meteo actuale.
   * Este interogat de serverul ESP32 simulat prin cereri HTTPS GET.
   * Returneaza datele in format JSON, inclusiv temperatura curenta.
4. Extensii viitoare: senzori si actuatori simulat in Wokwi:
   * In Wokwi, pot fi adaugati senzori (ex: DHT22) pentru a colecta date locale simulate.
   * Actuatorii, cum ar fi LED-uri sau ventilatoare, pot fi integrati pentru simularea reactiilor la conditiile meteo.

**Alegerea protocoalelor de comunicare**

1. HTTPS (HTTP securizat):
   * Protejeaza transferul de date intre aplicatia web si serverul ESP32.
   * Permite criptarea datelor transmise catre si de la API-ul Open Meteo.
2. JSON:
   * Format standard pentru transmiterea si procesarea datelor.
   * Usor de interpretat in aplicatia web si pe serverul ESP32.
3. CORS (Cross-Origin Resource Sharing):
   * Serverul ESP32 include header-ul Access-Control-Allow-Origin: \* pentru a permite aplicatiei web sa acceseze resursele de pe server.

**Avantaje ale arhitecturii simulative**

1. Accesibilitate si simplitate:
   * Utilizatorii pot selecta coordonatele pe harta, fara a introduce manual datele.
   * Simularea reduce costurile si permite testarea in medii virtuale.
2. Flexibilitate:
   * Wokwi permite integrarea viitoare a senzorilor si actuatorilor in proiect.
   * Sistemul poate fi extins rapid pentru a include functionalitati suplimentare.
3. Compatibilitate si scalabilitate:
   * Arhitectura bazata pe protocoale standard (HTTPS, JSON) este usor de scalat si de extins