IOT Based On Environmental Monitoring

Name of the student: Vupathi Manoj

Register Number: 723921243059

Mail ID: manojmodaliyar10@gmail.com

Hardware Components:

IoT SensIors: Temperature and humidity sensors (e.g., DHT22, DHT11, or other compatible sensors).

IoT Development Board: Raspberry Pi, Arduino, or other microcontroller boards.

Internet Connectivity: Wi-Fi module or GSM module for internet connectivity.

Power Supply: Battery or power source for the IoT devices.

Software Components:

Python for programming the IoT device.

A monitoring platform that can receive and display the incoming sensor data.

Here are the general steps:

1. Hardware Setup:

Connect the temperature and humidity sensors to the IoT development board as per the sensor's datasheet.

Ensure the IoT device has a stable power source.

Set up internet connectivity (Wi-Fi or GSM) on the IoT device.

2. Python Script for Sending Data:

You can use Python to read data from the sensors and send it to your monitoring platform. Below is an example script using a Raspberry Pi

and a DHT22 sensor. Make sure to install necessary Python libraries, such as Adafruit_DHT for sensor data.

python

Copy code

```
import Adafruit_DHT
import requests
import time

# Configuration
SENSOR_TYPE = Adafruit_DHT.DHT22  # Sensor type (DHT11, DHT22, etc.)
SENSOR_PIN = 4                # GPIO pin where the sensor is connected
MONITORING_API_URL = "http://your-monitoring-platform.com/api/data"  # Replace with your monitoring platform's API endpoint

def read_sensor_data():
    humidity, temperature = Adafruit_DHT.read_retry(SENSOR_TYPE, SENSOR_PIN)
    if humidity is not None and temperature is not None:
        return {
            "temperature": temperature,
            "humidity": humidity
        }
    else:
        return None

def send_data_to_platform(data):
```

```
    try:
        response = requests.post(MONITORING_API_URL, json=data)
        if response.status_code == 200:
            print("Data sent successfully")
        else:
            print("Failed to send data. Status code:", response.status_code)
    except Exception as e:
        print("Error sending data:", str(e))


while True:
    sensor_data = read_sensor_data()
    if sensor_data:
        send_data_to_platform(sensor_data)
    time.sleep(300)  # Send data every 5 minutes (adjust as needed)
```

3. Monitoring Platform:

Set up your monitoring platform to receive data from the IoT devices. The platform should have an API endpoint that the Python script can use to send data. You will also need to develop the platform's backend to handle incoming data and display it in a user-friendly interface.

4. Data Visualization:

Use a web application or dashboard to display the real-time environmental data collected from the IoT devices. This can be done using web development technologies like HTML, CSS, and JavaScript, and frameworks like Flask or Django for the backend.

Remember to handle security considerations like authentication and encryption when sending data to your monitoring platform, and consider backup power sources for the IoT devices to ensure uninterrupted data collection.

THANK YOU !