College code : 4212   421221106007 Phase 5

ENVIRONMENTAL MONITORING

OBJECTIVE :

Environmental monitoring aims to assess environmental quality, detect changes in ecosystems, ensure compliance with regulations, conserve biodiversity, predict and manage natural disasters, facilitate research and education, sustainably manage resources, establish early warning systems for environmental threats, inform policy development, and raise public awareness about environmental issues. By collecting and analyzing data on air, water, soil, wildlife, and natural phenomena, environmental monitoring informs decisions that balance human needs with environmental conservation efforts, ensuring a sustainable future for both people and the planet.

IoT Device Setup :

Temperature Sensor: Measures the temperature of the surrounding environment. It is crucial for understanding temperature variations which can impact ecosystems and human comfort.

Humidity Sensor: Measures the level of moisture or humidity in the air. High humidity can affect air quality and human health.

Platform Development :

Sensor Integration: Choose appropriate sensors for data collection, considering factors like accuracy, reliability, and cost. Integrate these sensors with your platform, ensuring seamless data transmission.

Data Transmission: Implement a robust communication system (Wi-

Fi, cellular, LoRa, etc.) to transmit data from sensors to the platform securely and in real-time.

Data Storage: Set up a database or cloud storage system to store the collected data. Ensure the storage system can handle large volumes of data and is scalable for future expansion.

Data Processing: Develop algorithms for data processing and analysis. This step involves cleaning, aggregating, and interpreting raw sensor data to extract meaningful insights. Machine learning algorithms can be employed for predictive analysis.

Code Implementation :

A general outline using Python programming language as an example, assuming you're working with IoT devices sending data to a cloud-based platform.

Set Up Cloud Database

Create a database or storage system on your chosen cloud platform. Note down the connection details (URL, API keys, etc.).

Install Necessary Libraries:

Install Python libraries required for communication with your IoT devices and cloud service. For example, if you are using AWS IoT, you might need **boto3** library.

Write Code for Data Retrieval:

Write Python code to retrieve data from your IoT devices. This might involve using libraries specific to your devices. For example, if you're using Raspberry Pi with sensors, libraries like **Adafruit_DHT** for temperature and humidity sensors can be used.

Code for Data Processing and Analysis:

Process the raw data as per your requirements. This could include data cleaning, aggregating, and applying algorithms for analysis. Python offers libraries like **pandas** for data manipulation and analysis.

```
import pandas as pd


# Process raw data (replace with your processing logic)

raw_data = get_raw_data_from_sensors() processed_data

= process_data(raw_data)
```

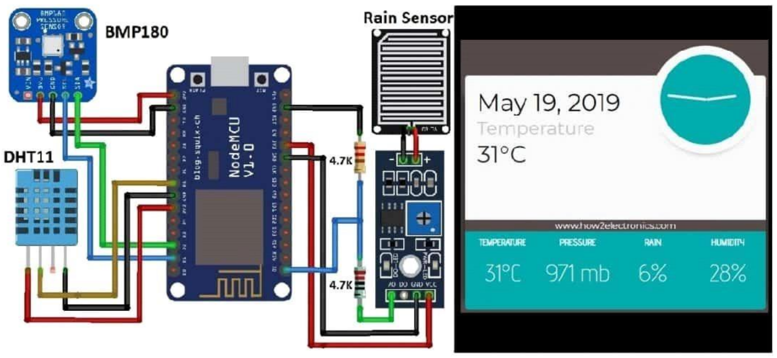
Code for Data Upload to Cloud:

```
import boto3


# Upload data to AWS S3 (replace with your cloud service logic)
s3 = boto3.client('s3',
aws_access_key_id='YOUR_ACCESS_KEY',
aws_secret_access_key='YOUR_SECRET_KEY')

s3.upload_file('processed_data.csv', 'your-bucket-name', 'processed_data.csv')
```

Explanation and Details :

Environmental monitoring is vital for understanding, preserving, and sustainably managing our natural environment. By employing cutting-edge technologies and fostering international collaboration, we can address current environmental challenges and work towards a greener, healthier future for all.

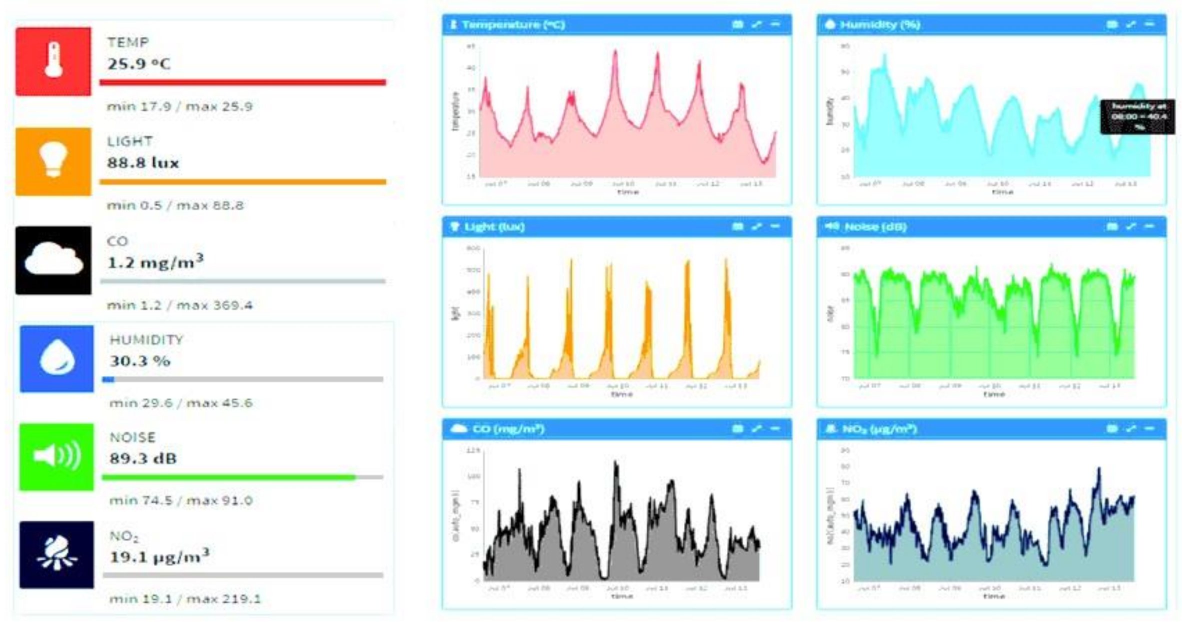
Circuit diagram of Environmental Monitoring :



Code of running above circiut :

```
import Adafruit_DHT

import time


# Set up the DHT11 sensor

DHT_SENSOR = Adafruit_DHT.DHT11

DHT_PIN = 4  # GPIO pin number where the DHT11 sensor is connected (BCM
numbering)


# Main loop for continuous monitoring  while

True:

    # Attempt to read data from the DHT11 sensor      humidity, temperature =

Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)      # Check if data

reading was successful      if humidity is not None and temperature is not

None:

        # Print temperature and humidity values        print(f'Temperature:

{temperature:.2f}°C, Humidity: {humidity:.2f}%')
```

```
    else:
        # Print an error message if reading failed
print('Failed to retrieve data from DHT11 sensor.
```



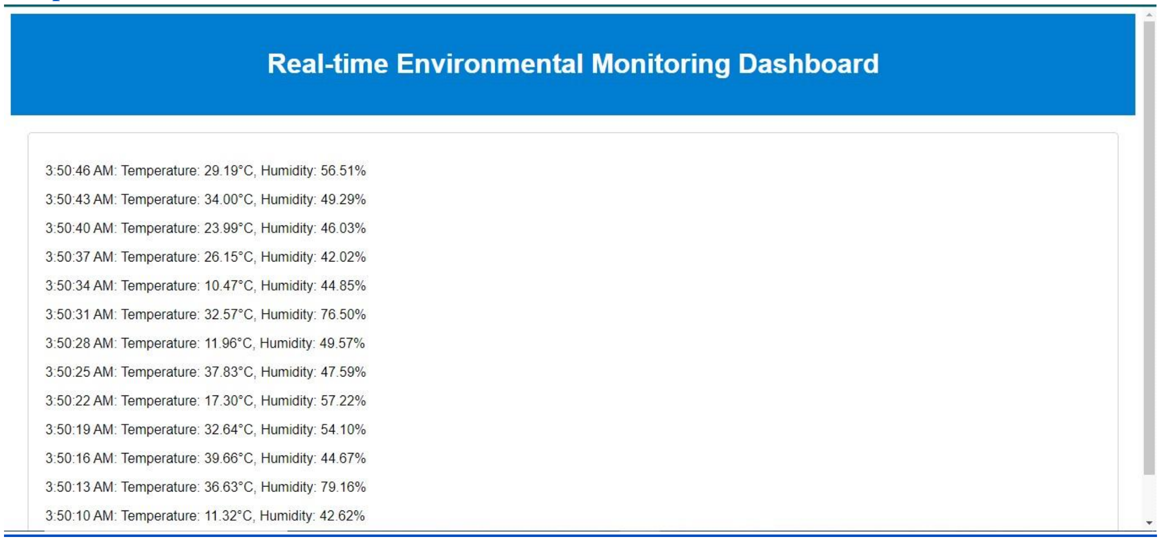
Platform UI Code for Environmental Monitoring :

```
import React, { useEffect, useState } from 'react';
import './App.css'; import Chart from
'chart.js/auto';
```

```
function App() {   const [data,
setData] = useState([]);

  useEffect(() => {     fetch('API_ENDPOINT')
.then((response) => {         if (!response.ok) {
throw new Error('Network response was not ok');
        }
        return response.json();
      })
      .then((data) => setData(data))
      .catch((error) => console.error('Error fetching data:', error));
  }, []);

  useEffect(() => {
if (data.length > 0) {
const ctx =
document.getElement
ById('data-
chart').getContext('2d'
);      new Chart(ctx,
{        type: 'line',
data: {
          labels: data.map((item) => item.timestamp),
          datasets: [
            {
              label: 'Temperature (°C)',                data:
data.map((item) => item.temperature),
borderColor: 'rgba(255, 99, 132, 1)',
backgroundColor: 'rgba(255, 99, 132, 0.2)',
            },
            {
              label: 'Humidity (%)',                data:
data.map((item) => item.humidity),
borderColor: 'rgba(54, 162, 235, 1)',
backgroundColor: 'rgba(54, 162, 235, 0.2)',
            },
          ],
        },
        options: {
scales: {             x: {
type: 'linear',
position: 'bottom',
            },
          },
        },
      });
    }
```

```
  }, [data]);

  return (
    <div className="App">
      <h1>Environmental Monitoring Dashboard</h1>
      <div className="chart-container">
        <canvas id="data-chart"></canvas>
      </div>
    </div>
  );
}
```

Output :



Real-time Environmental Monitoring Dashboard

```
3:50:46 AM: Temperature: 29.19°C, Humidity: 56.51%
3:50:43 AM: Temperature: 34.00°C, Humidity: 49.29%
3:50:40 AM: Temperature: 23.99°C, Humidity: 46.03%
3:50:37 AM: Temperature: 26.15°C, Humidity: 42.02%
3:50:34 AM: Temperature: 10.47°C, Humidity: 44.85%
3:50:31 AM: Temperature: 32.57°C, Humidity: 76.50%
3:50:28 AM: Temperature: 11.96°C, Humidity: 49.57%
3:50:25 AM: Temperature: 37.83°C, Humidity: 47.59%
3:50:22 AM: Temperature: 17.30°C, Humidity: 57.22%
3:50:19 AM: Temperature: 32.64°C, Humidity: 54.10%
3:50:16 AM: Temperature: 39.66°C, Humidity: 44.67%
3:50:13 AM: Temperature: 36.63°C, Humidity: 79.16%
3:50:10 AM: Temperature: 11.32°C, Humidity: 42.62%
```