ENVIRONMENTAL MONITORING SYSTEM

PROBLEM STATEMENT;

The main objective of environmental monitoring is to manage and minimize the impact on organization activities have an environment, either to ensure complains with law and regulations or to mitigate risks of harmful effect on the natural environment and protect the health of human beings.

Innovation:

After through research and analysis, we arrived at an innovative solution to solve the above problem as detailed in phase1 of our project.

Lowest costSmart Garden includes Nedelcu as a hub. NodeMCU is an open source IoT platform. It runs on ESP8266 Wi-Fi SoC from Espessif Systems, and hardware based on the ESP-12 module available at low cost.

**Introduction:**

An Environmental Monitoring System using the Internet of Things (IoT) is a cutting-edge solution that leverages interconnected devices, sensors, and data analytics to gather, manage, and analyze environmental data in real-time. This technology plays a crucial role in addressing environmental challenges, tracking the state of natural resources, and ensuring a sustainable future. It can be applied in various contexts, from urban areas to remote wilderness, helping to assess and manage environmental conditions more effectively.

**DESIGN THINKING APPROACH:**

Design Thinking Approach for an Integrated Environmental Monitoring System for Parks and Natural Reserves:

1. Empathize:

- Begin by understanding the diverse stakeholders involved: park managers, conservationists, park visitors, and local communities.

- Conduct interviews, surveys, and workshops to gain insights into their needs, concerns, and expectations related to environmental monitoring in parks.

2. Define:

- Clearly define the problem statement and the specific goals of the integrated monitoring system.

- Prioritize objectives such as conservation, safety, and sustainability, and consider potential trade-offs.

3. Ideate:

- Organize brainstorming sessions with a multidisciplinary team to generate innovative ideas for monitoring solutions.

- Explore technologies like IoT sensors, satellite imagery, drones, and data analytics to collect and process environmental data.

4. Prototype:

- Develop a low-fidelity prototype or mockup of the monitoring system to visualize its components and how they will interact.

- Experiment with different sensor types, data transmission methods, and data visualization tools.

5. Test:

- Conduct field tests and simulations to assess the prototype's functionality and usability.

- Gather feedback from potential users and stakeholders to refine the system's design based on real-world insight

**Overview:**

The use of IoT in an EMS allows for the monitoring and control of various environmental parameters, such as air quality, water quality, energy consumption, waste management, and more. Sensors and devices can be deployed to collect data on these parameters, which is then transmitted to a central system for analysis and action.

The real-time nature of IoT data allows organizations to respond quickly to environmental incidents or deviations from set targets. For instance, if a sensor detects a sudden increase in air pollution levels, an alert can be generated, enabling immediate action to be taken to mitigate the issue.

In summary, incorporating IoT into an EMS provides organizations with the ability to collect and analyze real-time environmental data, leading to improved environmental performance, resource efficiency, and sustainability. It allows for proactive environmental management, reduces costs, and enhances overall operational effectiveness.

**VARIOUS TOOLS USED FOR THIS PROJECT :**

**Sensors**: IoT environmental monitoring systems rely on a variety of sensors designed to measure parameters like temperature, humidity, air quality, water quality, soil moisture, radiation levels, and more. These sensors are strategically deployed to collect real-time data.

**IoT Devices:** These are the hardware components that house sensors, process data, and facilitate communication. Devices like Raspberry Pi, Arduino, or specialized IoT modules are commonly used to collect data from sensors and transmit it to central systems.

**Communication Networks:** Data from sensors is transmitted using various communication protocols, such as Wi-Fi, Bluetooth, LoRaWAN, Zigbee, or cellular networks. The choice of network depends on the specific application's range and data transfer requirements.

**IoT Platform:** Data collected by sensors is sent to an IoT platform or cloud service, such as AWS IoT, Google Cloud IoT, or Microsoft Azure IoT. These platforms provide storage, data processing, real-time monitoring, and visualization tools.

**Data Processing and Analytics:** The collected data is processed and analyzed to derive valuable insights. Advanced analytics techniques may be used to detect trends, anomalies, and patterns within the data.

1.PROPOSED SYSTEM :

•The program for NodeMCU can be written in any programming language. The Arduino software provides a better Integrated Development Environment (IDE) for programming the NodeMCU. It is a cross-platform application written in Java.

•This software consists of various features which include code editor, text cutting and pasting, replacing text and searching, brace matching, automatic indenting, and syntax highlighting.

•It usually consists of a central microcontroller to which other objects are connected. The smart garden consists of NodeMCU as a hub to which different types of sensors such as moisture sensor, humidity sensor, temperature sensor and ultrasonic sensor are connected.

• The ultrasonic sensor is connected to a water tank which indicated the level of water in the tank. Other sensors are connected to their respective positions and these sensors send the data to NodeMCU which consists of an inbuilt Wi-Fi technology.

2. NODEMCU ;

platform. It runs o•Smart Garden includes NodeMCU as a hub. NodeMCU is an open source IoT n ESP8266 Wi-Fi SoC from Espessif Systems, and hardware based on the ESP-12 module available at lowest cost.

•It is a Single – board microcontroller consists of 128kBytes of memory and 4Mbytes of storage. It was designed to for easyprogramming and allows easy prototyping for developers.

3. ARDUINO:

This software consists of various features which include code editor, text cutting and pasting, replacing text and searching, brace matching, automatic indenting, and syntax highlighting n.

•The program for NodeMCU can be written in any programming language. The Arduino software provides a better Integrated Development Environment (IDE) for programming

Coding:

import network

import time

from machine import Pin,ADC

import dht

import ujson

from umqtt.simple import MQTTClient

# MQTT Server Parameters

MQTT\_CLIENT\_ID = "micropython-weather-demo"

MQTT\_BROKER    = "broker.mqttdashboard.com"

MQTT\_USER      = ""

MQTT\_PASSWORD  = ""

MQTT\_TOPIC     = "wokwi-weather"

sensor = dht.DHT22(Pin(15))

MQ7=ADC(Pin(35))

MQ8=ADC(Pin(32))

button=Pin(34,Pin.IN)

led=Pin(33,Pin.OUT)

min\_rate=0

max\_rate=4095

print("Connecting to WiFi", end="")

sta\_if = network.WLAN(network.STA\_IF)

sta\_if.active(True)

whilenot sta\_if.isconnected():

  print(".", end sta\_if.connect('Wokwi-GU

EST', '')

="")

  time.sleep(0.1)

print(" Connected!")

print("Connecting to MQTT server... ", end="")

client = MQTTClient(MQTT\_CLIENT\_ID, MQTT\_BROKER, user=MQTT\_USER, password=MQTT\_PASSWORD)

client.connect()

print("Connected!")

prev\_weather = ""

whileTrue:

  CO\_sensor=(MQ7.read())\*100/(max\_rate)

  print("CO Sensor value: " + "%.2f" % CO\_sensor +"%")

  Hydrogen\_sensor=(MQ8.read())\*100/(max\_rate)

  print("Soil Sensor value: " + "%.2f" % Hydrogen\_sensor +"%")

  button\_value=button.value()

  if button\_value == True:

    led.value(000)

    print("It's Raining")

  else:

    led.value(0)

  print("Measuring weather conditions... ", end="")

  sensor.measure()

  message = ujson.dumps({

    "temp": sensor.temperature(),

    "humidity": sensor.humidity(),

  })

  if message != prev\_weather:

    print("Updated!")

    print("Reporting to MQTT topic {}: {}".format(MQTT\_TOPIC, message))

    client.publish(MQTT\_TOPIC, message)

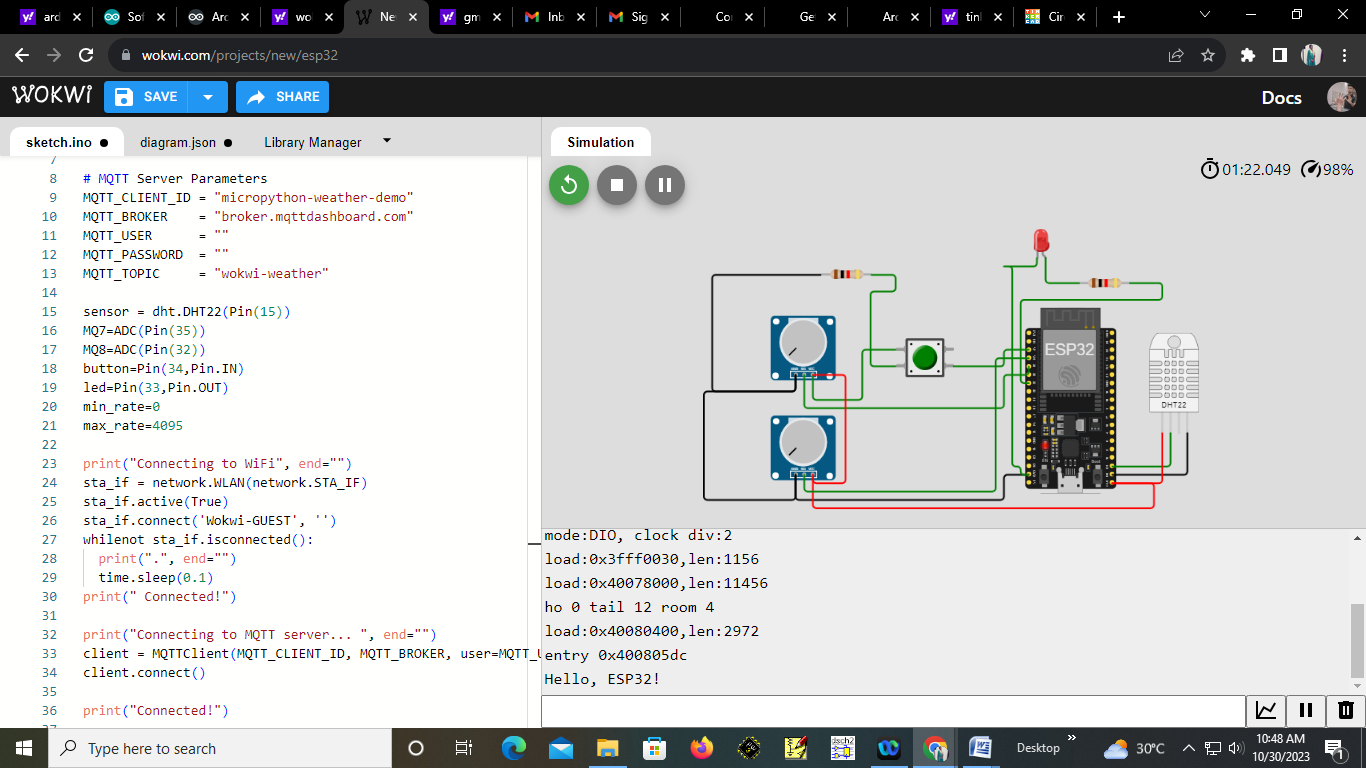
    prev\_weather = message

  else:

    print("No change")

  time.sleep(1)

RESULT:



**Conclusion:**

In conclusion, an Environmental Monitoring System using the Internet of Things (IoT) represents a transformative and highly valuable technology for addressing a wide range of environmental challenges. This system harnesses the power of interconnected sensors, devices, and data analytics to collect, manage, and analyze environmental data in real-time.