**Environmental monitoring**

**Phase3:**

IoT devices and then Developing a Python script on the IoT devices as per the project .

**Introduction:.**

The purpose of an Environmental Monitoring Program is to identify problem areas where potentially harmful microorganisms may be harboring, becoming a source of contamination

**Microcontroller Naming:**

**1.ESP32:**

It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Tensilica’s 32-bit Xtensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth.

**2.Arduino uno:**

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

**Program:**

**PYTHON CODE**

import Adafruit\_DHT

import time

# Sensor should be set to Adafruit\_DHT.DHT11,

# Adafruit\_DHT.DHT22, or Adafruit\_DHT.AM2302.

DHT\_SENSOR = Adafruit\_DHT.DHT11

DHT\_PIN = 4 # GPIO pin the sensor is connected to

def get\_temperature\_humidity():

humidity, temperature = Adafruit\_DHT.read\_retry(DHT\_SENSOR, DHT\_PIN)

if humidity is not None and temperature is not None:

return temperature, humidity

else:

return None, None

def log\_data(temperature, humidity):

with open("environment\_log.txt", "a") as file:

current\_time = time.strftime('%Y-%m-%d %H:%M:%S')

log\_entry = f"{current\_time} - Temperature: {temperature}°C, Humidity: {humidity}%\n"

file.write(log\_entry)

if \_\_name\_\_ == '\_\_main\_\_':

while True:

temperature, humidity = get\_temperature\_humidity()

if temperature is not None and humidity is not None:

log\_data(temperature, humidity)

print(f"Temperature: {temperature}°C, Humidity: {humidity}%")

else:

print("Failed to retrieve data from the sensor.")

time.sleep(60) # Adjust this time to set the interval for data collection.

**Components:**

**1.Sensors:**

These are the physical devices responsible for collecting data about various environmental parameters such as temperature, humidity, air quality, pollution levels, etc.

**2.Data Logger:**

This component stores the data collected from the sensors for further processing and analysis.

**3.Microcontroller/Single Board Computer (SBC):**

It processes the data from the sensors and communicates with the data logger. It might also handle data transmission to a cloud storage system for long-term data storage.

**4.Cloud storage:**

This is optional but often used for long-term storage of large datasets. It provides accessibility and scalability for handling vast amounts of environmental data.

**5.Data Processing Algorithms:**

These algorithms process the collected data, perform statistical analysis, and identify patterns or trends. They may also trigger alerts or warnings based on predefined thresholds.

**6.Analysis Report:**

This component involves generating insights, creating reports, and visualizing the data in a meaningful way for decision-making purposes.

**7.Visualization Interface:**

This is the user interface that allows stakeholders to interact with the system, view real-time data, access historical data, and interpret the analyzed results. It can include dashboards, charts, graphs, and other visual representations of the environmental data.