PROJECT TITLE: ENVIRONMENTAL MONITORING

PHASE- 3 DEVELOPMENT

STEPS FOR FLOWCHART

STEP 1: Start

STEP 2: Select Monitoring Parameters

- Identify the specific variables or pollutants to measure (e.g., temperature, humidity, pollutants, pH levels).

STEP 3: Set Monitoring Frequency

- Determine how often data will be collected (e.g., hourly, daily, monthly).

STEP 4: Deploy Monitoring Equipment

- Install sensors, data loggers, or other monitoring devices at chosen locations.

STEP 5: Data Collection

- Continuously or periodically collect data from monitoring equipment.

STEP 6: Data Analysis

- Process and analyze the collected data to identify trends, anomalies, or environmental issues.

STEP 7: Quality Control

- Ensure data accuracy and reliability through calibration and maintenance.

STEP 8: Interpret Results

- Draw conclusions from the data analysis and assess its impact on the environment.

STEP 9: Feedback Loop

- Use monitoring data to adjust monitoring strategies and environmental management practices.

STEP 10: End

FLOW CHART

START

INITIALISE

SENSORS

COLLECT

ALL DATA

ANALYSE

THE DATA

DISPLAY

RESULTS

TAKE ACTION

IF NECESSARY

DISPLAY RESULTS

END

MAIN PROGRAM

import Adafruit\_DHT

import RPi.GPIO as GPIO

import time

# Pin specifications for each sensor

temperature\_pin = 4 # GPIO pin for the DHT22 temperature and humidity sensor

humidity\_pin = 17 # GPIO pin for the DHT22 temperature and humidity sensor

air\_flow\_pin = 18 # GPIO pin for the air flow sensor

water\_leak\_pin = 22 # GPIO pin for the water leak detection sensor

sound\_pin = 23 # GPIO pin for the sound sensor

# Setup GPIO

GPIO.setmode(GPIO.BCM)

GPIO.setup(air\_flow\_pin, GPIO.IN)

GPIO.setup(water\_leak\_pin, GPIO.IN)

GPIO.setup(sound\_pin, GPIO.IN)

# Create an instance of the DHT sensor

dht\_sensor = Adafruit\_DHT.DHT22

while True:

# Read temperature and humidity data from DHT22 sensor

humidity, temperature = Adafruit\_DHT.read\_retry(dht\_sensor, temperature\_pin)

# Read data from the humidity sensor (if using a separate sensor)

humidity\_data = Adafruit\_DHT.read\_retry(dht\_sensor, humidity\_pin)

# Read data from air flow sensor (replace this with your specific code)

air\_flow\_data = GPIO.input(air\_flow\_pin)

# Read data from water leak detection sensor (replace this with your specific code)

water\_leak\_data = GPIO.input(water\_leak\_pin)

# Read data from sound sensor (replace this with your specific code)

sound\_data = GPIO.input(sound\_pin)

# Print the sensor data

if humidity is not None and temperature is not None:

print(f'Temperature: {temperature:.2f}°C')

print(f'Humidity: {humidity:.2f}%')

else:

print('Failed to retrieve data from the DHT22 sensor.')

if humidity\_data is not None:

print(f'Humidity Sensor Data: {humidity\_data:.2f}%')

print(f'Air Flow: {air\_flow\_data}')

print(f'Water Leak: {water\_leak\_data}')

print(f'Sound: {sound\_data}')

time.sleep(60) # Adjust the delay as needed

TEMPERATURE SENSOR

import random

import time

# Define the temperature range in Celsius

min\_temperature = 20

max\_temperature = 30

def read\_temperature():

# Simulate reading temperature data from a sensor

temperature = random.uniform(min\_temperature, max\_temperature)

return temperature

while True:

temperature = read\_temperature()

# Send the temperature data to an IoT platform or process it as needed

print(f"Temperature: {temperature}°C")

time.sleep(1) # Simulate reading every 1 second

AIR FLOW SENSOR

import random

import time

def read\_airflow():

# Simulate reading airflow data from a sensor

airflow = random.uniform(0, 100) # You can adjust the range as needed

return airflow

while True:

airflow = read\_airflow()

# Send the airflow data to an IoT platform or process it as needed

print(f"Airflow: {airflow} CFM") # CFM (Cubic Feet per Minute) is a common unit for airflow

time.sleep(1) # Simulate reading every 1 second

HUMIDITY SENSOR

import random

import time

def read\_humidity():

# Simulate reading humidity data from a sensor

humidity = random.uniform(20, 80) # You can adjust the range as needed

return humidity

while True:

humidity = read\_humidity()

# Send the humidity data to an IoT platform or process it as needed

print(f"Humidity: {humidity}%")

time.sleep(1) # Simulate reading every 1 second

SOUND SENSOR

import random

import time

def read\_sound\_intensity():

# Simulate reading sound intensity data from a sensor

sound\_intensity = random.uniform(40, 100) # You can adjust the range as needed

return sound\_intensity

while True:

sound\_intensity = read\_sound\_intensity()

# Send the sound intensity data to an IoT platform or process it as needed

print(f"Sound Intensity: {sound\_intensity} dB")

time.sleep(1) # Simulate reading every 1 second

WATER LEAK DETECTOR

import random

import time

def read\_water\_leak\_status():

# Simulate reading water leak status from a sensor

# In a real scenario, this would involve interfacing with a physical sensor

# For simulation, we'll use a random value where 0 represents no leak, and 1 represents a leak.

water\_leak = random.choice([0, 1])

return water\_leak

while True:

water\_leak = read\_water\_leak\_status()

if water\_leak == 1:

# Trigger an action when a water leak is detected

print("Water Leak Detected! Take action immediately.")

else:

print("No Water Leak Detected.")

time.sleep(1) # Simulate reading every 1 second