Phase 3 Submission

SMART WATER MANAGEMENT

Introduction:

Building an IoT water consumption monitoring system involves several steps, starting with configuring IoT sensors to measure water consumption in public places and developing a Python script to send real-time data to a data-sharing platform. Here's a step-by-step guide to get you started:

1.Select Suitable IoT Flow Meters:

Choose IoT flow meters that are appropriate for measuring water consumption in public places. Consider factors such as flow rate range, accuracy, connectivity options, and power requirements.

2.Install and Set Up Flow Meters:

Install the IoT flow meters at the desired locations in public places where you want to monitor water consumption. Ensure they are properly calibrated and securely installed to provide accurate measurements.

3.Establish Connectivity:

Connect the IoT flow meters to a communication interface, such as a microcontroller (e.g., Arduino or Raspberry Pi) that will facilitate

data collection and transmission. Use appropriate wiring and protocols (e.g., Modbus, MQTT) for communication.

4.Interface IoT Flow Meters with Microcontroller:

Configure the microcontroller to interface with the IoT flow meters and read water consumption data. Utilize the manufacturer's guidelines and documentation to ensure proper setup and communication.

5.Write Python Script:

Develop a Python script on IoT sensor to send real-time water consumption data to the data-sharing platform.

6.Implement Data Transmission:

Use MQTT, HTTP, or other IoT protocols to send real-time data from the microcontroller to your chosen data-sharing platform. This may require setting up credentials and authentication.

7.Data Storage and Processing:

Set up data-sharing platform (AWS IoT, Google Cloud IoT, etc.) to receive, store, and process the incoming data. Define data structures and rules for handling data.

8.Real-time Monitoring and Analytics:

Configure real-time monitoring and analytics tools to visualize and analyse the water consumption data. You can use services like AWS Lambda, Google Cloud Functions, or custom scripts for this.

9.Security Measures:

Ensure data security and device security by using encryption, authentication, and access control measures.

10.Testing and Calibration:

Test the system thoroughly to ensure accurate data collection. Calibrate the sensors if necessary.

11.Deployment:

Install the IoT sensors in public places where water consumption monitoring is required.

12.Maintenance:

Regularly maintain and monitor the system to ensure it continues to function correctly.

Python script on the IoT sensors to send real-time water consumption data to the data-sharing platform

```
import paho.mqtt.client as mqtt
import time
import random

MQTT_BROKER = "mqtt.example.com"
MQTT_PORT = 1883
MQTT_TOPIC = "water_consumption"

def read_water_consumption():
    return random.uniform(0, 10)
client = mqtt.Client()
def on_connect(client, userdata, flags, rc):
    if rc == 0:
        print("Connected to MQTT broker")
    else:
        print(f"Connection to MQTT broker failed with code {rc}")
client.on_connect = on_connect
client.connect(MQTT_BROKER, MQTT_PORT, 60)
while True:
    water_data = read_water_consumption()
    payload = {
        "timestamp": int(time.time()),
        "water_consumption": water_data
    }
    client.publish(MQTT_TOPIC, str(payload))
    print(f"Published: {payload}")
```

```
    time.sleep(60)
client.loop_start()
```

Conclusion:

The IoT water consumption monitoring system represents a powerful tool for addressing one of the world's most pressing challenges—efficient water resource management. By configuring IoT sensors, developing Python scripts for real-time data transmission, and implementing data-sharing platforms, we've established a framework to collect and analyse water consumption data in public places