KGiSL Institute of Technology

(Affiliated to ANNA University, Chennai and Approved by AICTE, New Delhi)

365, KGiSL Campus, Thudiyalur Road, Saravanampatti
Coimbatore – 641035

Department of Artificial Intelligence and Data Science

Name	: Isaac.S
Register Number	: 711721243035
Regulation	: R-2021
Branch	: B.Tech -Artificial Intelligence and Data Science
Project Title	: Smart Water Fountain
Semester/ Year	: V / III

Project Title: Smart Water Fountains

Phase 1: Development Part 1

Hardware and Sensors:

- Select the appropriate IoT sensors for your project. In this case, you'll need flow rate sensors and pressure sensors.
- Choose microcontrollers or IoT development boards that can connect to the sensors and send data to the platform. Common options include Raspberry Pi, Arduino, or ESP8266/ESP32.

Wiring and Sensor Integration:

- Connect the flow rate and pressure sensors to the chosen microcontroller. Refer to the sensor datasheets and microcontroller documentation for wiring instructions. Ensure power and ground connections are made correctly.
- Write code on the microcontroller to read data from these sensors. Libraries and sample code are often available for specific sensors.

IoT Platform Selection:

- Choose an IoT platform to collect and manage data. Common platforms include AWS IoT, Google Cloud IoT, or Azure IoT.
- Create an account and set up a new IoT project on the chosen platform.

Python Script for Data Transmission:

Develop a Python script to read sensor data and send it to your selected IoT platform. The exact code will depend on the microcontroller and platform you're using.

Here's an example Python script for sending data to AWS IoT using a Raspberry Pi and Python:

```
import time
import boto3
import RPi.GPIO as GPIO  # Assuming you are using a Raspberry Pi
```

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTClient

# AWS IoT Configuration
IoT_ENDPOINT = "your-iot-endpoint.amazonaws.com"
ROOT_CA = "root-CA.pem"
PRIVATE_KEY = "your-private-key.pem.key"
CERTIFICATE = "your-certificate.pem.crt"
TOPIC = "water-fountain-status"

# Initialize AWS IoT MQTT Client
myMQTTClient = AWSIoTMQTTClient("WaterFountainClient")
myMQTTClient.configureEndpoint(IoT_ENDPOINT, 8883)
myMQTTClient.configureCredentials(ROOT_CA, PRIVATE_KEY, CERTIFICATE)
myMQTTClient.configureOfflinePublishQueueing(-1)
myMQTTClient.configureDrainingFrequency(2)
myMQTTClient.configureConnectDisconnectTimeout(10)
myMQTTClient.configureMQTTOperationTimeout(5)

# Connect to AWS IoT
myMQTTClient.connect()

# Function to read sensor data
def read_sensors():
    # Replace with your code to read from flow rate and pressure sensors
    flow_rate = 0.0
    pressure = 0.0
    return flow_rate, pressure

try:
    while True:
        flow_rate, pressure = read_sensors()
        data = {
            "flow_rate": flow_rate,
            "pressure": pressure
        }
        myMQTTClient.publish(TOPIC, str(data), 1)
        time.sleep(5)  # Adjust the interval as needed

except KeyboardInterrupt:
    GPIO.cleanup()
    myMQTTClient.disconnect()
```

Data Analytics and Visualization:

- Set up the IoT platform to store incoming data and perform real-time analysis.
- You can use services like AWS Lambda, Azure Functions, or Google Cloud Functions to trigger actions based on the data received.

Monitoring and Maintenance:

- Continuously monitor the system to ensure it is collecting and transmitting data accurately.
- Implement alerting mechanisms for malfunctions or abnormal behavior.