

PROJECT DEVELOPMENT PHASE

SPRINT 1

Date	16 November 2022
Team ID	PNT2022TMID13514
Project name	Gas Leakage Monitoring & Alerting System for Industries
Maximum marks	2 marks

ANALYZE THE PREREQUISITES

Needed prerequisites for real time river water quality monitoring and control system using Internet Of Things (IoT) were

- ❖ IBM Watson IoT Platform
- ❖ Node-RED Service
- ❖ Cloudant DB

Python code:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

organization = "l27fmg"

deviceType = "12345"

deviceId = "123456"

authMethod = "token"

authToken = "123456789"

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    status=cmd.data['command']

    if status=="lighton":

        print ("led is on")

    elif status == "lightoff":
```

```

    print ("led is off")

else:

    print ("Please send proper command")

try:

    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
method": authMethod, "auth-token" : authToken}

    deviceCli = ibmiotf.device.Client(deviceOptions)

except Exception as e:

    print("Caught exception connecting device %s" % str(e))

    sys.exit()

deviceCli.connect()

while True:

    gasconcentration = random.randint(90,110)

    Humidity = random.randint(90,110)

    Temperature = random.randint(90,110)

    data = {'gasconcentration' : gasconcentration, 'Humidity' : Humidity, 'Temperature'
:Temperature}

    def myOnPublishCallback():

        print(" GasConcentration = %s PPM" % gasconcentration, "to IBM Watson")

        print(" Humidity = %s%%" % Humidity, "to IBM Watson")

        print(" Temperature = %s C" % Temperature, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)

        if not success:

            print("Not connected to IoT")

            time.sleep(10)

deviceCli.commandCallback=myCommandCallback

deviceCli.disconnect()

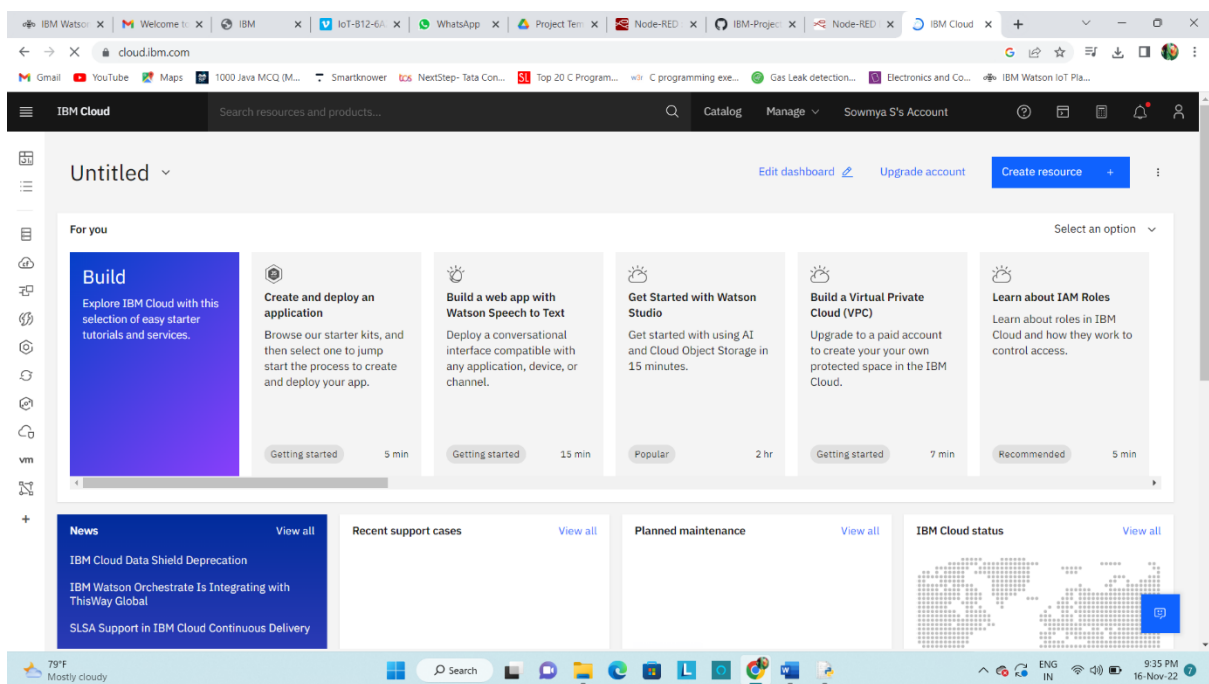
```

```
ibmpython.py - C:\Users\USER\Desktop\ibm\ibmpython.py (3.7.0)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "127fmg"
deviceType = "12345"
deviceId = "123456"
authMethod = "token"
authToken = "123456789"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status=="lightoff":
        print ("led is off")
    else:
        print ("Please send proper command")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-method":authMethod}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device %s" % str(e))
    sys.exit()
deviceCli.connect()
while True:
    gasconcentration = random.randint(90,110)
    Humidity =random.randint(90,110)
    Temperature = random.randint(90,110)
    data = {'gasconcentration': gasconcentration,'Humidity': Humidity,'Temperature':Te
    def myOnPublishCallback():
        print(" GasConcentration = %s PPM" % gasconcentration, "to IBM Watson")
        print(" Humidity = %s%%" % Humidity, "to IBM Watson")
        print(" Temperature = %s C" % Temperature, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPub
    if not success:
        print("Not connected to IoT")
        time.sleep(10)
    deviceCli.commandCallback=myCommandCallback
    deviceCli.disconnect()

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\USER\Desktop\ibm\ibmpython.py =====
2022-11-16 21:31:18,844 ibmiotf.device.Client INFO Connected successfully: d:127fmg:12345:123456
GasConcentration = 93 PPM to IBM Watson
Humidity = 108% to IBM Watson
Temperature = 104 C to IBM Watson
GasConcentration = 94 PPM to IBM Watson
Humidity = 97% to IBM Watson
Temperature = 103 C to IBM Watson
GasConcentration = 99 PPM to IBM Watson
Humidity = 94% to IBM Watson
Temperature = 96 C to IBM Watson
GasConcentration = 106 PPM to IBM Watson
Humidity = 90% to IBM Watson
Temperature = 96 C to IBM Watson
GasConcentration = 98 PPM to IBM Watson
Humidity = 99% to IBM Watson
Temperature = 106 C to IBM Watson
GasConcentration = 100 PPM to IBM Watson
Humidity = 104% to IBM Watson
Temperature = 107 C to IBM Watson
GasConcentration = 104 PPM to IBM Watson
Humidity = 96% to IBM Watson
Temperature = 110 C to IBM Watson
GasConcentration = 94 PPM to IBM Watson
Humidity = 98% to IBM Watson
Temperature = 100 C to IBM Watson
GasConcentration = 106 PPM to IBM Watson
Humidity = 96% to IBM Watson
Temperature = 96 C to IBM Watson
GasConcentration = 105 PPM to IBM Watson
Humidity = 94% to IBM Watson
Temperature = 101 C to IBM Watson
GasConcentration = 101 PPM to IBM Watson
Humidity = 103% to IBM Watson
Temperature = 106 C to IBM Watson
GasConcentration = 104 PPM to IBM Watson
Humidity = 100% to IBM Watson
Temperature = 108 C to IBM Watson
GasConcentration = 95 PPM to IBM Watson
Humidity = 93% to IBM Watson
Temperature = 97 C to IBM Watson
```

Code runs successfully and random output values are generated

Creation of IBM cloud



Procedure for the creation of IBM IOT watson

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456	Disconnected	12345	Device	Nov 12, 2022 10:49 AM	

Items per page 50 | 1-1 of 1 item

1 of 1 page

Device creation

IBM Watson IoT Platform

Browse Action Device Types Interfaces

Add Device

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456	Disconnected	12345	Device	Nov 12, 2022 10:49 AM	

Items per page 50 | 1-1 of 1 item

1 of 1 page

Identity Device Information Recent Events State Logs

Device ID 123456

Device Type 12345

Date Added Nov 12, 2022 10:49 AM

Added By sowmyasekar2306@gmail.com

Connection Status Disconnected

Generation of random values in IBM Watson

The screenshot displays the IBM Watson IoT Platform dashboard. The browser address bar shows the URL: `127fmg.internetofthings.ibmcloud.com/dashboard/devices/browse`. The dashboard header includes the IBM Watson IoT Platform logo and a user profile for `sowmyasekar2306@gmail.com` with ID `127fmg`. The main navigation bar has tabs for `Browse`, `Action`, `Device Types`, and `Interfaces`. A sidebar on the left contains icons for various IoT functions. The main content area is titled `Recent Events` and displays a table of live data streams. The table has four columns: `Event`, `Value`, `Format`, and `Last Received`. It lists four events from an `IoTSensor`, each with a JSON value string and a timestamp of 'a few seconds ago'. The footer of the dashboard shows 'Items per page 50' and '1 of 1 page'.

Event	Value	Format	Last Received
IoTSensor	{"gasconcentration":100,"humidity":109,"tempe...	json	a few seconds ago
IoTSensor	{"gasconcentration":94,"humidity":104,"temper...	json	a few seconds ago
IoTSensor	{"gasconcentration":104,"humidity":108,"tempe...	json	a few seconds ago
IoTSensor	{"gasconcentration":98,"humidity":109,"temper...	json	a few seconds ago