SPRINT – 1 DELIVERY

|  |  |
| --- | --- |
| Date | November 13 2022 |
| Team ID | PNT2022TMID07793 |
| Project Name | Real-Time River Water Quality Monitoring and Control System |
| Maximum Mark |  |

**PYTHON PROGRAM:-**

import time import sys importibmiotf.application

import ibmiotf.device import random

# #Provide your IBM Watson Device

Credentials organization = "7wqirt" deviceType

= "raspberrypi" deviceId = "12345" authMethod

= "token" authToken = "123456789" 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()

# # Connect and send a datapoint "hello" with value "world" into the cloud as an event of type

"greeting" 10 times deviceCli.connect() while True:

**#Get Sensor Data from randam value function** temp=random.randint(0,50) ph=random.uniform(0.0,14.0) turb=random.uniform(0.0,3.0)

data1={'temp':temp,'ph':ph,'turb':turb,'str1':"Not safe to drink"}

data2={'temp':temp,'ph':ph,'turb':turb,'str2':"safe to drink"}

**#print data**

def myOnPublishCallback():

print ("Published Temperature = %s C" % temp,"Ph = %.1f " % ph,"Turbidity = %.1f NTU" % turb, "to IBM Watson")

if((temp > 6 and temp < 20) and (ph > 6.5 and ph < 8.5) and turb < 1): print(data2) else:

print(data1)

success = deviceCli.publishEvent("IoTSensor", "json", data1 or data2, qos=2, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF") time.sleep(20)

# # Disconnect the device and application from the cloud

deviceCli.disconnect()

**OUTPUT:**

