**Develop the Python Script**

|  |  |
| --- | --- |
| Date | 01 November 2022 |
| Team ID | PNT2022TMID30374 |
| Project Name | Real Time River Water Quality Monitoring and Control System |

**Develop a Python Script**

**Code:**

import random import time import sys

import ibmiotf.application import ibmiotf.device

# Provide your IBM Watson Device Credentials

organization = "nqat1y" # repalce it with organization ID deviceType = "NodeMCU" # replace it with device type deviceId = "501238" # repalce with device id authMethod = "token"

authToken = "10571213" # repalce with token

def myCommandCallback(cmd):

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

if status == 'lighton': print("LIGHT ON")

elif status == 'lightoff': print("LIGHT 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:

pH = random.randint(0,100) conductivity = random.randint(0,100) T = random.randint(0,100)

oxygen = random.randint(0,100) turbidity = random.randint(0,100)

# Send Temperature & Humidity to IBM Watson

data = {'T': T,'pH':pH,'conductivity':conductivity,'oxygen':oxygen,"turbidity":turbidity}

# print data

def myOnPublishCallback():

print("Published data",data, "to IBM Watson")

success = deviceCli.publishEvent("event", "json", data, 0, myOnPublishCallback) if not success:

print("Not connected to IoTF") time.sleep(5)

deviceCli.commandCallback = myCommandCallback # Disconnect the device and application from the cloud