SPRINT 1
TEAM ID: PNT2022TMID11080

REAL TIME RIVER-WATER QUALITY MONITORING AND CONTROL SYSTEM

PYTHON CODE:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "84708c"
deviceType = "abcd"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
def myCommandCallback (cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status== "motoron":
        print ("motor is on")
    elif status == "motoroff":
        print ("motor 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 evention connecting device: %s" % str(e))
        sys.exit()
```

```
deviceCli.connect()
while True:
    temp=random.randint (90,110)
    Humid=random.randint (60,100)
    Ph=random.randint (0,14)
    Water_turbidity=random.randint (15,60)
    data = {'temp' : temp, 'Humid': Humid, 'Ph' : Ph, 'Water_turbidity' : Water_turbidity}
    def myonPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Ph = %s" % Ph,"Water Turbidity = %s NTU" % Water_turbidity, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish = myonPublishCallback)
    if not success:
        print("Not connected to IOTF")
        time.sleep (10)
        deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

OUTPUT: