

REAL TIME RIVER WATER QUALITY MONITORING AND CONTROL SYSTEM

TEAM ID : PNT2022TMID06942

TEAM LEADER : VINUPRIYA K P

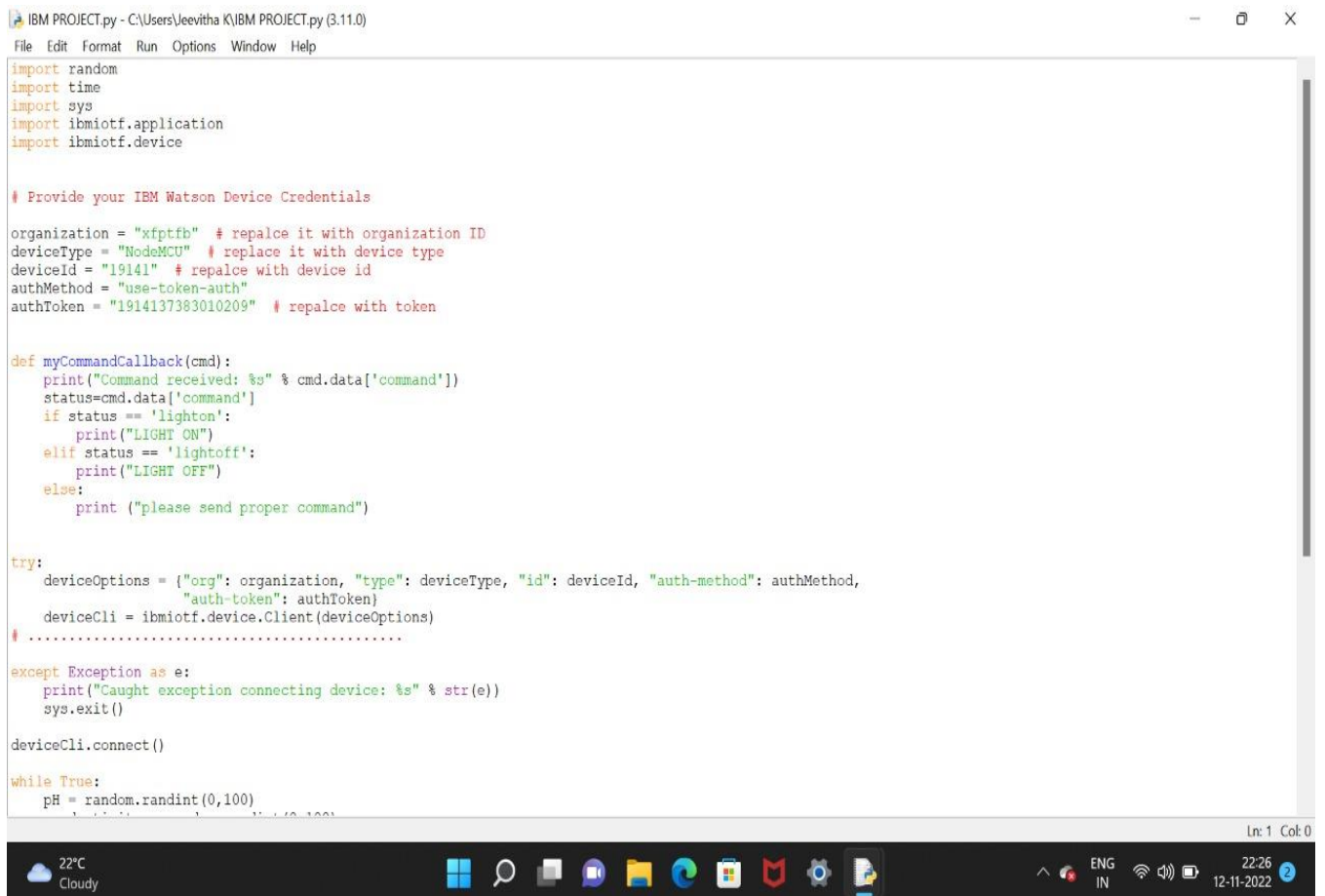
TEAM MEMBERS : VISHNU PRIYA E

ANNE SHIFANA S R

JEEVITHA K

MONISHA R

PYTHON CODE



```
IBM PROJECT.py - C:\Users\Jeevitha K\IBM PROJECT.py (3.11.0)
File Edit Format Run Options Window Help

import random
import time
import sys
import ibmiotf.application
import ibmiotf.device

# Provide your IBM Watson Device Credentials

organization = "xftfb" # replace it with organization ID
deviceType = "NodeMCU" # replace it with device type
deviceId = "19141" # replace with device id
authMethod = "use-token-auth"
authToken = "1914137383010209" # replace 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)
```

Ln: 1 Col: 0

22°C Cloudy

22:26 12-11-2022

IBM PROJECT.py - C:\Users\Jeevitha K\IBM PROJECT.py (3.11.0)

File Edit Format Run Options Window Help

```
        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 = {'temperature': 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 IoT")
        time.sleep(5)

    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
```

Ln: 1 Col: 0

22°C
Cloudy



ENG
IN 22:26
12-11-2022