

ASSIGNMENT-3

Name: Y.Sushwanth Reddy

Reg.no:19BEC0271

PYTHON CODE:

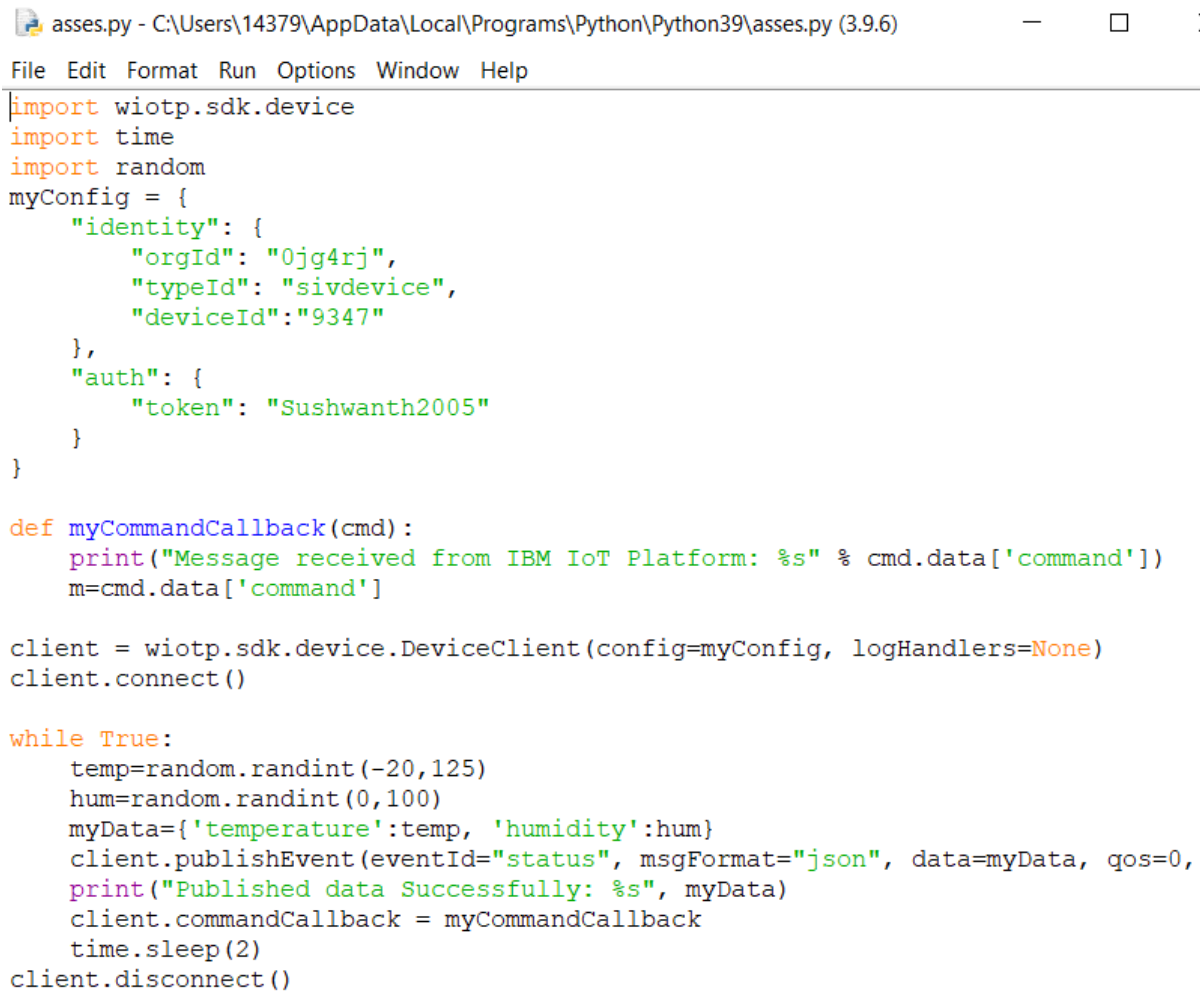
```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "0jg4rj",
        "typeId": "sivdevice",
        "deviceId": "9347"
    },
    "auth": {
        "token": "Sushwanth2005"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" %
    cmd.data['command'])
    m=cmd.data['command']
    client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
    client.connect()while True:
        water_level=random.randint(0,100)
```

```
intensity=random.randint(0,100)
myData={'water_level':water_level, 'intensity':intensity}
client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
print("Published data Successfully: %s", myData)
client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()
```

CODE:



The image shows a screenshot of a Python IDE window titled 'asses.py - C:\Users\14379\AppData\Local\Programs\Python\Python39\asses.py (3.9.6)'. The window has a menu bar with 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code is written in Python and is as follows:

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "0jg4rj",
        "typeId": "sivdevice",
        "deviceId": "9347"
    },
    "auth": {
        "token": "Sushwanth2005"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

NODE-RED DESIGN:

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes tabs for 'Node-RED Use Case, Integrating', 'Node-RED: node-red-mfczc-201', and 'Node-RED Dashboard'. The address bar shows the URL: <https://node-red-mfczc-2021-07-19.eu-gb.mybluemix.net/red/#flow/d4a87cb4.973a5>. The interface features a left sidebar with a 'filter nodes' search bar and a 'dashboard' category. The main workspace, titled 'Flow 1', contains a flow diagram. The flow starts with an 'IBM IoT' node (green) connected to two function nodes: 'temperature' and 'Humidity' (both orange). The 'temperature' function node is connected to a 'msg.payload' node (green) and a 'temperature' gauge node (blue). The 'Humidity' function node is connected to a 'humidity' gauge node (blue). The right sidebar shows a 'debug' console with a list of messages. The messages are JSON objects containing temperature and humidity data, such as:

```
{ temperature: 41, humidity: 48 }
```

 and

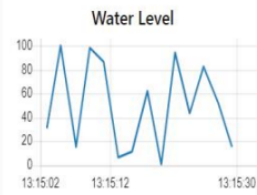
```
{ temperature: -20, humidity: 3 }
```

. The debug console also shows the timestamp and node ID for each message.

WEB-DESIGN:

≡ Smart Work

Water Level



Intensity

