SPRINT 1

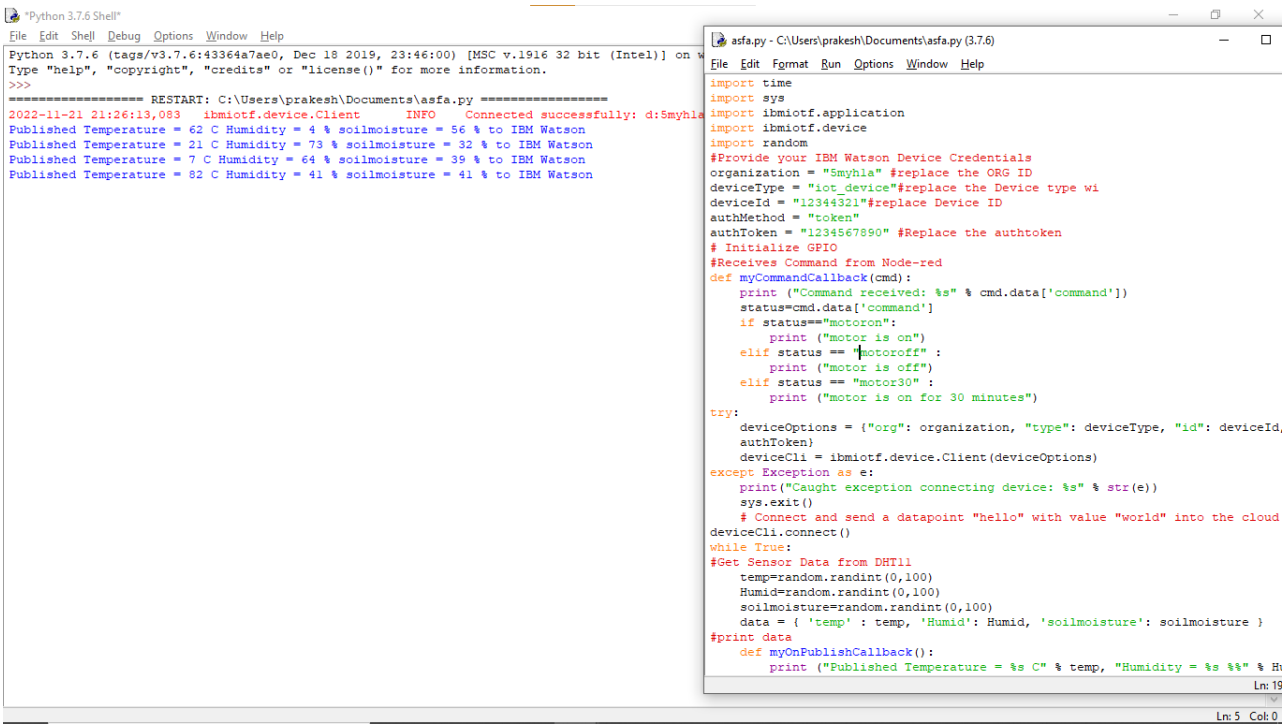
TEAM ID	PNT2022TMID41919
PROJECT NAME	Smart Farmer - IoT Enabled Smart Farming Application
Date	29 October 2022

Python code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "5myh1a" #replace the ORG ID
deviceType = "iot_device"#replace the Device type wi
deviceId = "12344321"#replace Device ID
authMethod = "token"
authToken = "1234567890" #Replace the authtoken
# Initialize GPIO
#Receives Command from Node-red
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")
    elif status == "motor30" :
        print ("motor is on for 30 minutes")
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:
    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    soilmoisture=random.randint(0,100)
    data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture':
soilmoisture }
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp,
"Humidity = %s %%" % Humid, "soilmoisture = %s
%%"%soilmoisture, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json",
data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(5)
    deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

Output:



IBM Watson: