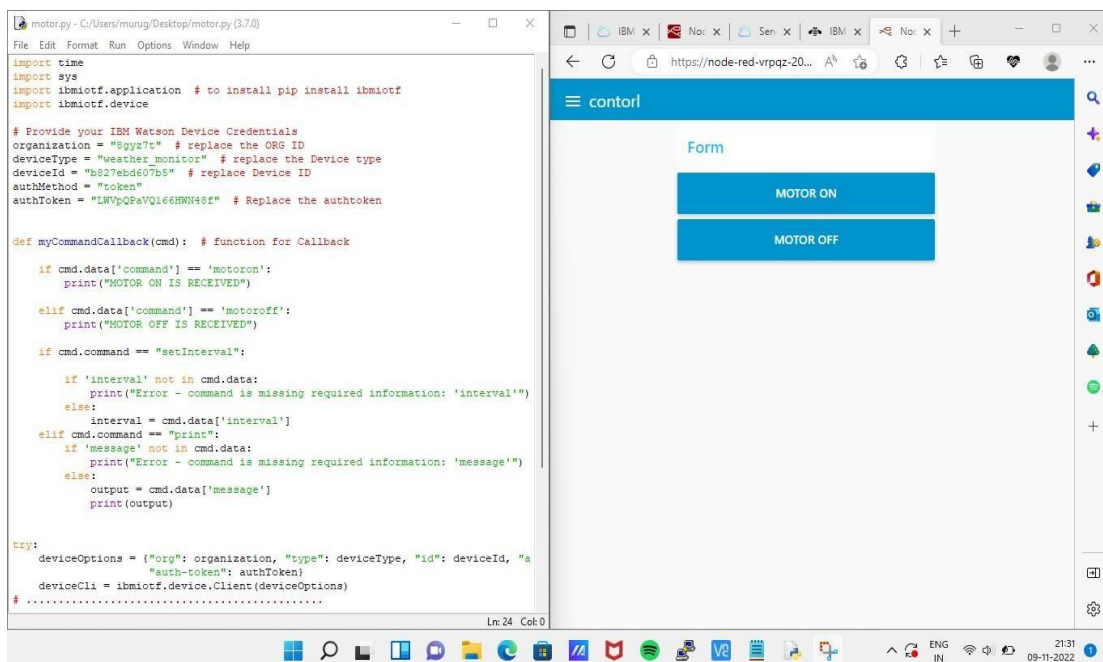


PROJECT DEVELOPMENT PHASE

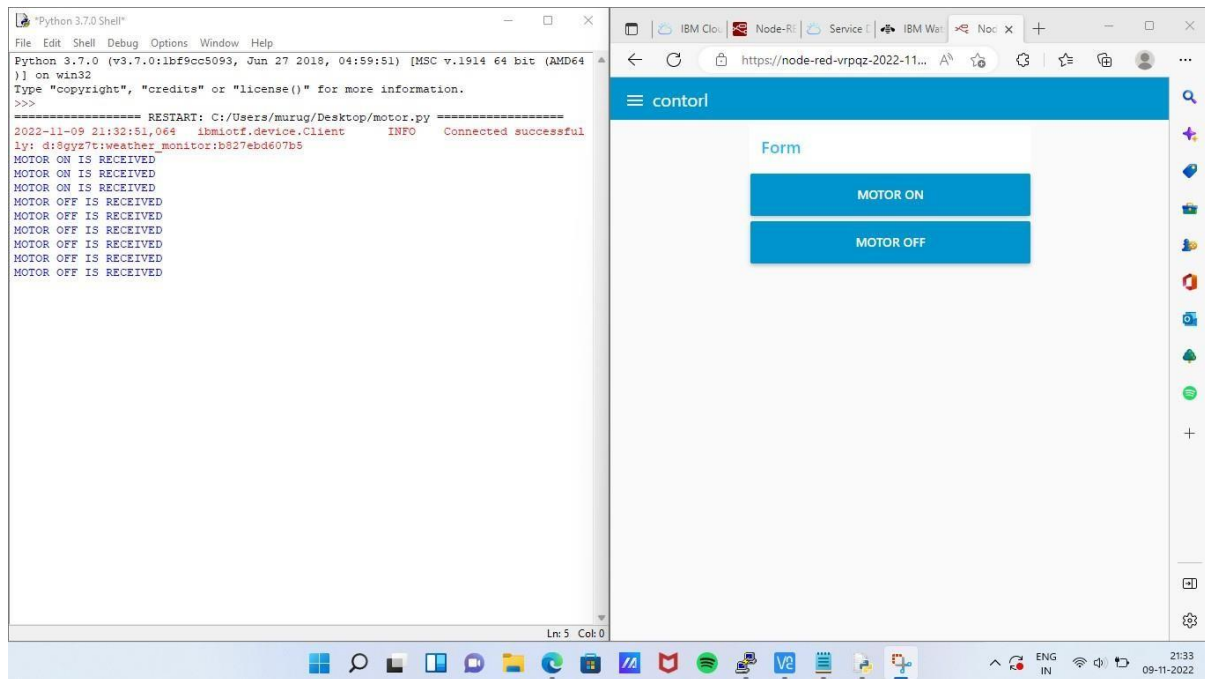
SPRINT 1

TEAM ID	PNT2022TMID44961
PROJECT NAME	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
DATE	17-11-2022

STEP 1: First open python IDLE .Then create a new file called motor.py and write a python code. Then open Node-RED flow for motor status. And open motor user interface.



STEP 2: Then run the python code then press motor on or motor off using Node-RED user interface. This shows the result in python output window.



PYTHON CODE:

```
import time
```

```
import sys
```

```
import ibmiotf.application # to install pip install ibmiotf
```

```
import ibmiotf.device
```

```
# Provide your IBM Watson Device Credentials
```

```
organization = "8gyz7t" # replace the ORG ID
```

```
deviceType = "weather_monitor" # replace the Device type
```

```
deviceId = "b827ebd607b5" # replace Device ID
```

```
authMethod = "token"
```

```
authToken = "LWVpQPpVQ166HWN48f" # Replace the auth token
```

```
def myCommandCallback(cmd): # function for Callback
```

```
    if cmd.data['command'] == 'motoron':
```

```
        print("MOTOR ON IS RECEIVED")
```

```
    elif cmd.data['command'] == 'motoroff':
```

```
        print("MOTOR OFF IS RECEIVED")
```

```
    if cmd.command == "setInterval":
```

```
        if 'interval' not in cmd.data:
```

```
            print("Error - command is missing required information: 'interval'")
```

```
        else:
```

```
            interval = cmd.data['interval']
```

```
    elif cmd.command == "print":
```

```
        if 'message' not in cmd.data:
```

```
            print("Error - command is missing required information: 'message'")
```

```
        else:
```

```
            output = cmd.data['message']
```

```
            print(output)
```

```
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:
```

```
    deviceCli.commandCallback = myCommandCallback
```

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

```
deviceCli.disconnect()
```

Node-RED:

```
[{"id":"625574ead9839b34","type":"ibmiot
out","z":"630c8601c5ac3295","authentication":"apiKey","apiKey":"ef74
5d48e395ccc0","outputType":"cmd","deviceId":"b827ebd607b5","device
Type":"weather_monitor","eventCommandType":"data","format":"json
","data":"data","qos":0,"name":"IBM
IoT","service":"registered","x":680,"y":220,"wires":[]},{ "id":"4cff18c32
74cccc4","type":"ui_button","z":"630c8601c5ac3295","name":"","group
":"716e956.00eed6c","order":2,"width":"0","height":"0","passthru":fal
se,"label":"Motor
ON","tooltip":"","color":"","bgcolor":"","className":"","icon":"","p
ayload":"{\"command\":\"motoron\"}","payloadType":"str","topic":"m
otoron","topicType":"str","x":360,"y":160,"wires":[["625574ead9839b3
4"]]}, {"id":"659589baceb4e0b0","type":"ui_button","z":"630c8601c5ac3
295","name":"","group":"716e956.00eed6c","order":3,"width":"0","hei
ght":"0","passthru":true,"label":"Motor
OFF","tooltip":"","color":"","bgcolor":"","className":"","icon":"","
payload":"{\"command\":\"motoroff\"}","payloadType":"str","topic":"
motoroff","topicType":"str","x":350,"y":220,"wires":[["625574ead9839b
34"]]}, {"id":"ef745d48e395ccc0","type":"ibmiot","name":"weather_mon
itor","keepalive":"60","serverName":"","cleansession":true,"appId":"","
"shared":false}, {"id":"716e956.00eed6c","type":"ui_group","name":"Fo
rm","tab":"7e62365e.b7e6b8","order":1,"disp":true,"width":"6","collap
se":false}, {"id":"7e62365e.b7e6b8","type":"ui_tab","name":"contorl","i
con":"dashboard","order":1,"disabled":false,"hidden":false}]
```