Sprint 3

Date	12 November 2022
Team ID	PNT2022TMID21337
Project Name	Project – Smart Farmer-IoT Enabled Smart Farming Application
Maximum Marks	2 Marks

Code:

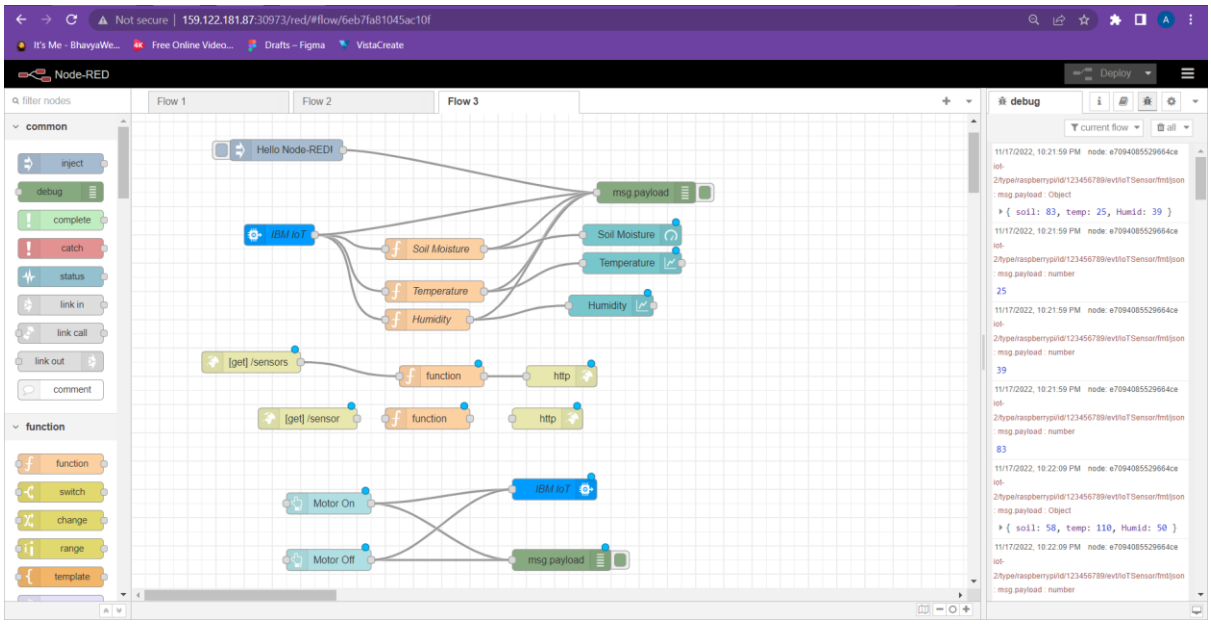
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "obbnyv"
deviceType = "raspberrypi"
deviceId = "123456789"
authMethod = "token"
authToken = "12345678910"
# Initialize GPIO
def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="lighton":
      print ("led is on")
   elif status == "lightoff":
      print ("led is off")
   elif 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 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:
    #Get Sensor Data from DHT11
    sm=random.randint(0,110)
    temp=random.randint(-20,125)
    Humid=random.randint(0,100)
    data = {'soil': sm, 'temp' : temp, 'Humid': Humid }
    #print data
    def myOnPublishCallback():
      print ("Published Soil Moisture = %s %%" %sm,"Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "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
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

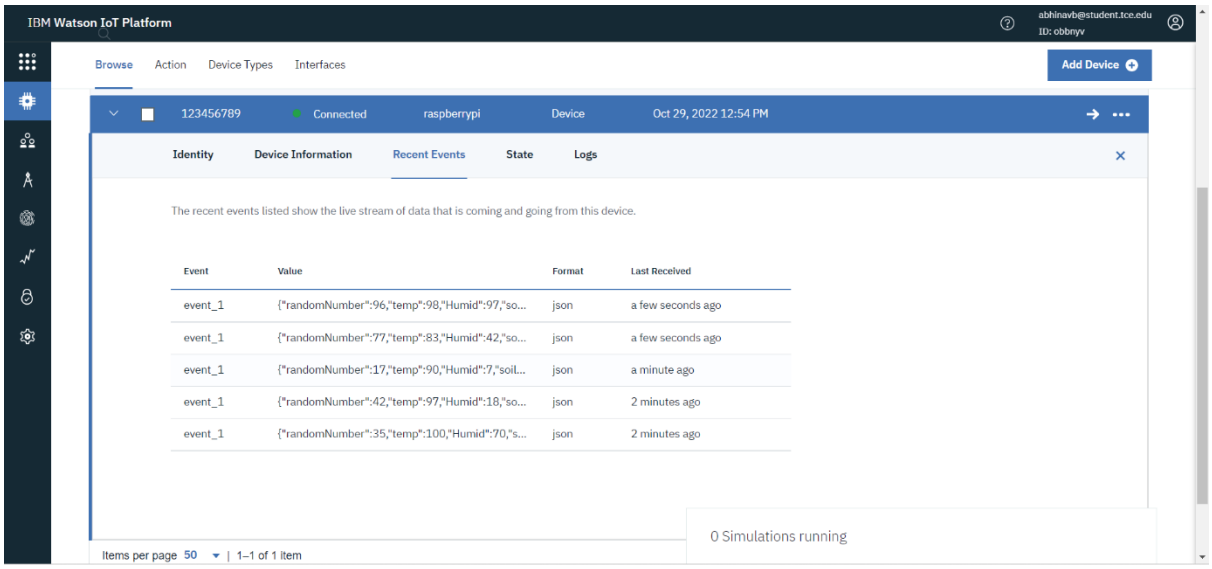
```
2022-11-17 22:21:29,167   ibmiotf.device.Client      INFO    Connected successfully:
d:obbnyv:raspberrypi:123456789
Published Soil Moisture = 104 % Temperature = 102 C Humidity = 87 % to IBM Watson
Published Soil Moisture = 88 % Temperature = 18 C Humidity = 96 % to IBM Watson
Published Soil Moisture = 52 % Temperature = 32 C Humidity = 63 % to IBM Watson
Published Soil Moisture = 83 % Temperature = 25 C Humidity = 39 % to IBM Watson
Published Soil Moisture = 58 % Temperature = 110 C Humidity = 50 % to IBM Watson
Published Soil Moisture = 2 % Temperature = 42 C Humidity = 85 % to IBM Watson
Published Soil Moisture = 34 % Temperature = 33 C Humidity = 56 % to IBM Watson
Published Soil Moisture = 44 % Temperature = 68 C Humidity = 90 % to IBM Watson
Published Soil Moisture = 58 % Temperature = 28 C Humidity = 2 % to IBM Watson
Command received: motoron
motor is on
Command received: motoron
motor is on
Published Soil Moisture = 37 % Temperature = 108 C Humidity = 34 % to IBM Watson
Published Soil Moisture = 104 % Temperature = -4 C Humidity = 0 % to IBM Watson
Traceback (most recent call last):
```

Configuration of Node-Red to send commands to IBM cloud

ibmiot out node I used to send data from Node-Red to IBM Watson device. So, after adding it to the flow we need to configure it with credentials of our Watson device.



IBM Watson:



Get/Sensors:


```
{"Soil":31,"Temperature":25,"Humid":73}
```