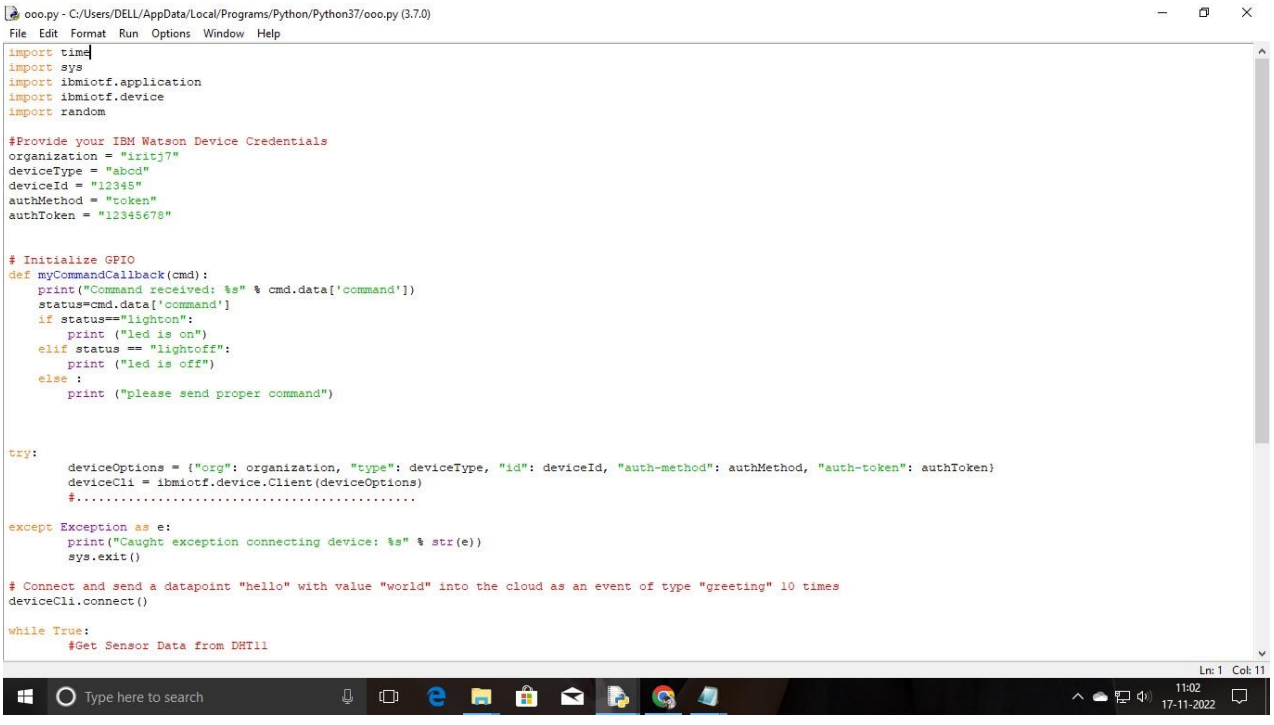
PROJECT DEVELOPMENT PHASE

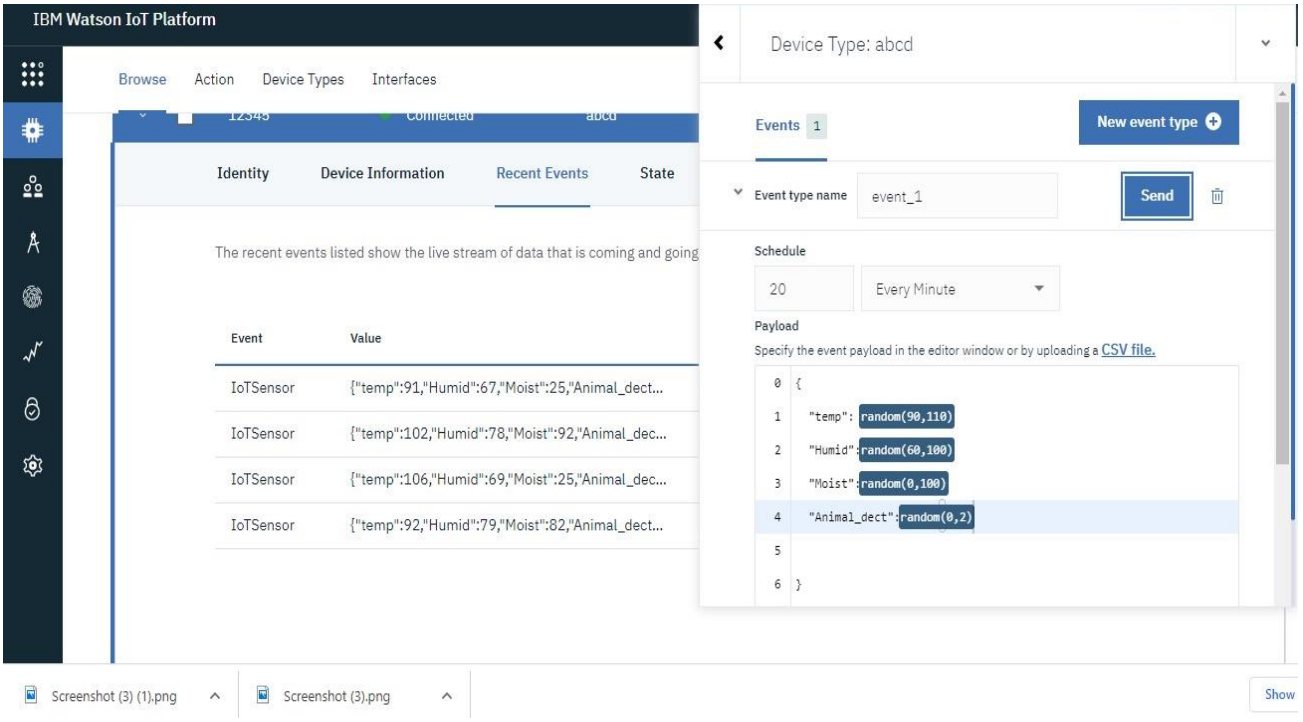
SPRINT 3

TEAM ID	**PNT2022TMID32078**
PROJECT NAME	**IOT BASEDSMART CROPPROTECTION SYSTEM FOR AGRICULTURE**
DATE	**31 OCTOBER 2022**

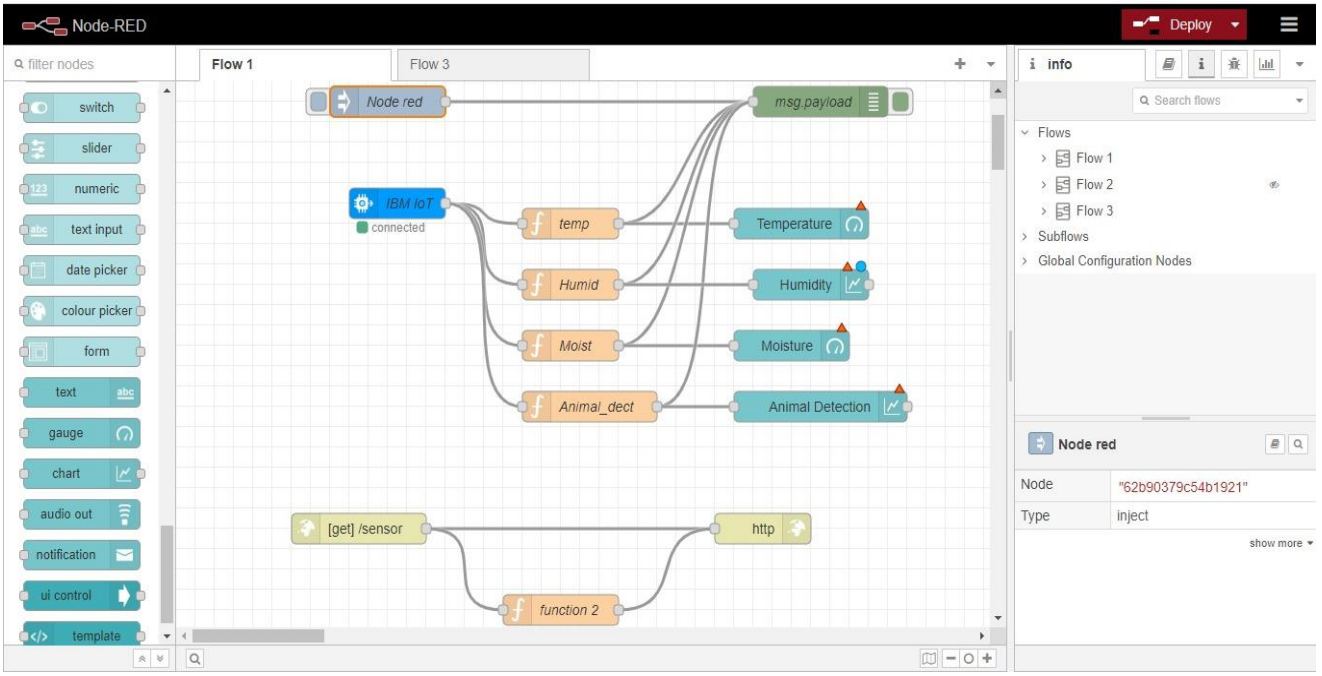
STEP 1: Write a python code for randomize Soil Moisture ,Temperature, Humidity and Animal detection.



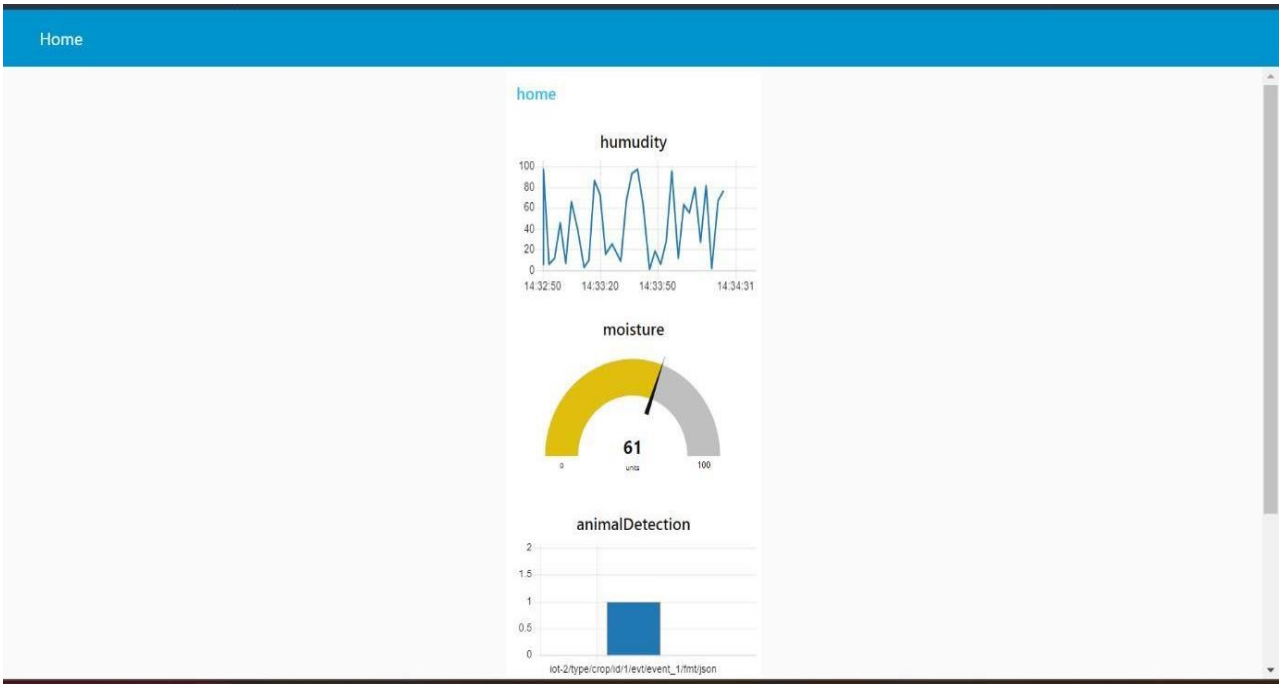
STEP 2: Run the python code it send data to IBM IoT Watson Platform.



STEP 3: Open Node-RED flow dashboard.



STEP 4: Open Node-RED user interface to show the Soil Moisture, Humidity and Temperature value in gauge.



PYTHON CODE :

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

#Provide your IBM Watson Device Credentials
organization = "iritj7"
deviceType = "abcd" deviceId
= "12345" authMethod =
"token"
authToken = "12345678"


# 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")
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

      temp=random.randint(90,110)
Humid=random.randint(60,100)
      Moist=random.randint(20,100)
      Animal_dect=random.randint(1,20)



      data = { 'temp' : temp, 'Humid': Humid, 'Moist' : Moist, 'Animal_dect' :
Animal_dect }
      #print data        def
myOnPublishCallback():
          print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid, "to IBM Watson", "Published Moisture= %s" % Moist, "Published Animal
detection = " , Animal_dect)

      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()
```