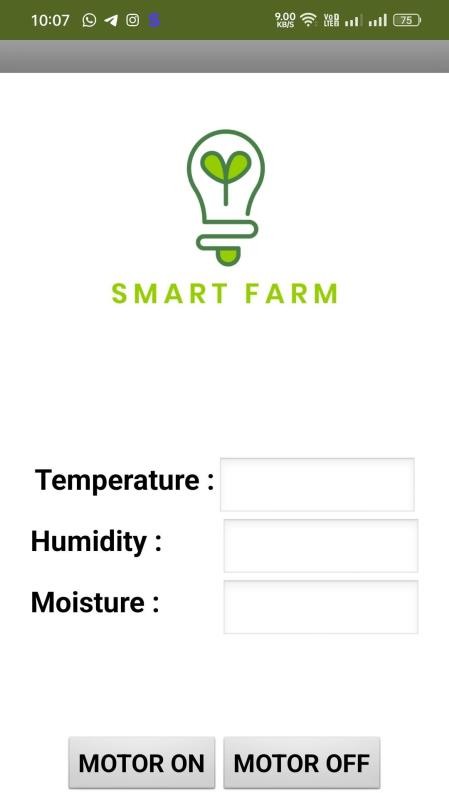
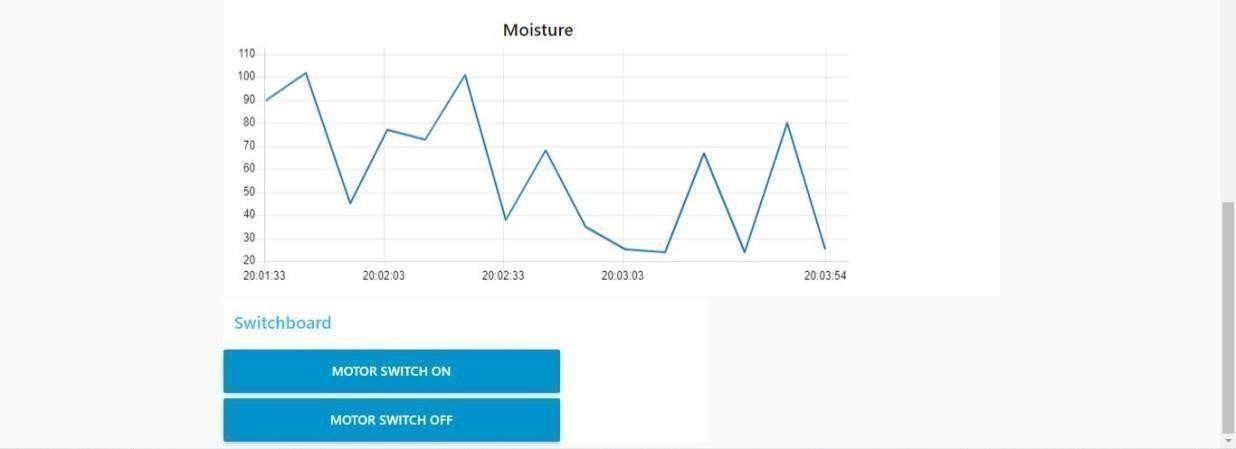
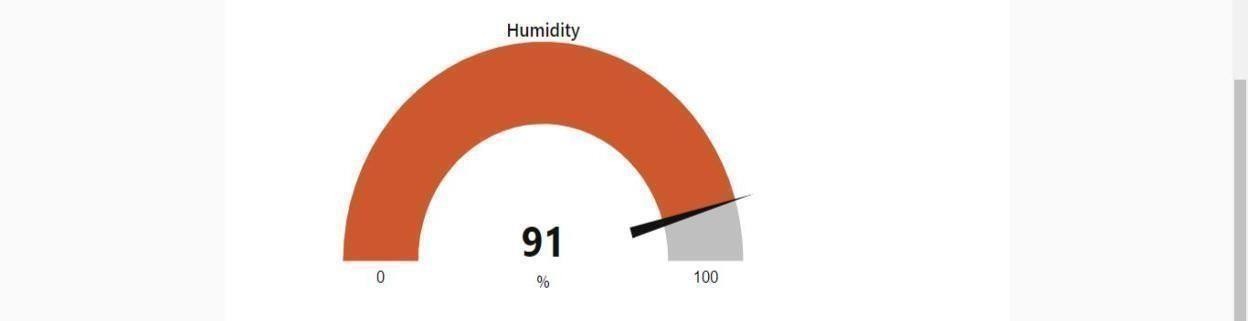
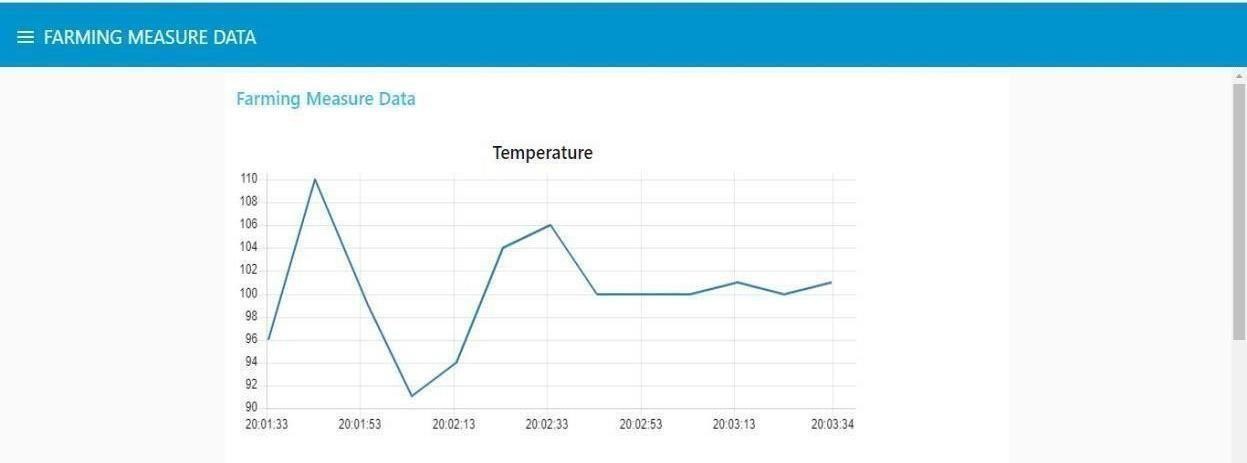
**SMART FARMER APPLICATION USER REGISTRATION WINDOW**

  
 **FUNCTION WINDOW**

## **Web APP UI Home Tab**



**CODING AND SOLUTIONING**:

PYTHON CODE :

import time

import sys

import ibmiotf.application # to install pip install ibmiotf

import ibmiotf.device

#Provide your IBM Watson Device Credentials

organization = "0lsrz8" # repalce it with organization ID

deviceType = "SMART FARMER #replace it with device type

deviceId = "Device\_1" #repalce with device id

authMethod = "token"

authToken = "SMARTFarmer@123"#repalce with token

def myCommandCallback(cmd): # function for Callback

print("Command received: %s" % cmd.data)

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

print("Turn Motor ON")

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

print("Turn Motor OFF")

elif cmd.data['command']=='lighton':

print("Turn Light ON")

elif cmd.data['command']=='lightoff':

print("Turn Light OFF")

if cmd.data['command']=='ACTIVATE IRRIGATION':

print("TurnON")

elif cmd.data['command']=='DEACTIVATE IRRIGATION':

print("TurnOFF")

elif cmd.data['command']=='HIGH TEMPERATURE':

print("TurnON")

elif cmd.data['command']=='LOW TEMPERATURE':

print("TurnOFF")

if cmd.data['command']=='BAD WEATHER':

print("TurnON")

elif cmd.data['command']=='GOOD WEATHER':

print("TurnOFF")

elif cmd.data['command']=='HUMIDITY HIGH':

print("TurnON")

elif cmd.data['command']=='HUMIDITY LOW':

print("TurnOFF")

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

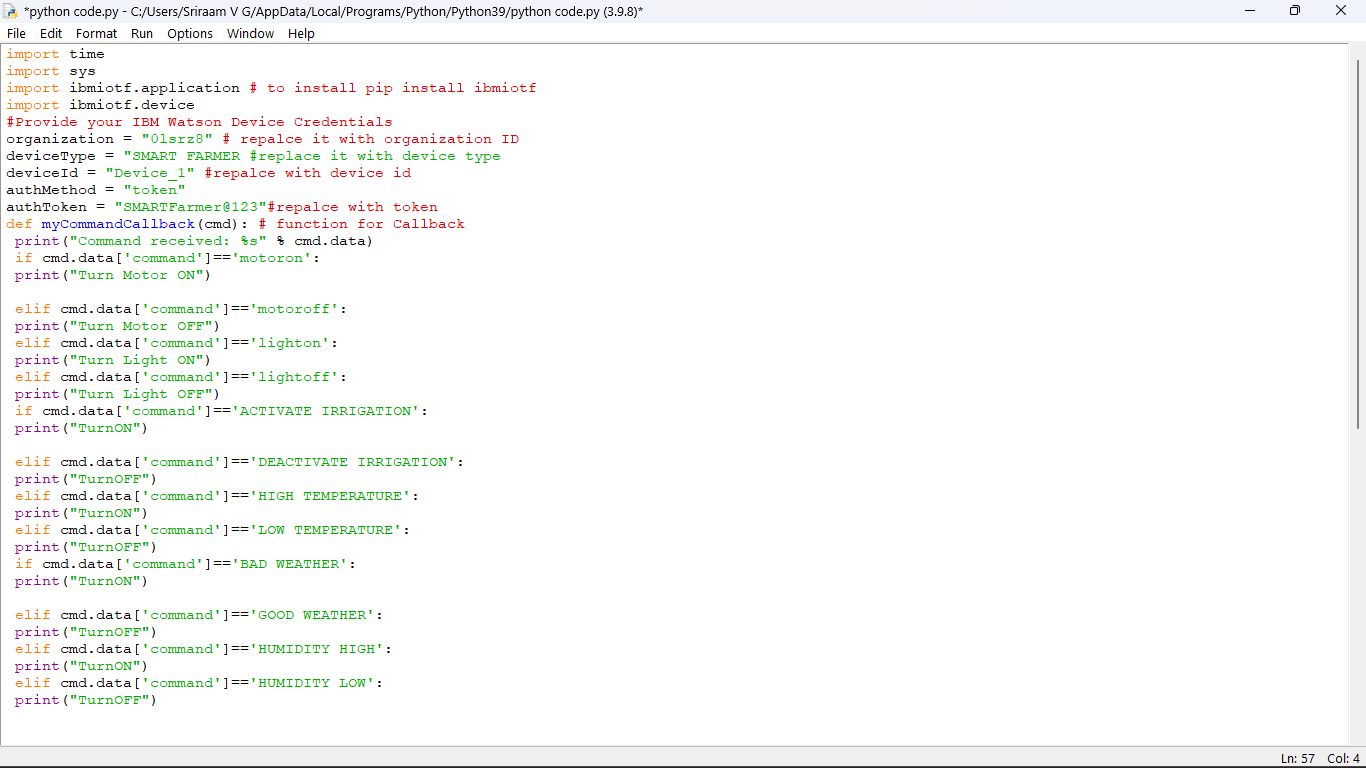
deviceCli.connect()

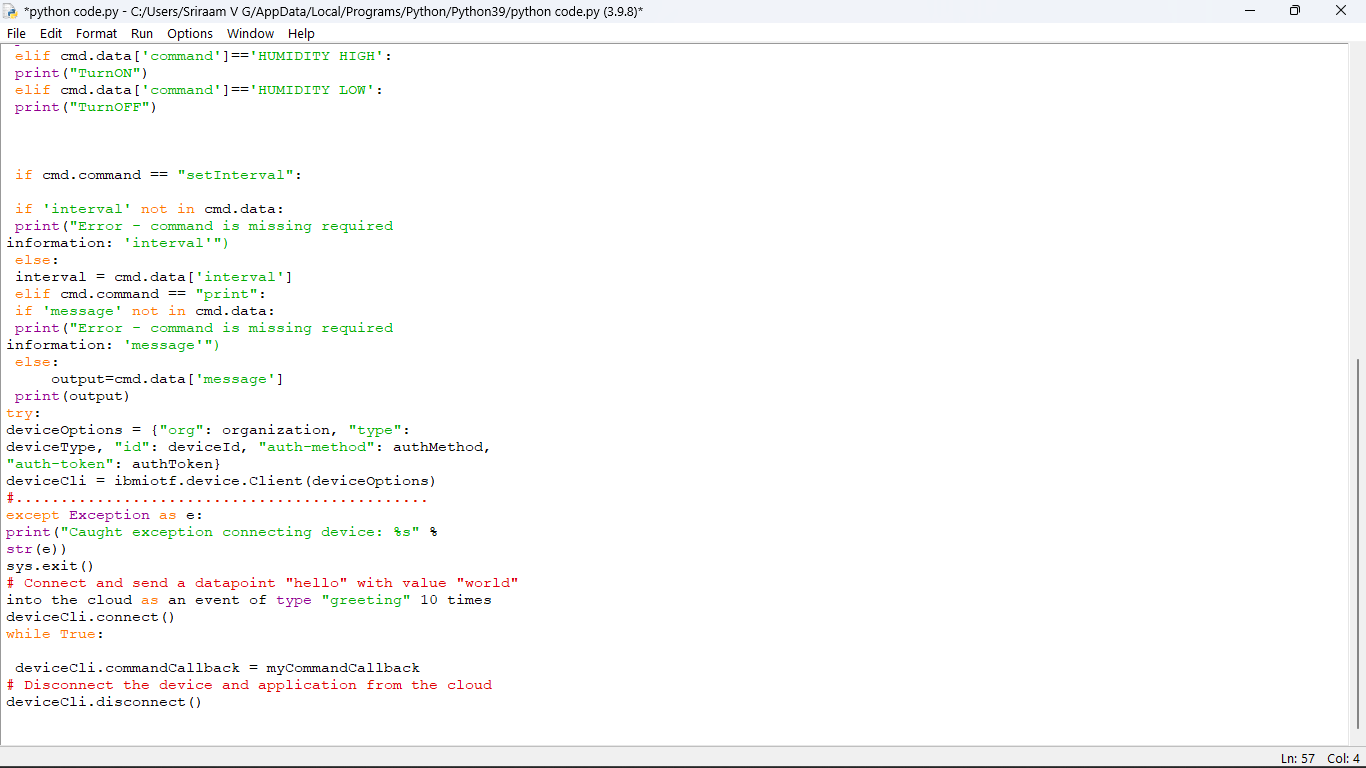
while True:

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()





PYTHON CODE FOR TEMPERATURE:

from random import \*

from random import \*

while True:

temperature = randrange(0,100)

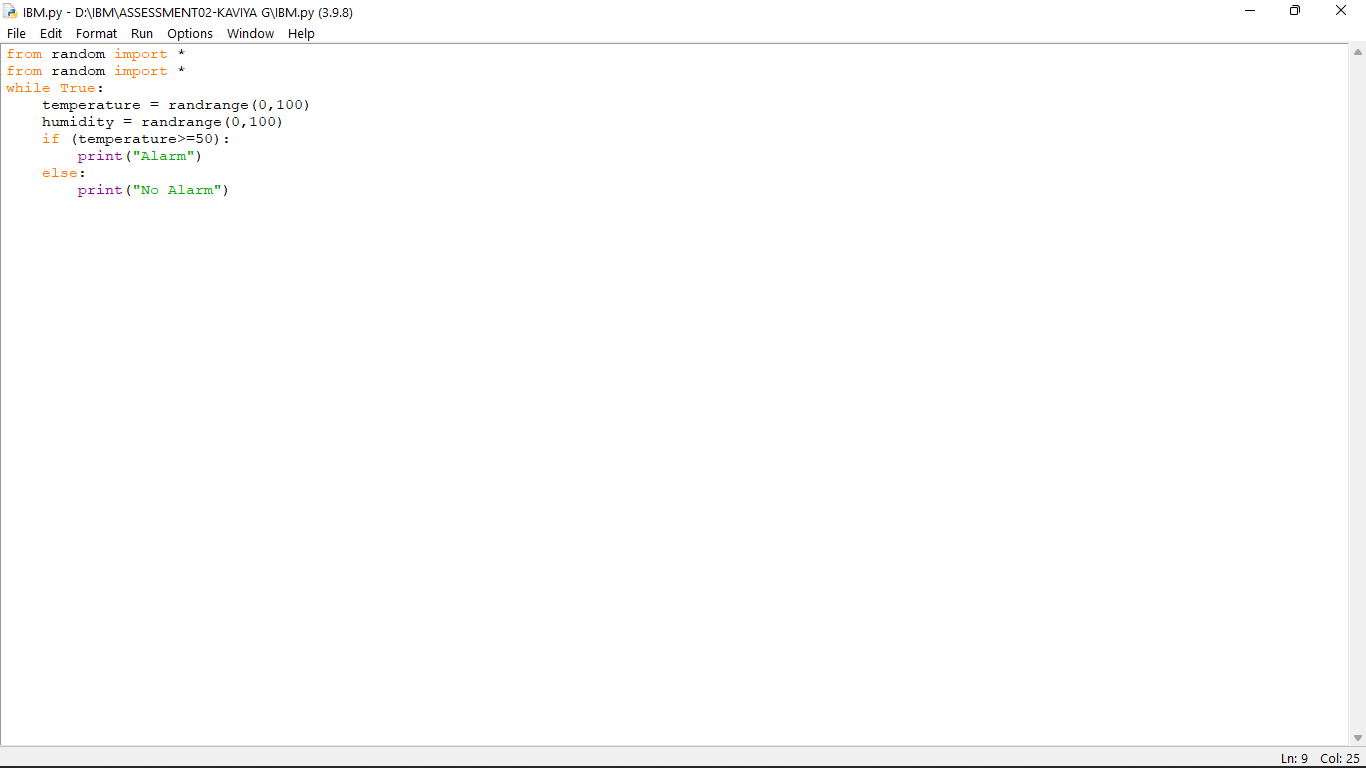
humidity = randrange(0,100)

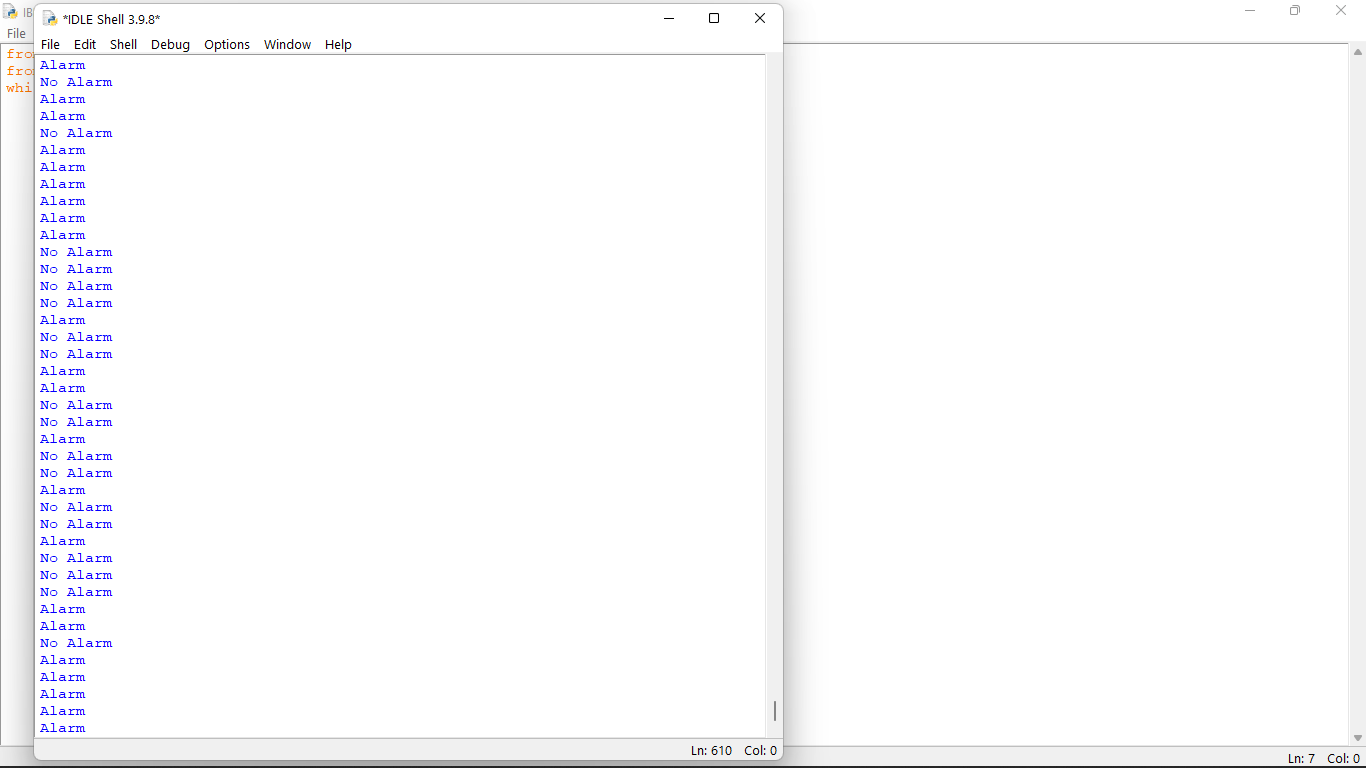
if (temperature>=50):

print("Alarm")

else:

print("No Alarm")





PYTHON CODE :

import random as rand

print("WELCOME SMART FARMER")

temperature = float(rand.uniform(15,50))

if(temperature>22 and temperature<40):

humidity = int(rand.randint(45,65))

elif(temperature<22):

humidity = int(rand.randint(60,70))

elif(temperature>40):

humidity = int(rand.randint(25,35))

moisture = int(rand.randint(00,70))

print("temperature:",temperature,"C","\n","humidity:",humidity,"\n","moisture:",moisture)

if(temperature>35 or moisture<20 ):

print("Irrigation required")

print("Activate irrigation ?")

decision = input()

if(decision == 'yes'):

print("Irrigation activated")

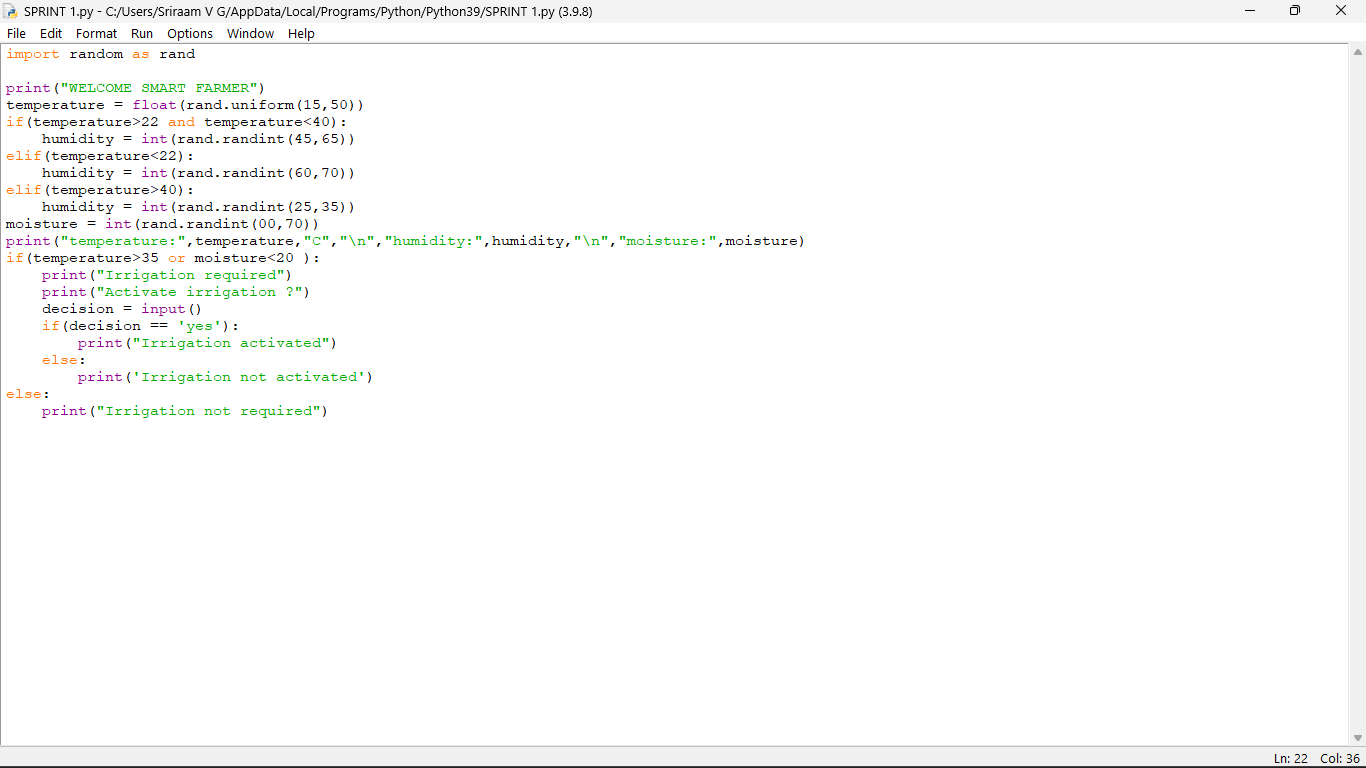
else:

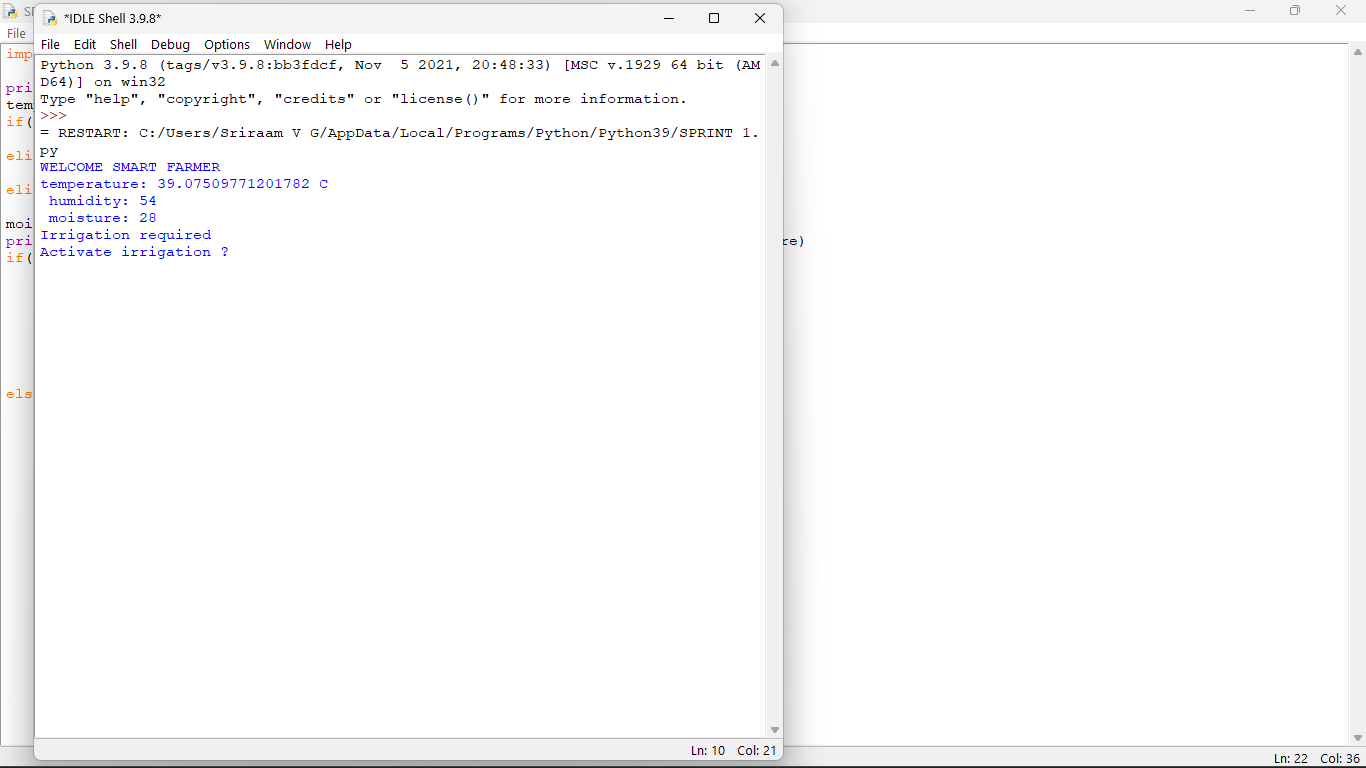
print('Irrigation not activated')

else:

print("Irrigation not required")

OUTPUT:





#Receiving commands from IBM cloud using Python program

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "157uf3"

deviceType = "abcd"

deviceId= "7654321"

authMethod = "token"

authToken = "87654321"

# Initialize GPIO

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")

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)

Mois=random. Randint(20,120)

data = { 'temp' : temp, 'Humid': Humid , ‘Mois’:

Mois}

#print data def myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%"

% Humid, “Moisture =%s deg c” % Mois “to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

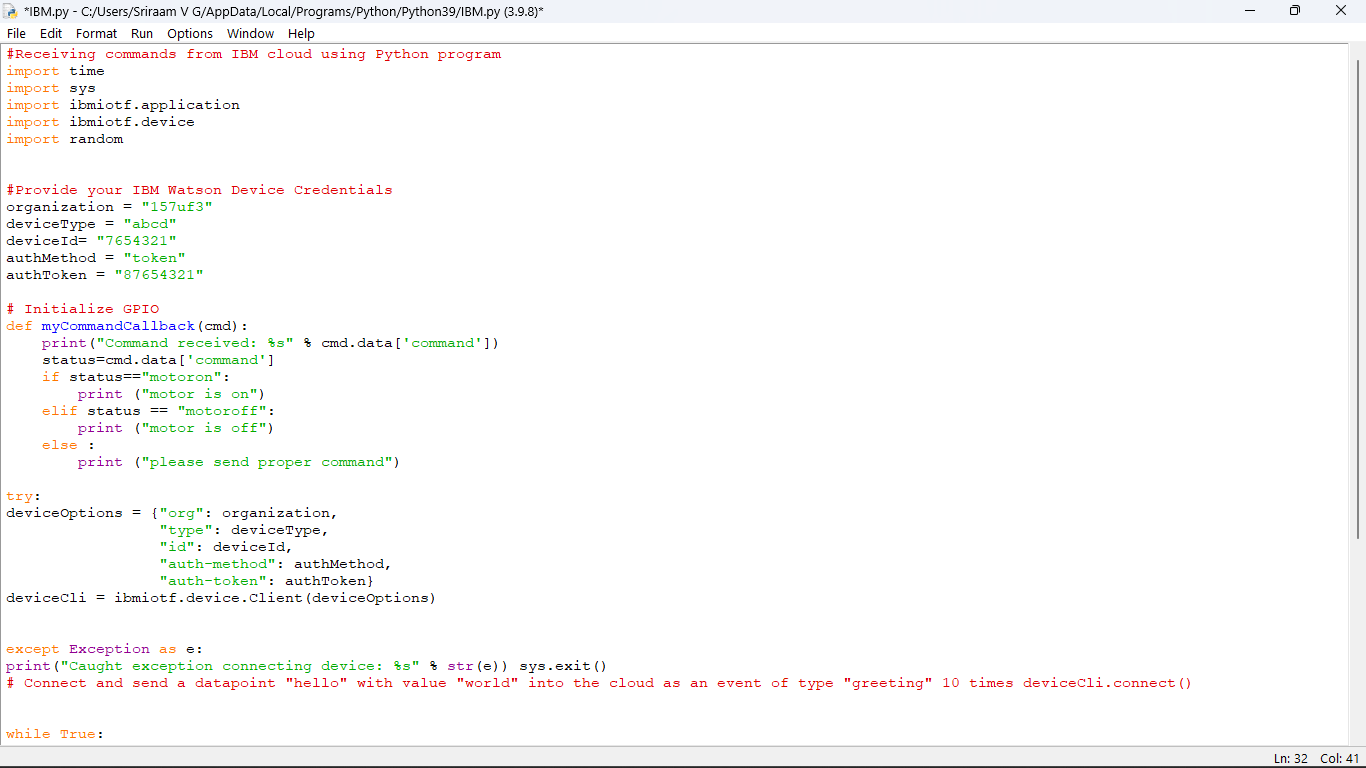
if not success:

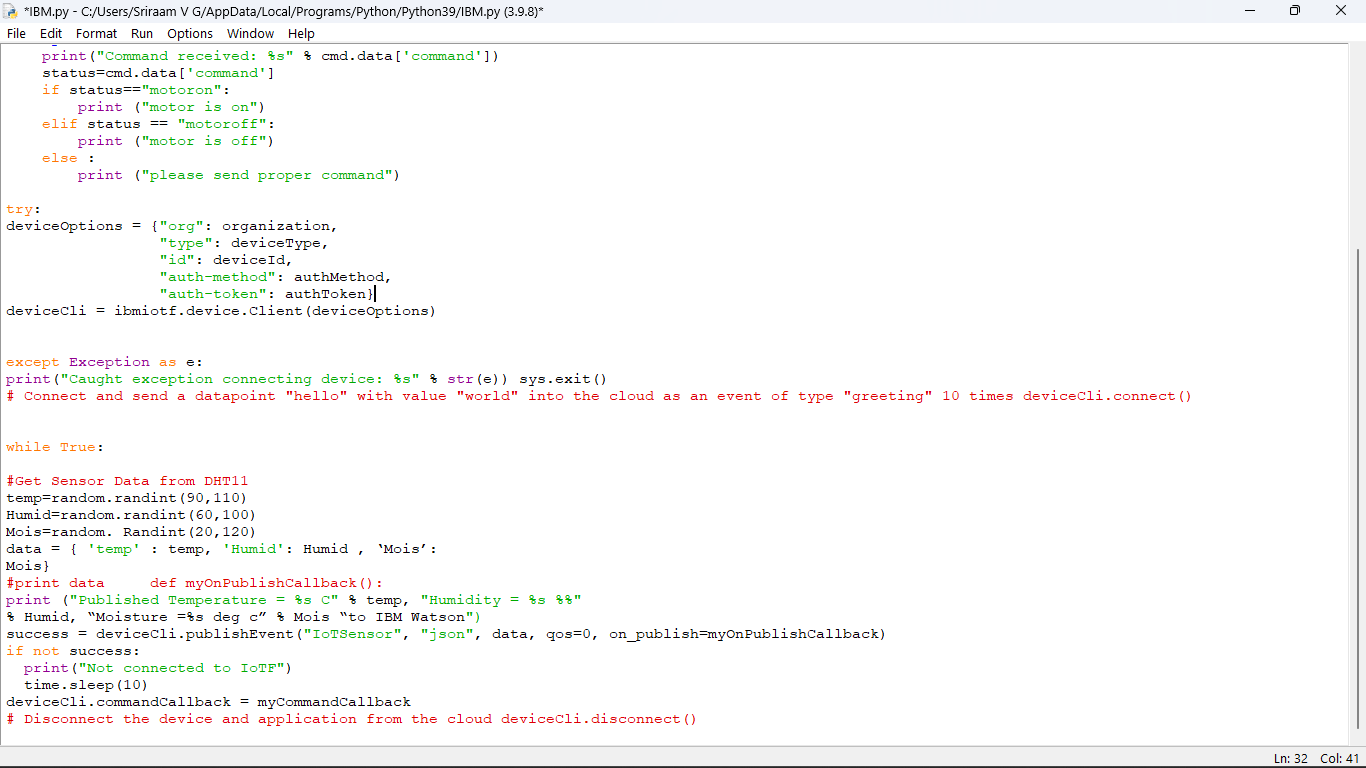
print("Not connected to IoTF")

time.sleep(10)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud deviceCli.disconnect()





WEB LINK: <http://node-red-ixvyk-2022-11-10.eu-gb.mybluemix.net/ui>

APP LINK:

