Smart Farmer-IOT Enabled Smart Farming Application

SPRINT DELIVERY-3

TEAM ID: PTN2022TMID02589

TEAM MEMBERS

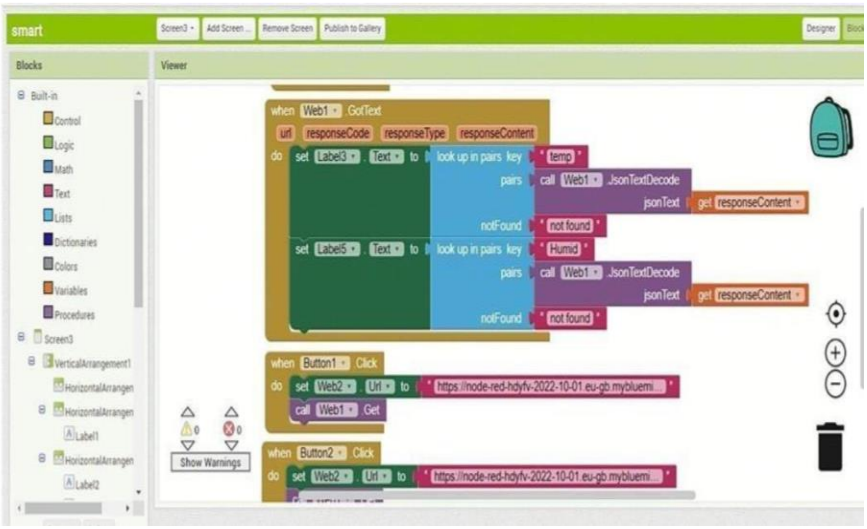
KESHIKA B

ABINAYA E

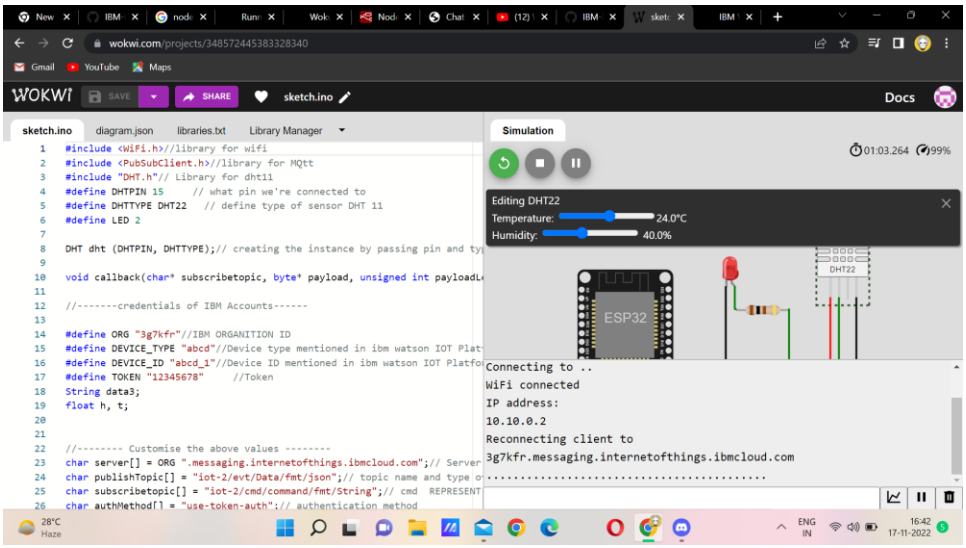
KAAVYA P

ARUNA K

CREATING MOBILE APP USING MIT APP INVENTOR



CONNECTING SKETCH WOWKI AND IBM IOT WATSON PLATFORM

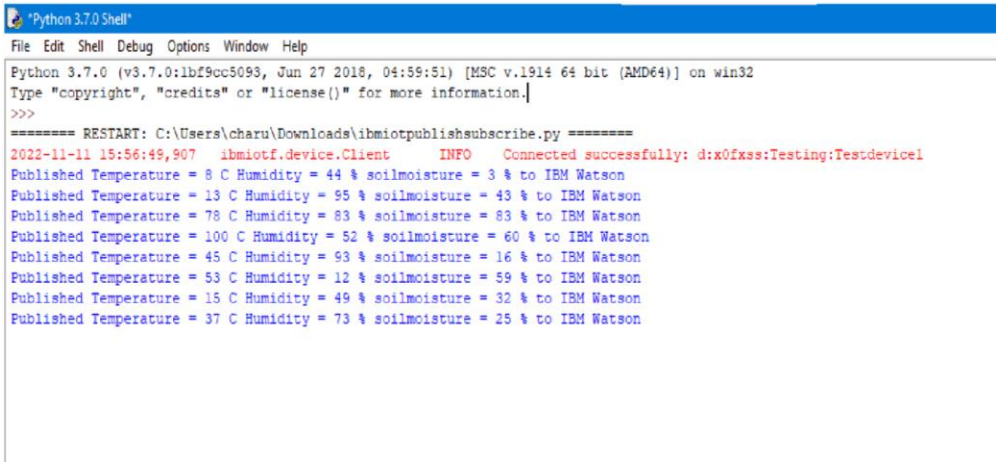


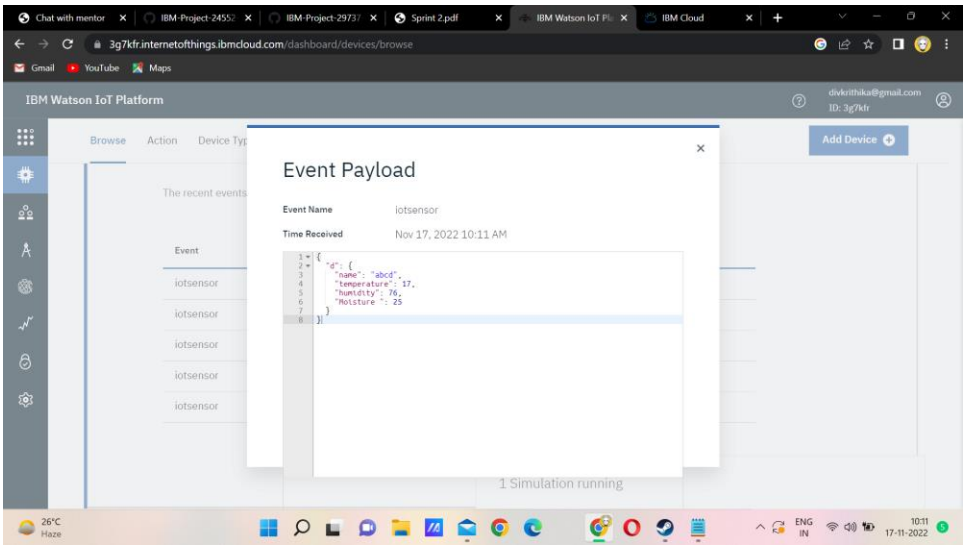
PYTHON SCRIPT AND IBM IOT WATSON IS EXECUTED

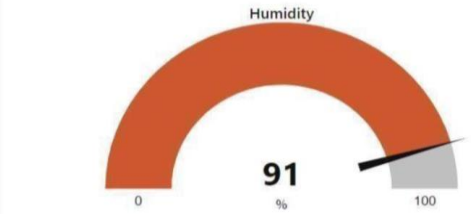
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "x0fxss" #replace the ORG ID
deviceType = "Testing"#replace the Device type wi
deviceId = "Testdevice1"#replace Device ID
authMethod = "token"
authToken = "123456789" #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)
```

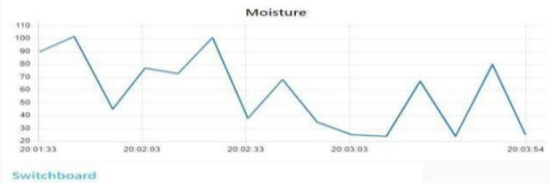
```
xcept Exception as e:
   print("Caught exception connecting device: %s" % str(e))
   sys.exit()

 Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
eviceCli.connect()

hile True:
       #Get Sensor Data from DHT11

       temp=random.randint(0,100)
       Humid=random.randint(0,100)
       soilmoisture=random.randint(0,100)

       data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture': soilmoisture }
       #print data
       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

 Disconnect the device and application from the cloud
eviceCli.disconnect()
```

RUNNING PYTHON SCRIPT TO SEND DATA TO IBM AND DATA IS DISPLAYED USING NODE RERD MODULES









Switchboard