Date	16 November 2022
Team ID	PNT2022TMID21337
Project Name	Project – Smart Farmer-IoT Enabled Smart Farming Application

CONFIGURATION OF NODE-RED TO SEND COMMANDS TO IBM CLOUD

Here we add two buttons in UI
1 -> for motor on

2 -> for motor off

We used a function node to analyses the data received and assign command to each number.

The python code for the analyses is:

```
if status == "motoron":
    print ("motor is on")
elif status == "motoroff":
    print ("motor is off")
else :
    print ("please send proper command")
```

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"
```

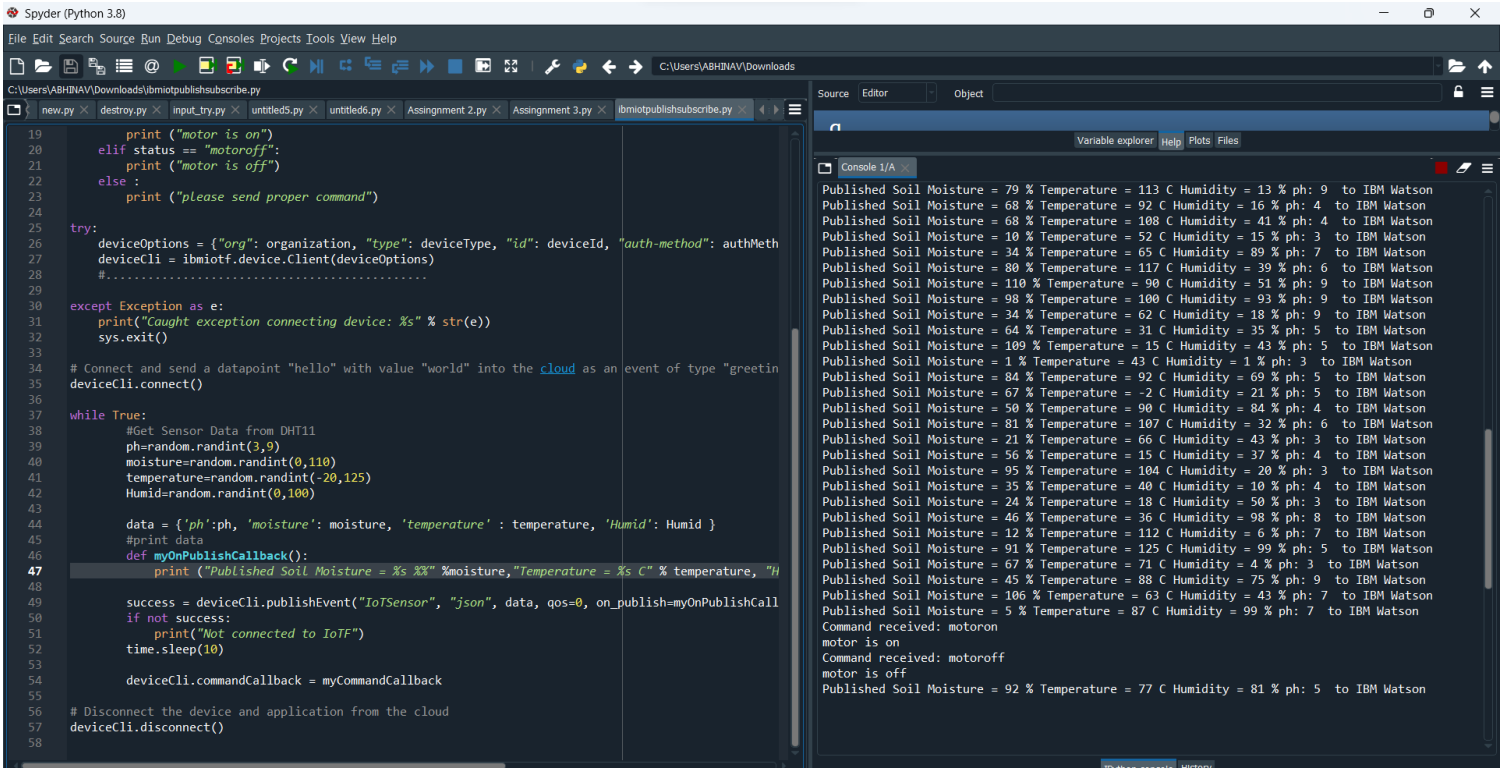
```
authMethod = "token"
authToken = "12345678910"


# Initialize GPIO
def myCommandCallback(cmd):
   print("Command received: %s" %
cmd.data['command'])
   status=cmd.data['command']
   if status == "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
        ph=random.randint(3,9)
        moisture=random.randint(0,110)
        temperature=random.randint(-
20,125)
        Humid=random.randint(0,100)


        data = {'ph':ph, 'moisture':
moisture, 'temperature' : temperature,
'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Soil Moisture
= %s %%" %moisture,"Temperature =
%s C" % temperature, "Humidity = %s
%%" % Humid, "ph: %s "%ph, "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()

Output:

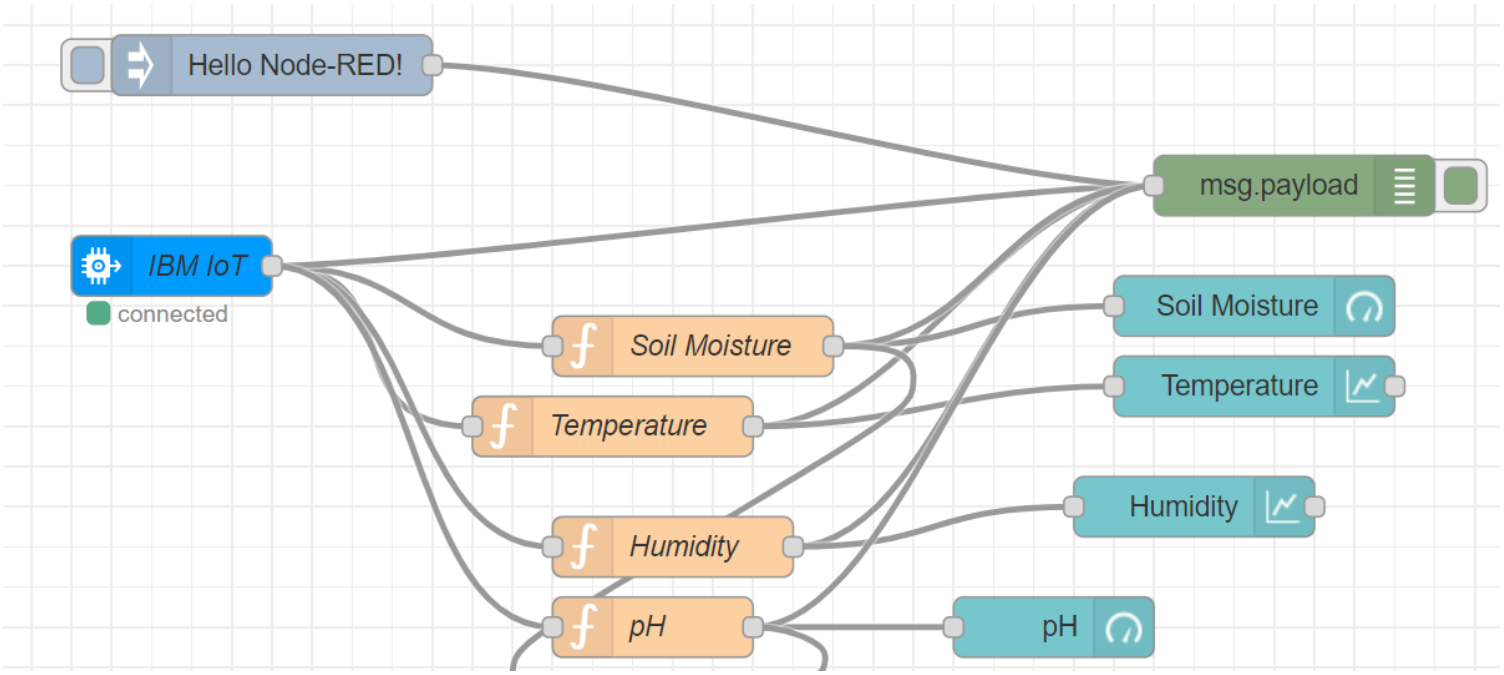


Adjusting User Interface

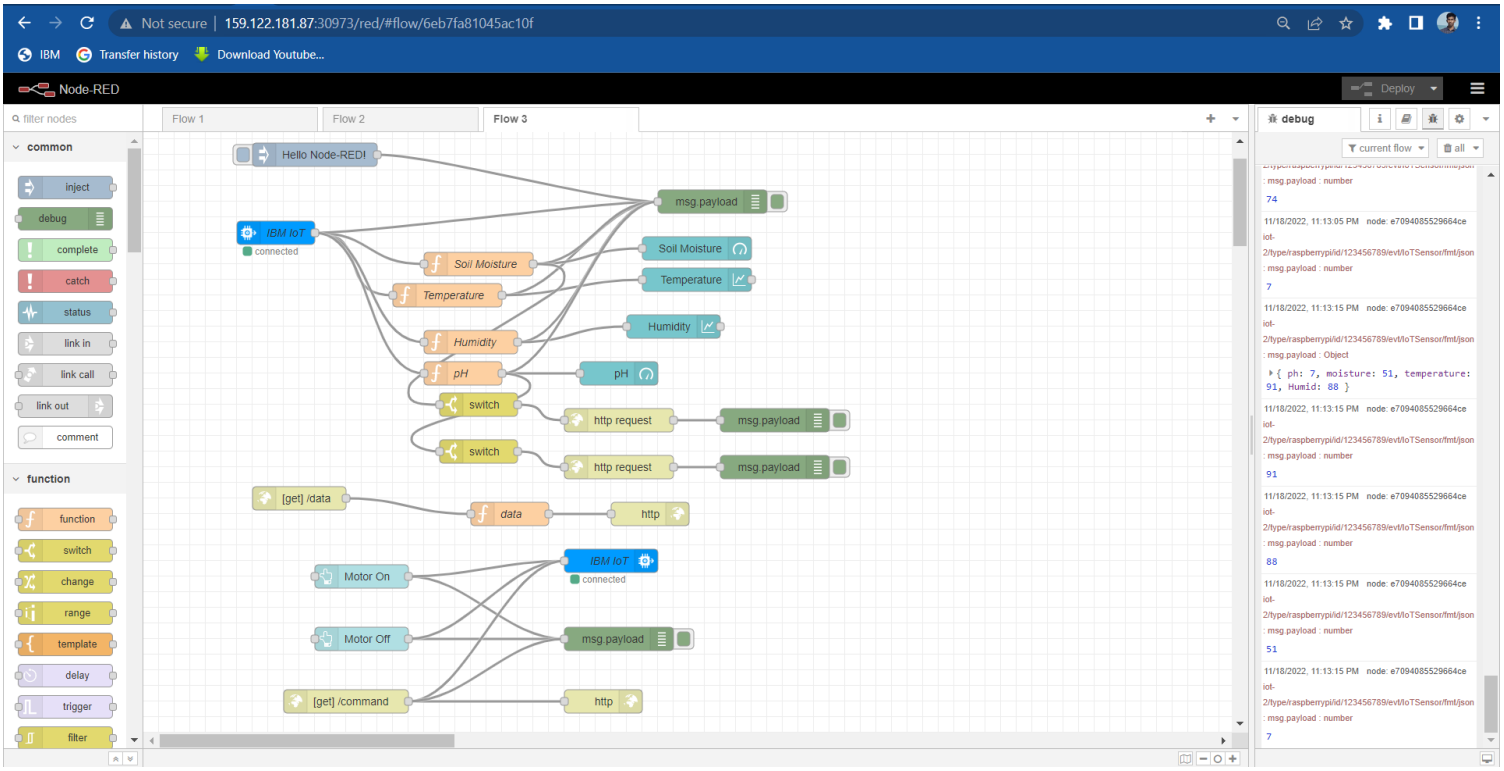
In order to display the parsed JSON data a Node-Red dashboard is created.

Here we are using Gauges, text and button nodes to display in the UI and helps tomonitor the parameters and control the farm equipment.

Below images we started to create the flow 1



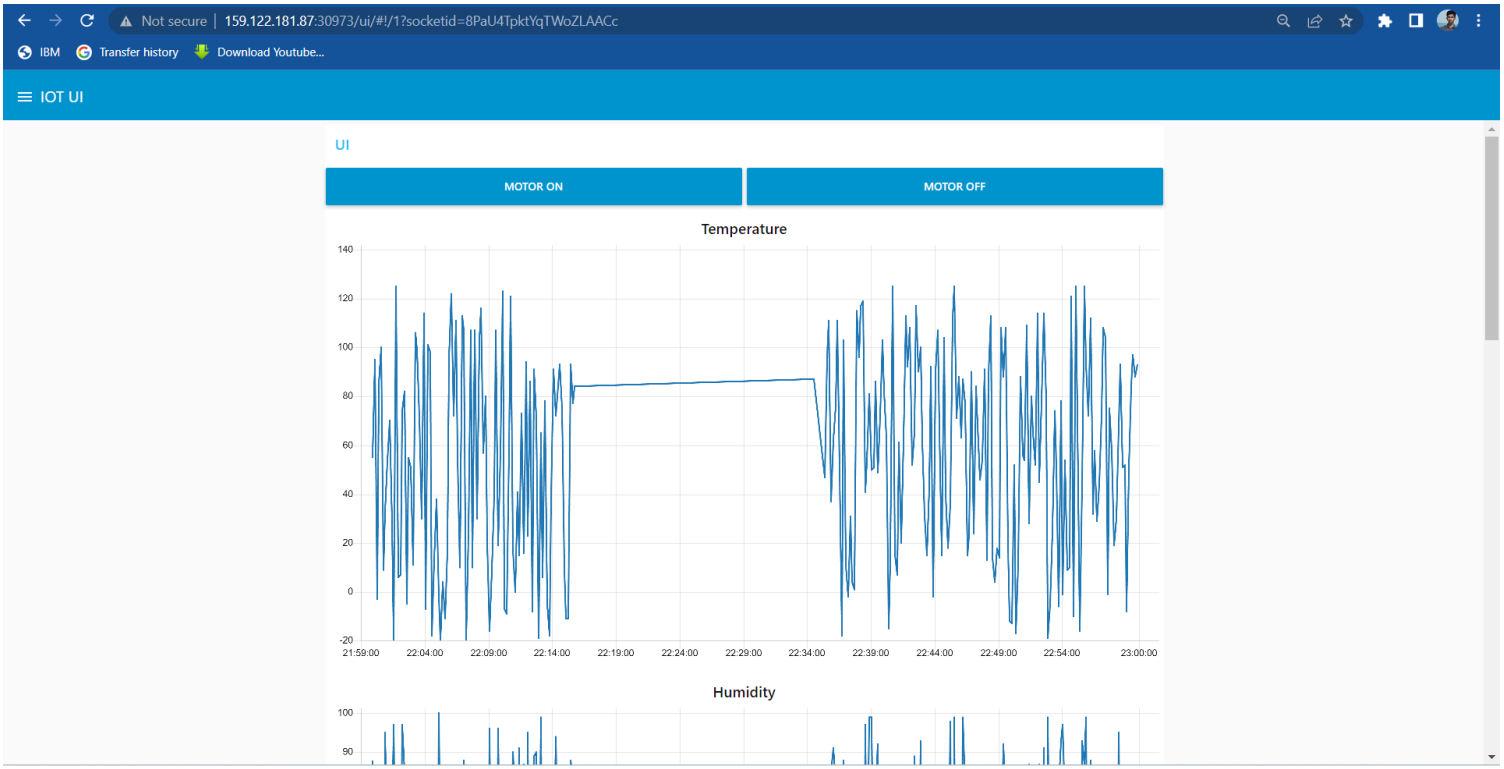
COMPLETE PROGRAM FLOW:

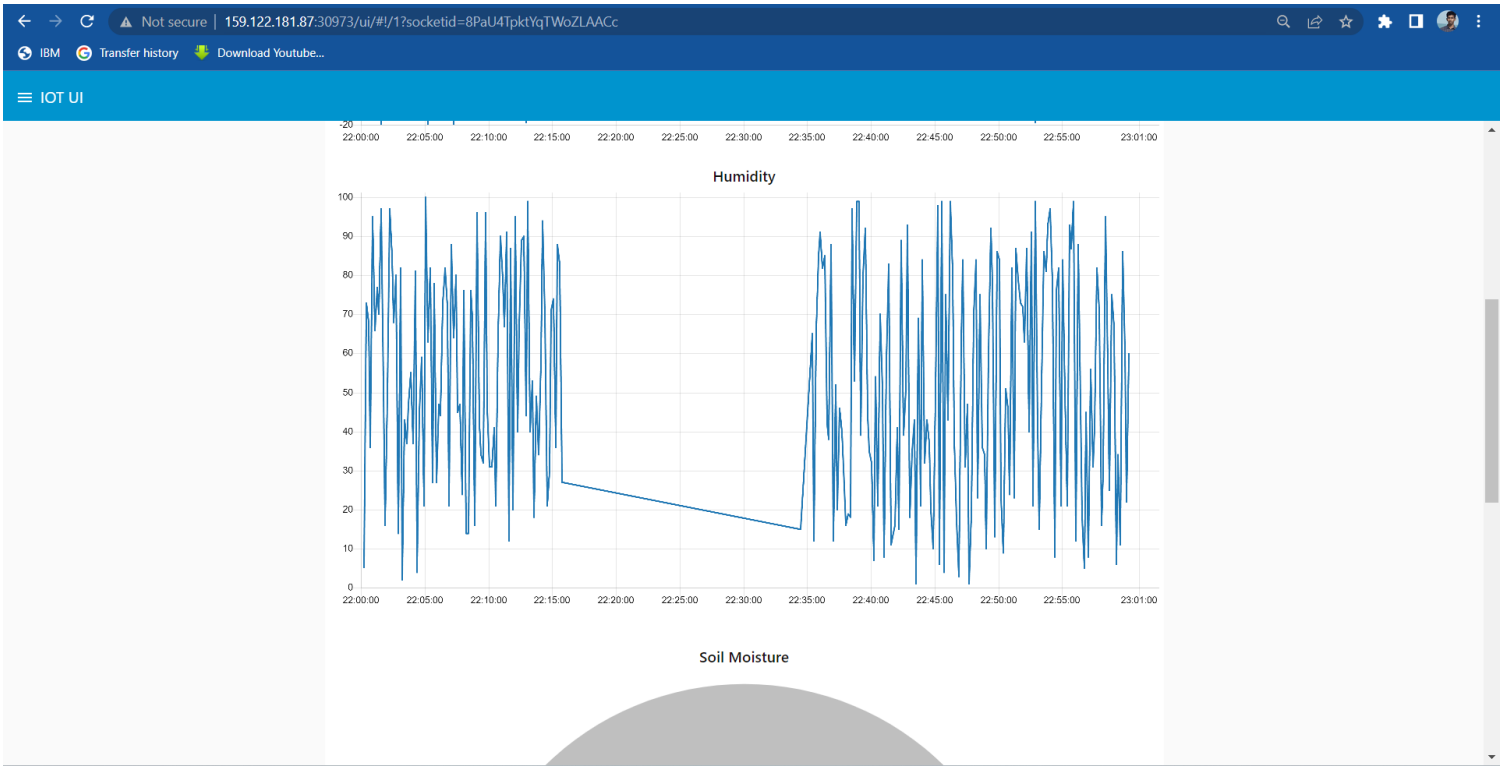


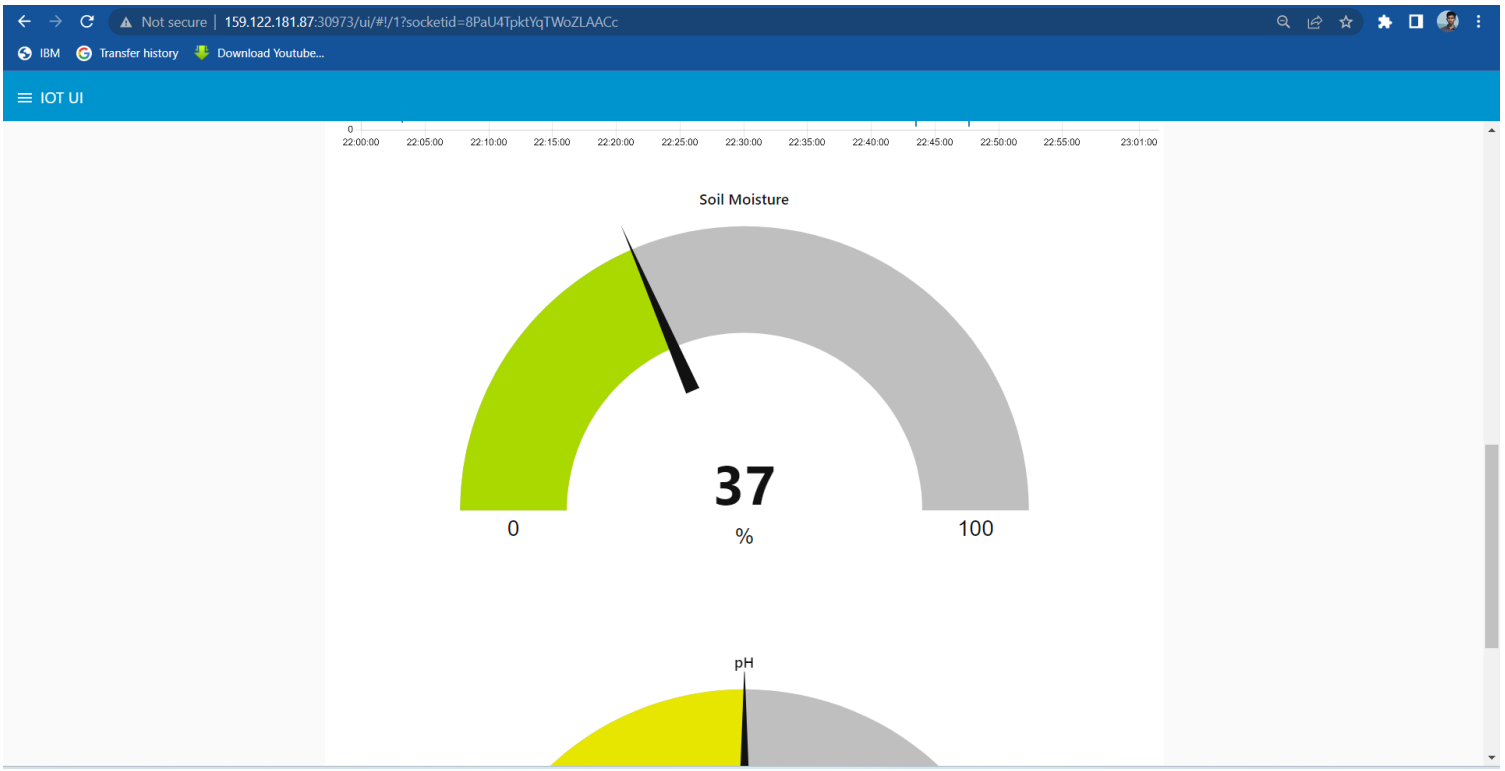
HTML Response:

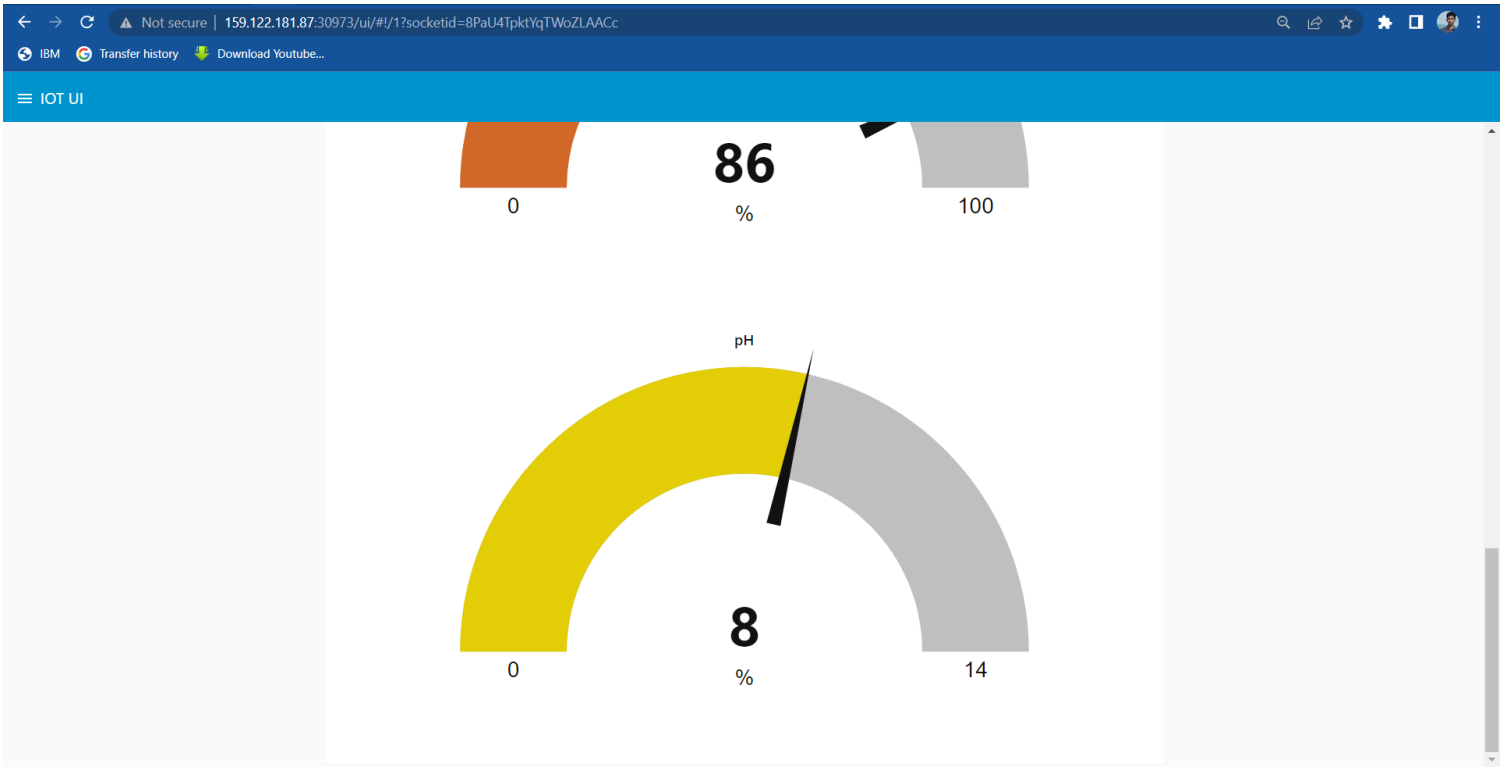
{"pH":5,"Soil":109,"Temperature":15,"Humid":43}

UI Dashboard:




Not secure | 159.122.181.87:30973/ui/#!/1?socketid=8PaU4TpktYqTWoZLAACc
IBM
Transfer history
Download Youtube...
IOT UI
22:00:00 22:05:00 22:10:00 22:15:00 22:20:00 22:25:00 22:30:00 22:35:00 22:40:00 22:45:00 22:50:00 22:55:00 23:01:00
Humidity
100 90 80 70 60 50 40 30 20 10 0
22:00:00 22:05:00 22:10:00 22:15:00 22:20:00 22:25:00 22:30:00 22:35:00 22:40:00 22:45:00 22:50:00 22:55:00 23:01:00
Soil Moisture

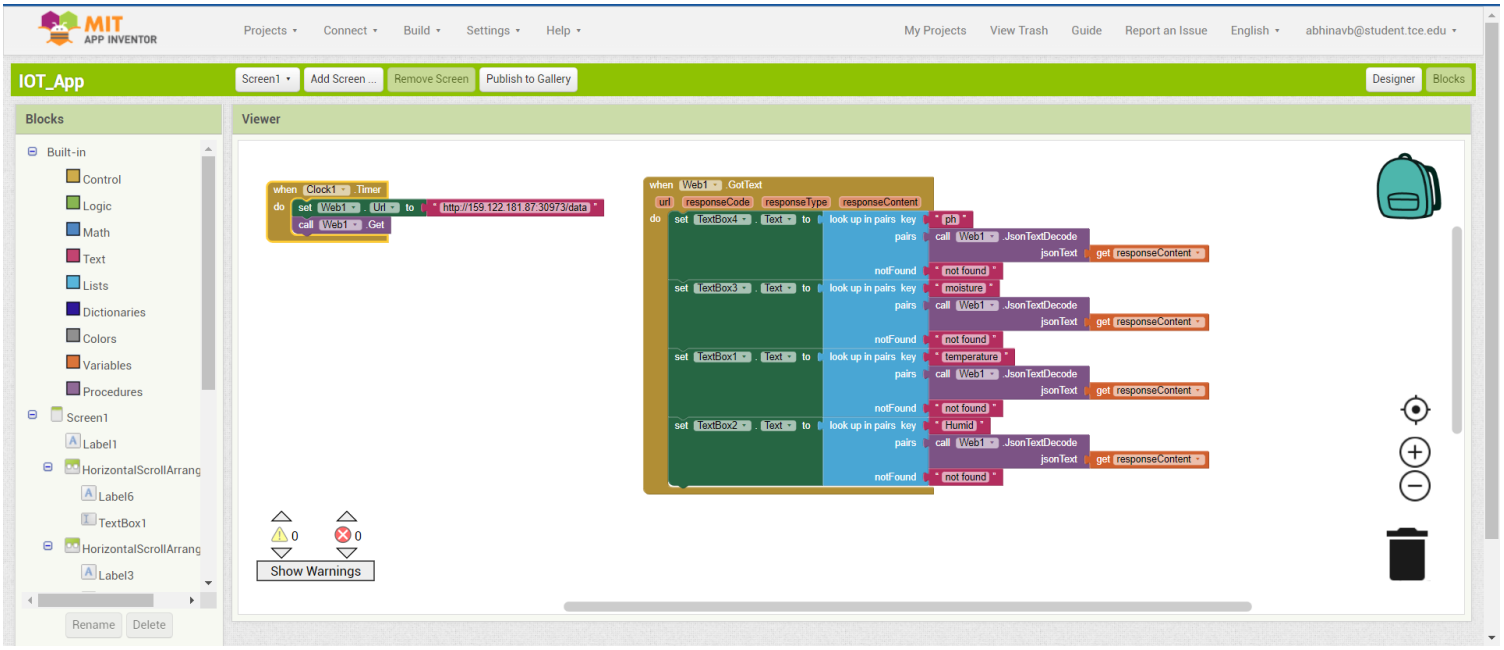

Not secure | 159.122.181.87:30973/ui/#!/1?socketid=8PaU4TpktYqTWoZLAACc
IBM
Transfer history
Download Youtube...
IOT UI
0
22:00:00 22:05:00 22:10:00 22:15:00 22:20:00 22:25:00 22:30:00 22:35:00 22:40:00 22:45:00 22:50:00 22:55:00 23:01:00
Soil Moisture
37
%
0
100
pH

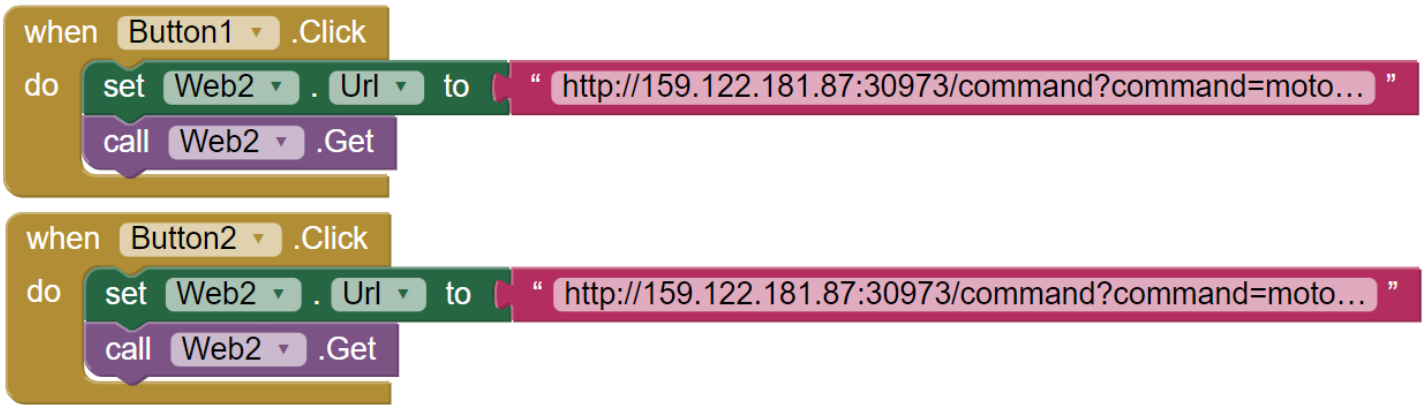


MIT App Inventor:

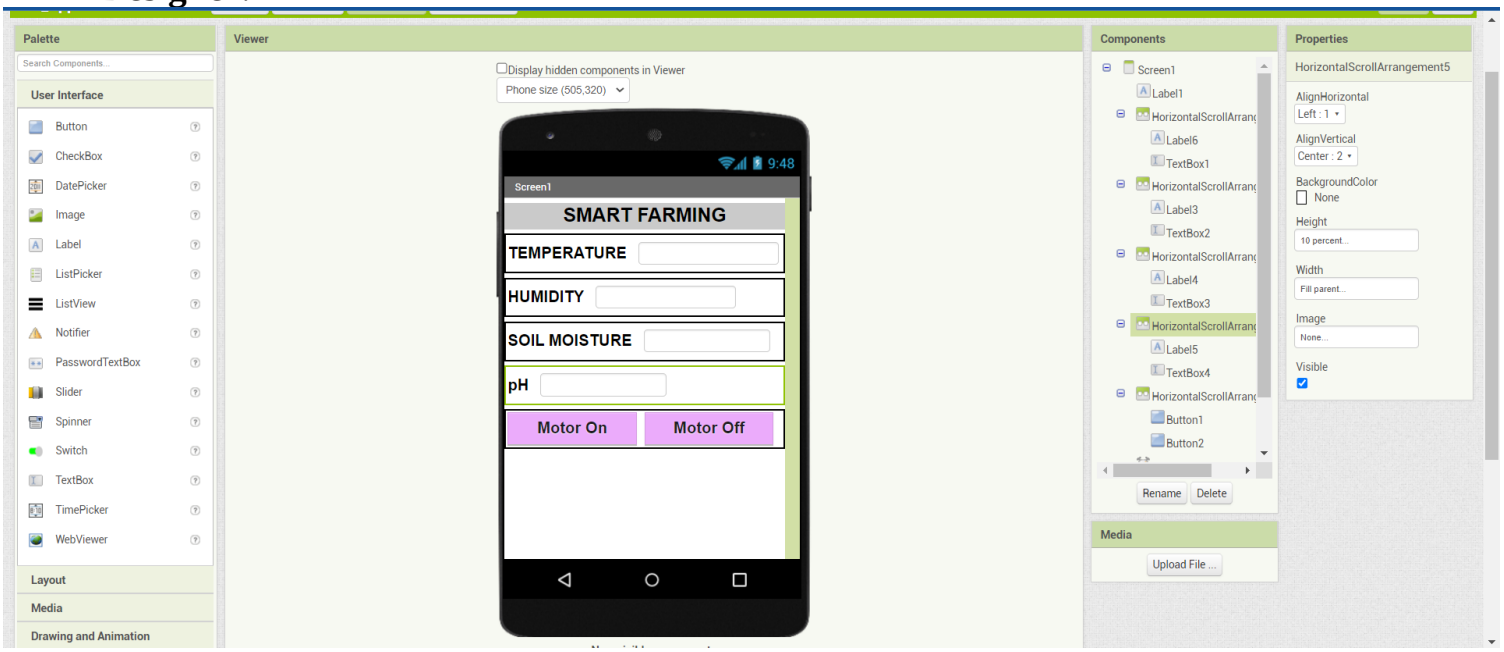
MIT App Inventor is an intuitive, visual programming environment that allows everyone even children to build fully functional apps for smartphones and tablets. Those new to MIT App Inventor can have a simple first app up and running in less than 30 minutes. And what's more, our blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments.

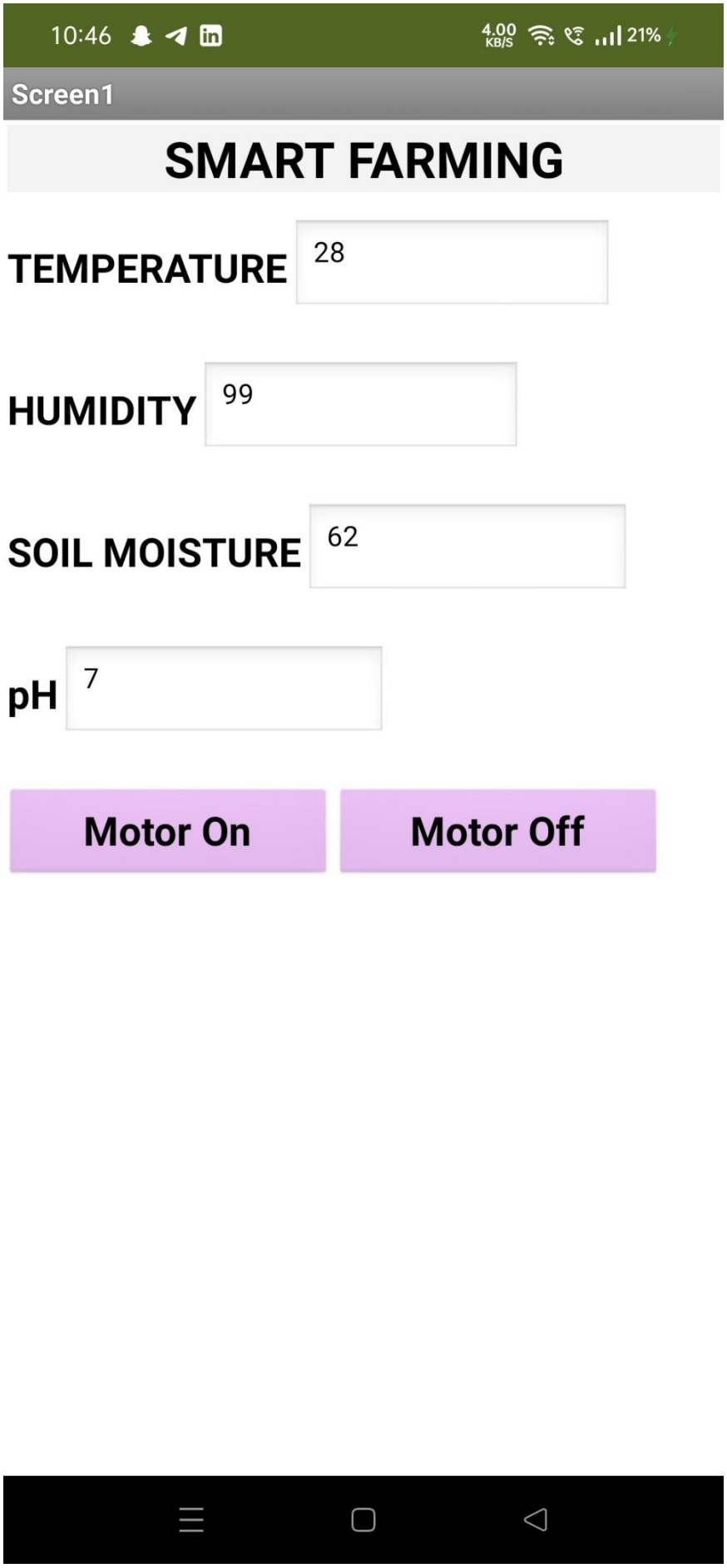
Blocks:



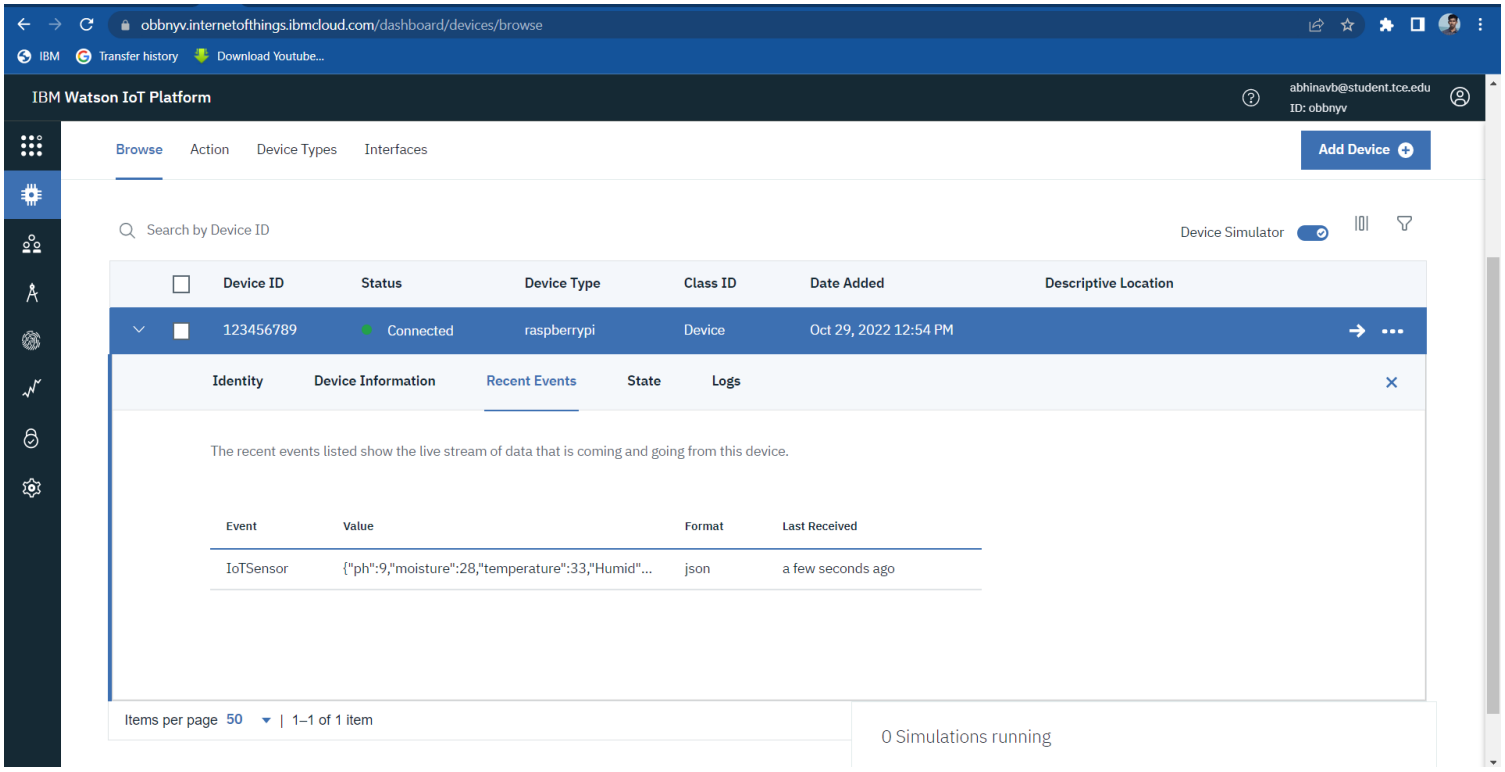

when Button1 .Click
do set Web2 . Url to " http://159.122.181.87:30973/command?command=moto… "
call Web2 .Get
when Button2 .Click
do set Web2 . Url to " http://159.122.181.87:30973/command?command=moto… "
call Web2 .Get

Designer:


Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
SMART FARMING
TEMPERATURE
HUMIDITY
SOIL MOISTURE
pH
Motor On
Motor Off
Components
Screen1
Label1
HorizontalScrollArrang
Label6
TextBox1
Label3
TextBox2
Label4
TextBox3
Label5
TextBox4
Button1
Button2
Rename
Delete
Media
Upload File ...
Properties
HorizontalScrollArrangement5
AlignHorizontal
Left : 1
AlignVertical
Center : 2
BackgroundColor
None
Height
10 percent...
Width
Fill parent...
Image
None...
Visible


10:46
4.00 KB/S
21%
Screen1
SMART FARMING
TEMPERATURE
28
HUMIDITY
99
SOIL MOISTURE
62
pH
7
Motor On
Motor Off

IBM Watson:


obbnyv.internetofthings.ibmcloud.com/dashboard/devices/browse
IBM
Transfer history
Download Youtube...
IBM Watson IoT Platform
abhinavb@student.tce.edu
ID: obbnyv
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
123456789
Connected
raspberrypi
Device
Oct 29, 2022 12:54 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor
{"ph":9,"moisture":28,"temperature":33,"Humid"...
json
a few seconds ago
Items per page 50
1–1 of 1 item
0 Simulations running