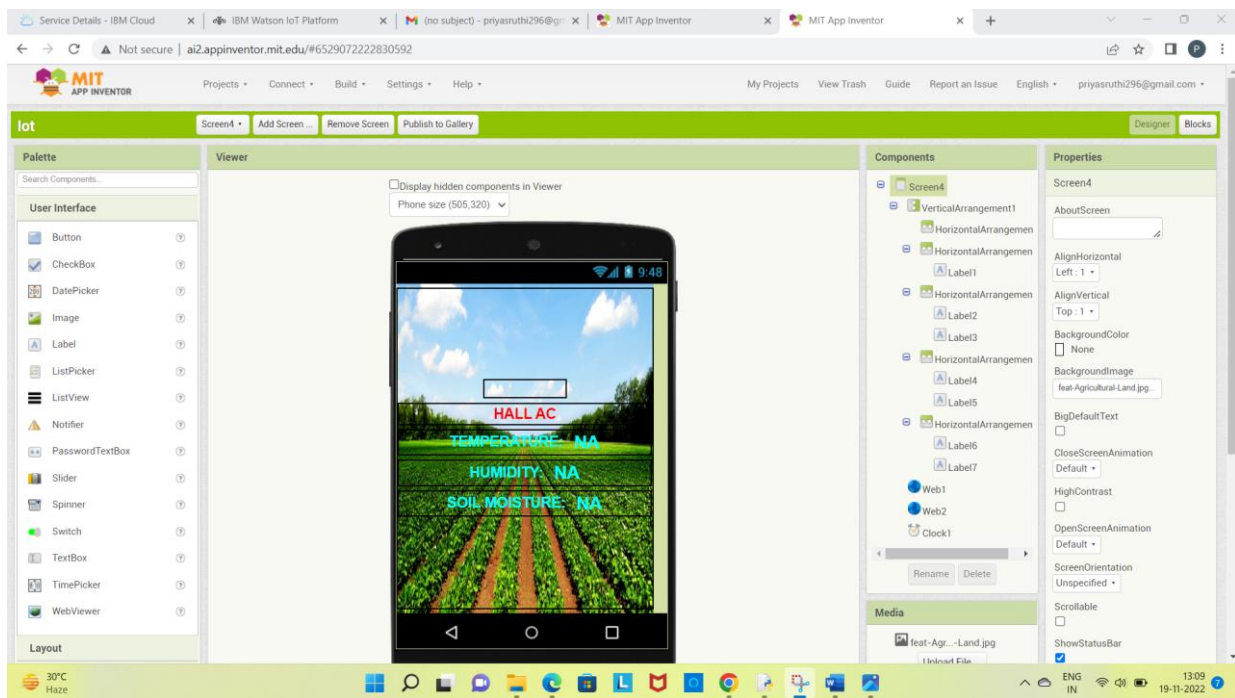


SPRINT-4

TEAM ID: PNT2022TMID54441

PROJECT NAME : IoT based smart crop protection system for Agriculture

Step1:MIT APP inventor to design the APP. This is the user interface of the app



Step2:Customize the APP interface to display the values

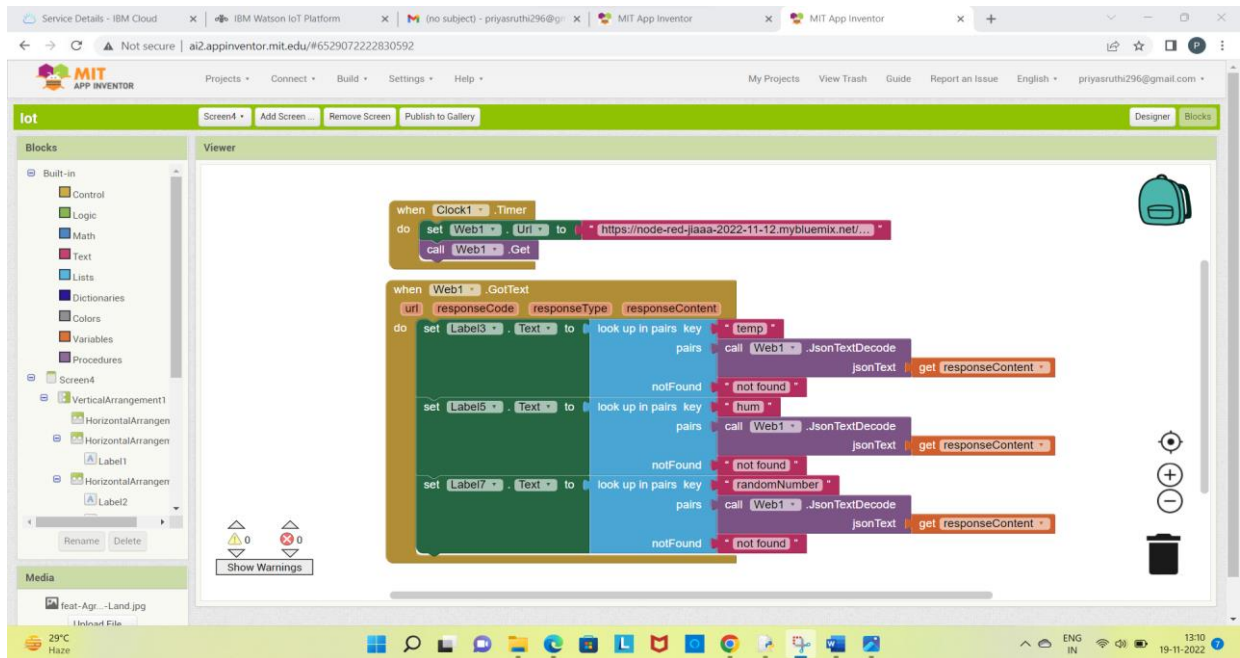
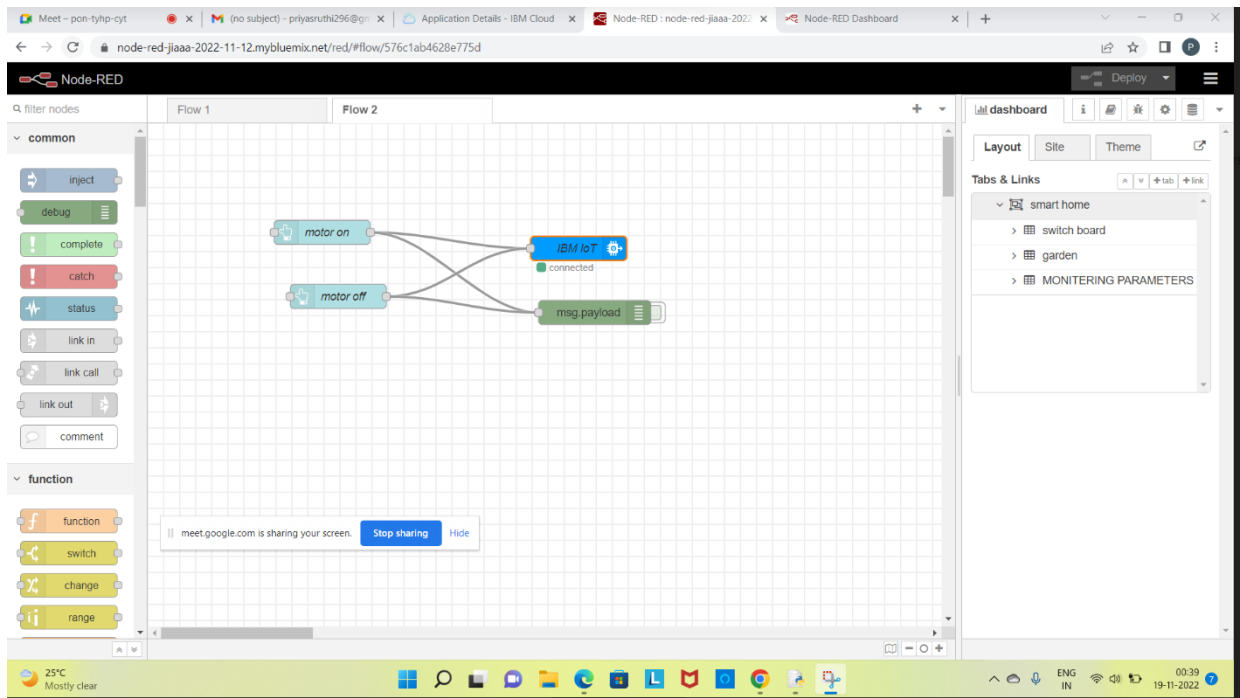
Firstly, Python and IBM Watson cloud platform are interfaced

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for navigation. The main content area displays a table of devices. The first device, '1206', is highlighted and its details are shown below the table. The details include a 'Recent Events' section with a table of events.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1206	Connected	Rasperry	Device	Nov 12, 2022 2:18 PM	

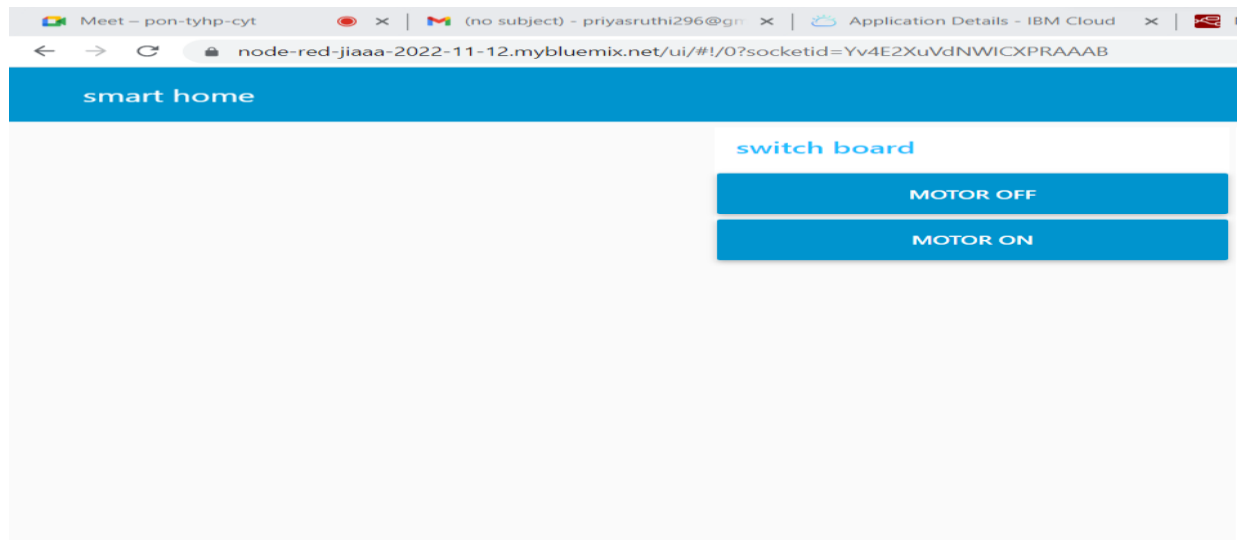
Event	Value	Format	Last Received
Alert1	{"alert1": "Temperature(76.74) is high, sprinklerle..."}	json	a few seconds ago
Water sensor	{"Water Level": 20.81}	json	a few seconds ago
Moisture sen...	{"Moisture Level": 7.79}	json	a few seconds ago
Flame sensor	{"Flame": "Not Detected"}	json	a few seconds ago

The screenshot shows the Node-RED web interface. The main workspace displays a flow named 'Flow 1'. The flow starts with an 'IBM IoT' node, which is connected to a 'msg.payload' node. This node then branches into three parallel paths, each leading to a 'temp', 'hum', and 'soil moisture' node. These nodes are then connected to a 'msg.payload' node, which is finally connected to an 'http' node. The right sidebar shows a 'dashboard' section with a 'smart home' tab and a 'MONITORING PARAMETER' section.



OUTPUT:





If the button switch is clicked, motor on and off will be indicated in the python shell prompt.

The screenshot shows a Python 3.7.0 Shell window with the following code and output:

```

import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

# Provide your IBM Watson Device Credentials
organization = "8gyz7t" # replace the ORG ID
deviceType = "weather_monitor" # replace the Device type
deviceId = "b827ebd607b5" # replace Device ID
authMethod = "token"
authToken = "LMVpQPaVQl66HWN40z" # Replace the auth token

def myCommandCallback(cmd): # function for Callback
    if cmd.data['command'] == 'motoron':
        print("MOTOR ON IS RECEIVED")
    elif cmd.data['command'] == 'motoroff':
        print("MOTOR OFF IS RECEIVED")
    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)
# .....

```

The output of the script shows a list of 20 sensor readings (Temperature, Humidity, Soil Moisture) for the device "b827ebd607b5". The output also shows a "RESTAPI" message indicating a successful connection to the IBM Watson IoT Platform. The final output is "MOTOR OFF IS RECEIVED".