

VEL TECH MULTI TECH DR.RANGARAJAN DR.SAKUNTHALA ENGINEERING COLLEGE

Department of Electronics and Communication Engineering

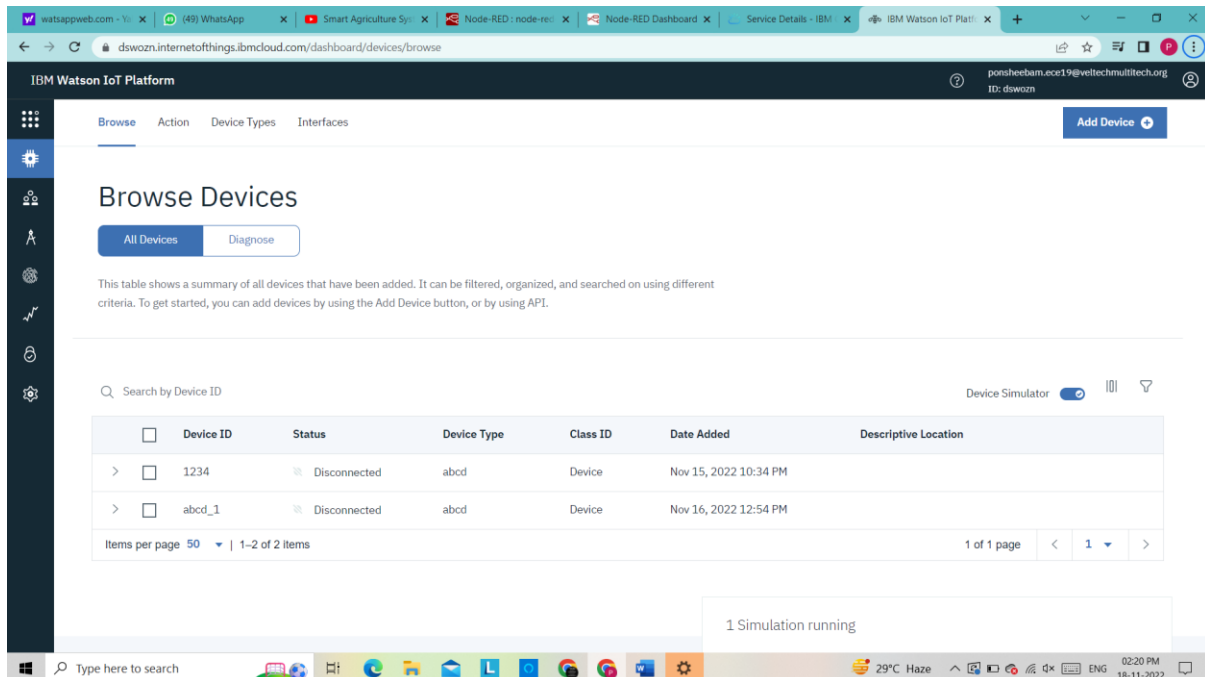
Smart Farmer-IOT Enabled Smart Farming SPRINT -4

TEAM ID	PNT2022TMID22476
LEADER NAME	SABARIYA A
TEAM MEMBER NAME	GAYATHRI M S PRIYADHARSHINI A PONSHEEBA M
MENTOR NAME	KALPANA

WEB UI:

IOT WATSON PLATFORM ,WOKWI ,NODE RED AND MIT APP INVENTOR
INTRACTION :

1.IOT WATSON PLATFORM AND WOKWI INTERACTION:

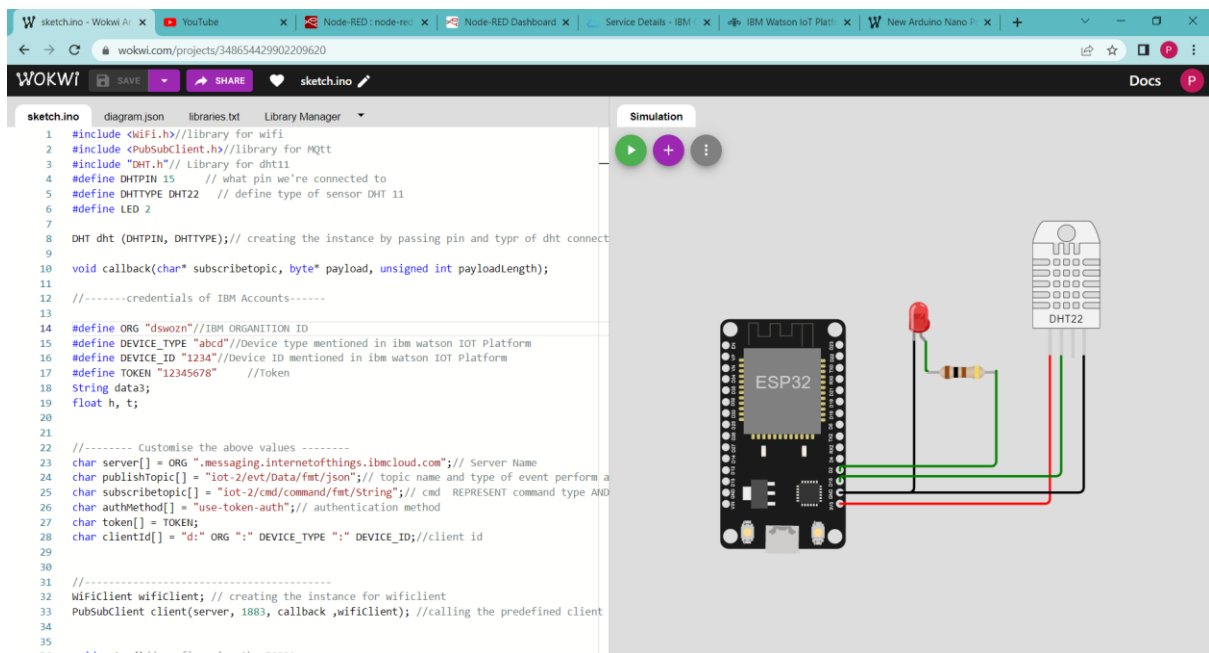


The screenshot displays the IBM Watson IoT Platform interface. The main heading is 'Browse Devices'. Below it, there are tabs for 'All Devices' and 'Diagnose'. A message states: 'This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.' Below this is a search bar labeled 'Search by Device ID'. The table below lists two devices:

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	abcd	Device	Nov 15, 2022 10:34 PM	
abcd_1	Disconnected	abcd	Device	Nov 16, 2022 12:54 PM	

At the bottom of the table, it says 'Items per page 50 | 1-2 of 2 items'. Below the table, there is a 'Device Simulator' toggle and a notification that says '1 Simulation running'.

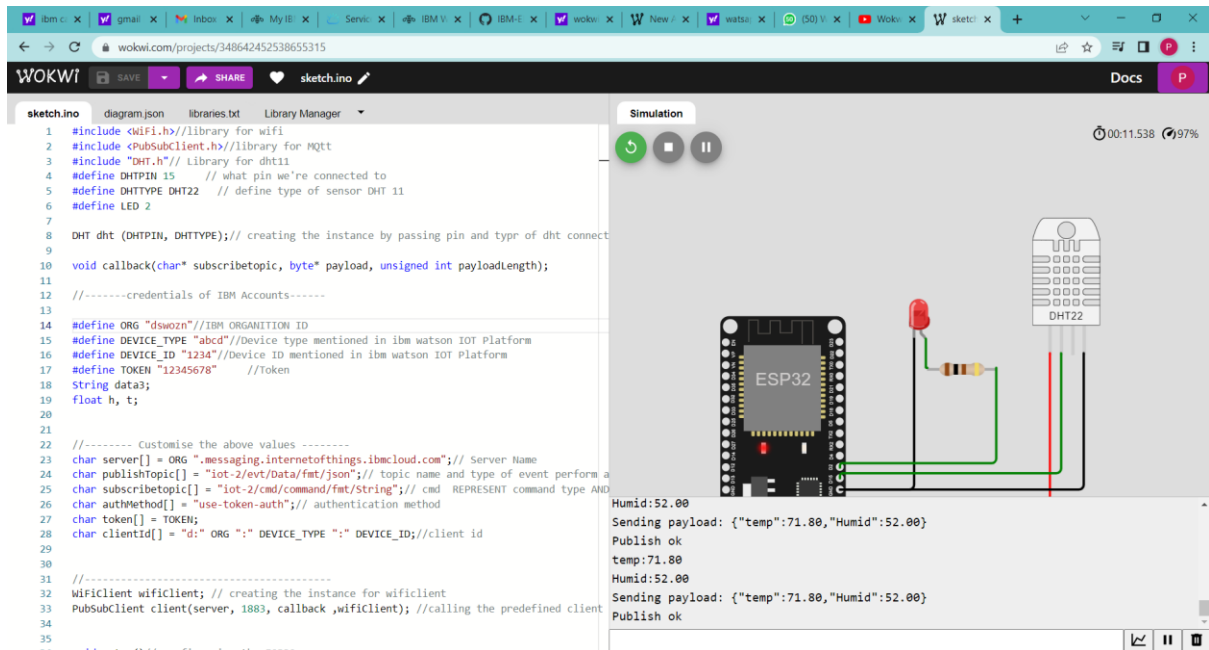
WOKWI CONNECTION AND CODE :



The screenshot shows the Wokwi web interface with the code editor on the left and the simulation on the right. The code is an Arduino sketch for an ESP32 connected to a DHT22 sensor. It includes libraries for WiFi, MQTT, and DHT. The code defines pins for the DHT22 sensor and an LED, and sets up a WiFi connection to an IBM Watson IoT Platform. The simulation shows the ESP32 board connected to the DHT22 sensor via I2C.

```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 15 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define LED 2
7
8 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of dht connect
9
10 void callback(char* subscribetopic, byte* payload, unsigned int payloadlength);
11
12 //-----credentials of IBM Accounts-----
13
14 #define ORG "dswozn" //IBM ORGANIZATION ID
15 #define DEVICE_TYPE "abcd" //Device type mentioned in ibm watson IOT Platform
16 #define DEVICE_ID "1234" //Device ID mentioned in ibm watson IOT Platform
17 #define TOKEN "12345678" //Token
18 String data3;
19 float h, t;
20
21 //----- Customise the above values -----
22 char server[] = ORG ".messaging.internetofthings.ibmcloud.com"; // Server Name
23 char publishTopic[] = "iot-2/evt/data/fmt/json"; // topic name and type of event perform a
24 char subscribetopic[] = "iot-2/cmd/command/fmt/String"; // cmd REPRESENT command type AND
25 char authMethod[] = "use-token-auth"; // authentication method
26 char token[] = TOKEN;
27 char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID; //client id
28
29 //-----
30
31 //-----
32 WiFiClient wifiClient; // creating the instance for wifiClient
33 PubSubClient client(server, 1883, callback, wifiClient); //calling the predefined client
34
35
36 void setup() { // configuring the pins
```

AFTER WOKWI SIMULATION THE CONNECTION OF WOKWI AND IOT WATSON PLATFORM:



The screenshot shows the Wokwi web interface with the code editor on the left and the simulation on the right. The code is the same as in the previous screenshot. The simulation shows the ESP32 board connected to the DHT22 sensor via I2C. A terminal window at the bottom right displays the output of the simulation, showing the temperature and humidity readings.

```
Humid:52.00
Sending payload: {"temp":71.80,"Humid":52.00}
Publish ok
temp:71.80
Humid:52.00
Sending payload: {"temp":71.80,"Humid":52.00}
Publish ok
```

IBM Watson IoT Platform

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Connected	abcd	Device	Nov 15, 2022 10:34 PM	
abcd_1	Disconnected	abcd	Device	Nov 16, 2022 12:54 PM	

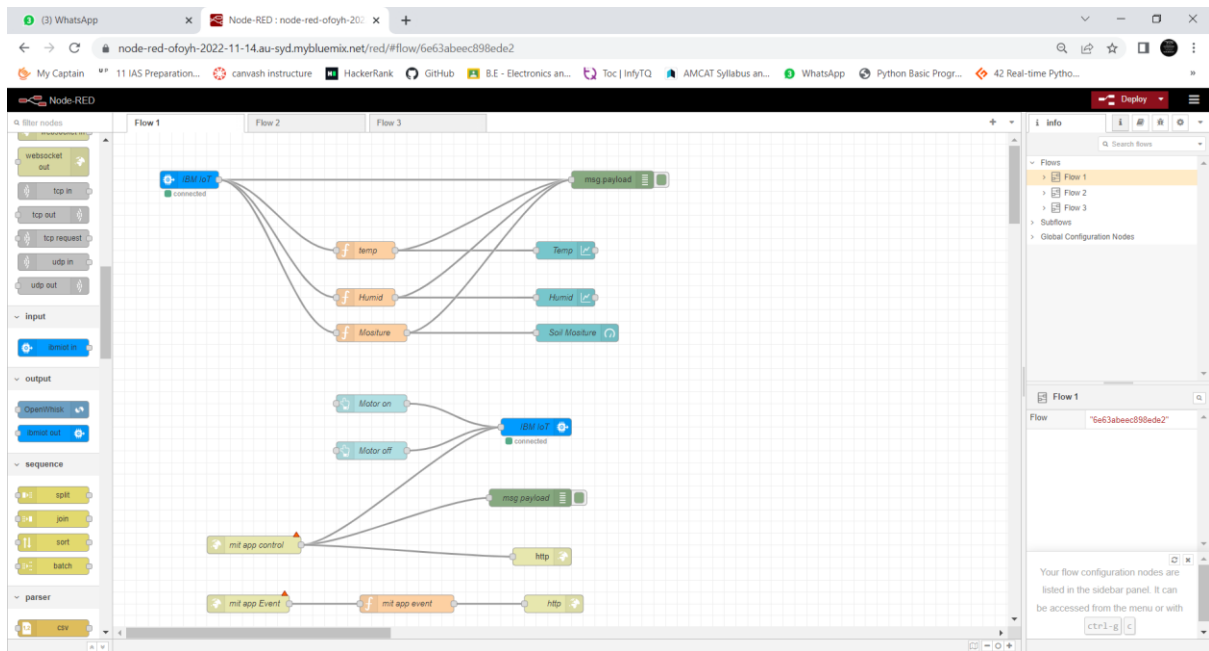
Items per page 50 | 1-2 of 2 items

1 of 1 page

1 Simulation running

The device in IOT Watson platform dhow's connected

2.NODE RED AND MIT :



Node-RED interface showing a flow named "Flow 1". The flow starts with an "IBM IoT" node (connected) which triggers three parallel "function" nodes: "temp", "Humid", and "Moisture". These nodes output data to a "Line chart" node. The "Line chart" node is configured with the following properties:

- Group: [Smart Farmer] Humidity
- Size: auto
- Label: temp
- Type: Line chart
- X-axis: last 4 minutes OR 1000 points
- X-axis Label: HH:mm:ss
- Y-axis: min max
- Legend: None
- Interpolate: linear
- Series Colours: (various colors)

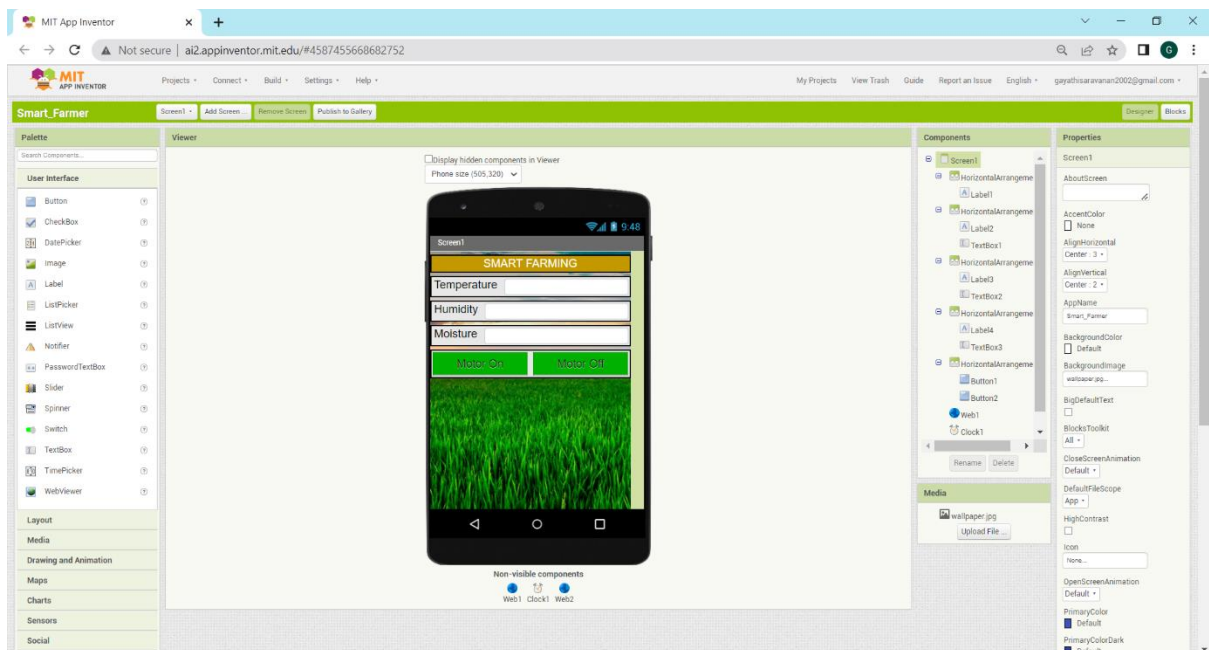
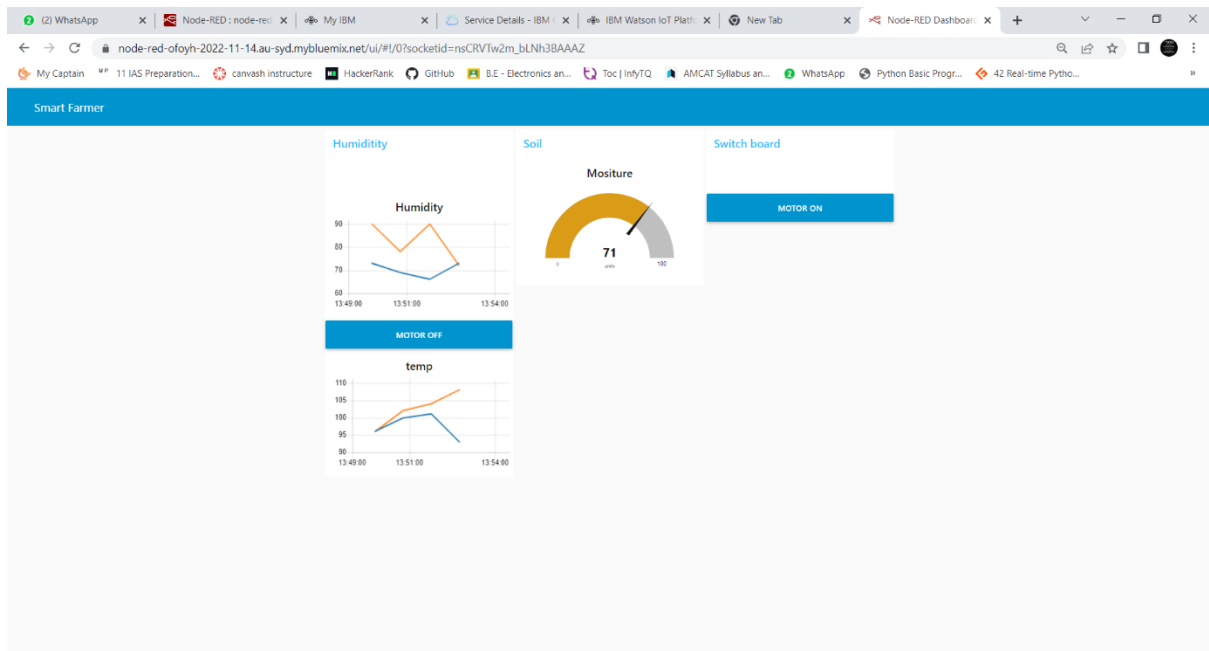
The "debug" console shows the following log entries:

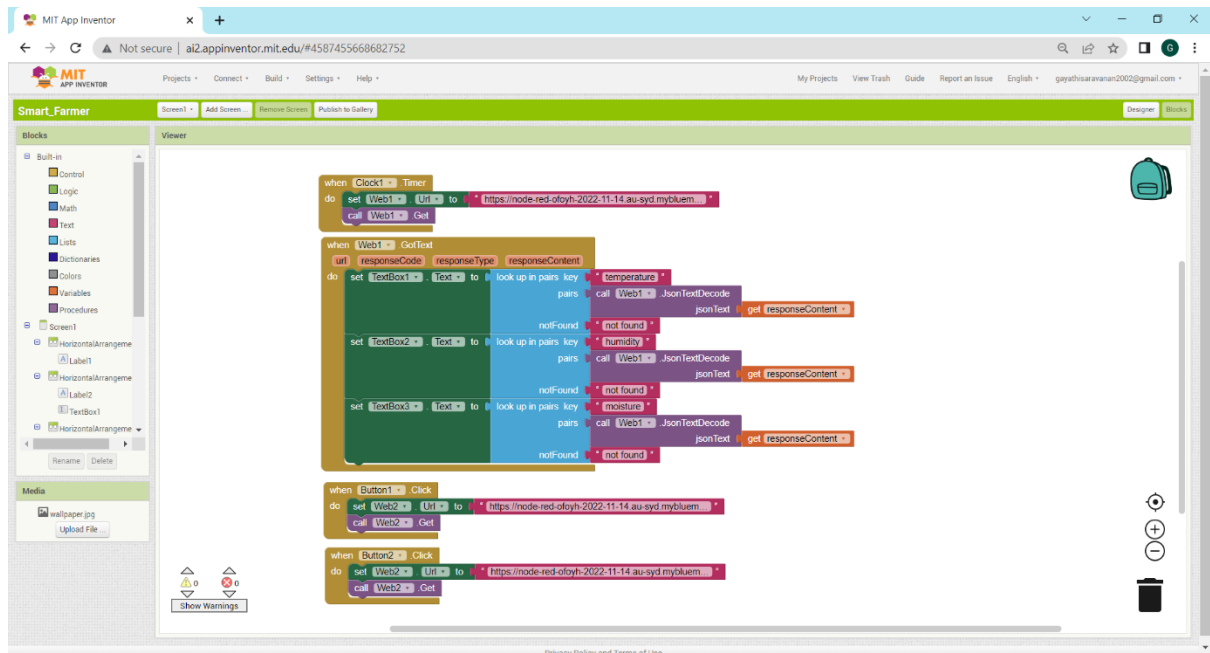
```
11/16/2022, 12:54:21 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload : Object  
{ randomNumber: 36, temp: 93, Humid: 68 }  
11/16/2022, 12:54:21 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
93  
11/16/2022, 12:54:22 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
68  
11/16/2022, 12:54:23 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
36  
11/16/2022, 12:54:24 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
Object  
{ randomNumber: 21, temp: 105, Humid: 97 }  
11/16/2022, 12:54:25 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
105  
11/16/2022, 12:54:26 PM node: 122649a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :
```

Node-RED interface showing a flow named "Flow 1". The flow starts with an "IBM IoT" node (connected) which triggers three parallel "function" nodes: "temp", "Humid", and "Moisture". These nodes output data to a "msg.payload" node. The "msg.payload" node triggers three parallel "function" nodes: "Temp", "Humid", and "Soil Moisture". These nodes output data to a "msg.payload" node. The "msg.payload" node triggers three parallel "function" nodes: "Motor on", "Motor off", and "MIT app control". These nodes output data to a "msg.payload" node. The "msg.payload" node triggers three parallel "function" nodes: "MIT app event", "MIT app control", and "MIT app event". These nodes output data to a "msg.payload" node. The "msg.payload" node triggers three parallel "function" nodes: "MIT app event", "MIT app control", and "MIT app event". These nodes output data to a "msg.payload" node.

The "debug" console shows the following log entries:

```
11/16/2022, 1:32:31 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
98  
11/16/2022, 1:32:32 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
92  
11/16/2022, 1:32:33 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
68  
11/16/2022, 1:32:34 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
Object  
{ randomNumber: 70, temp: 100, Humid: 80 }  
11/16/2022, 1:32:35 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
100  
11/16/2022, 1:32:36 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
80  
11/16/2022, 1:32:37 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
Object  
{ randomNumber: 24, temp: 97, Humid: 74 }  
11/16/2022, 1:32:38 PM node: 020849a-0d0d98  
iot-2/type/abcd/1234/ev/evflow/ftm/json : msg.payload :  
number  
97
```





WEB USER LOGIN:

2:01 PM

•   R  79%

Screen1

Login and Signup Form

Username

Password

Login

SignUp

2:01 PM

•   R  79%

Screen1

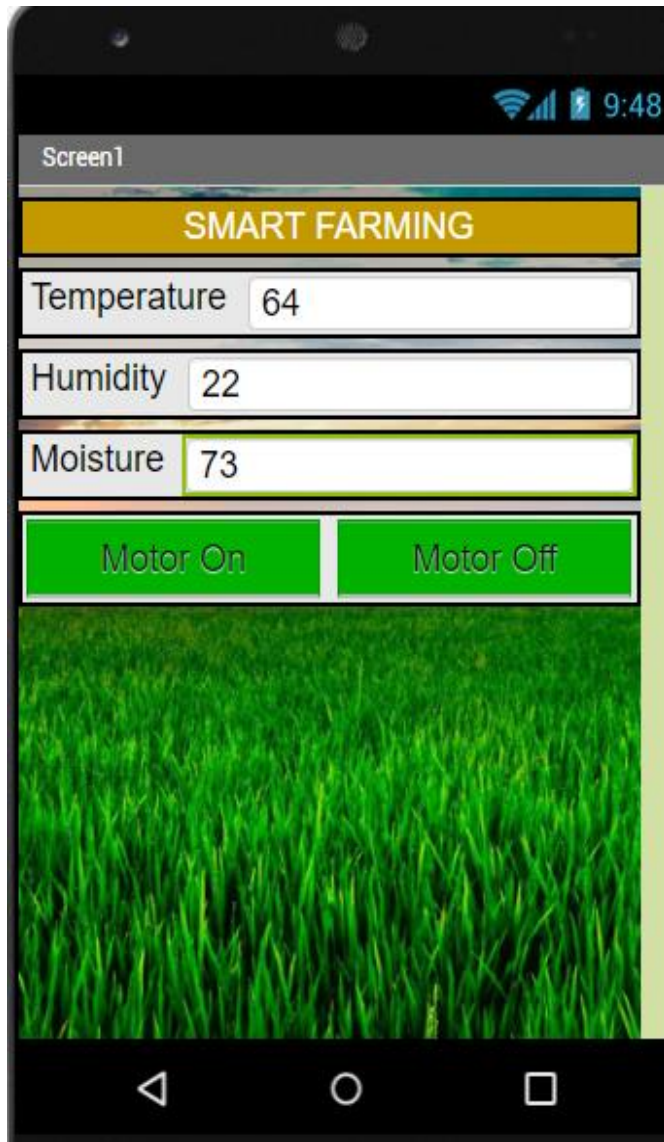
Login and Signup Form

Gayathri

....

Login

SignUp



The user first create a login id and password then they login using that id and password . If the login id was correct then it would proceed to the next screen were it shows the temperature ,humidity and soil moisture so they can track the values .