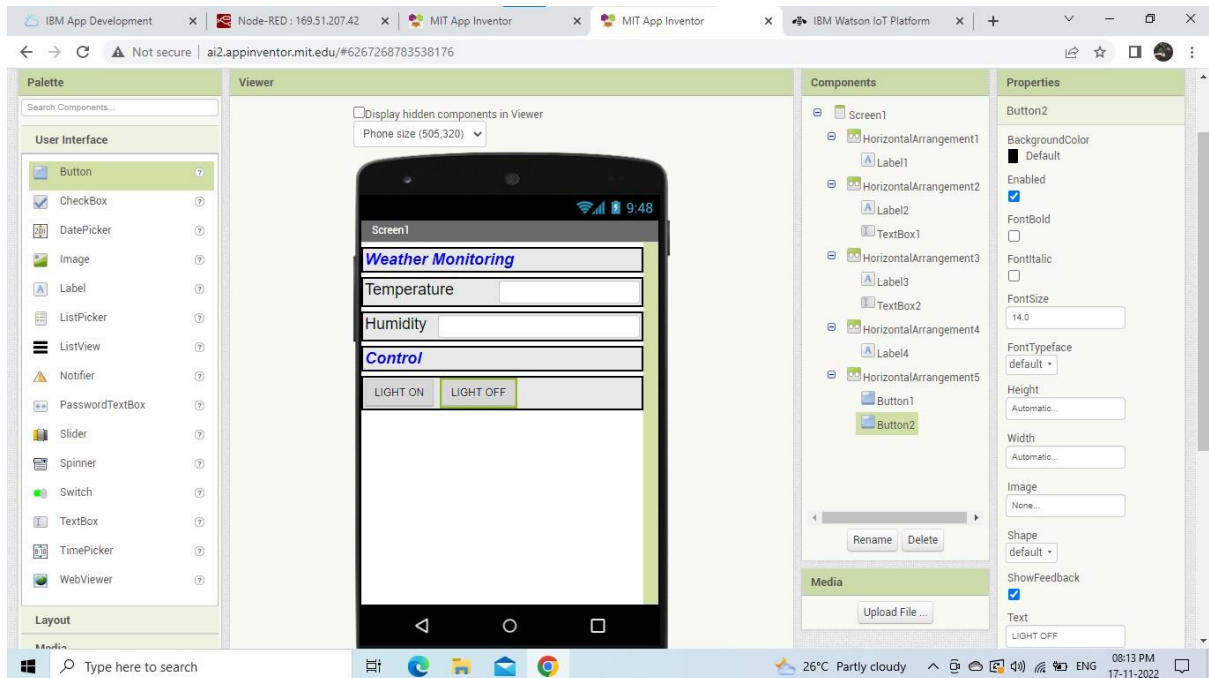


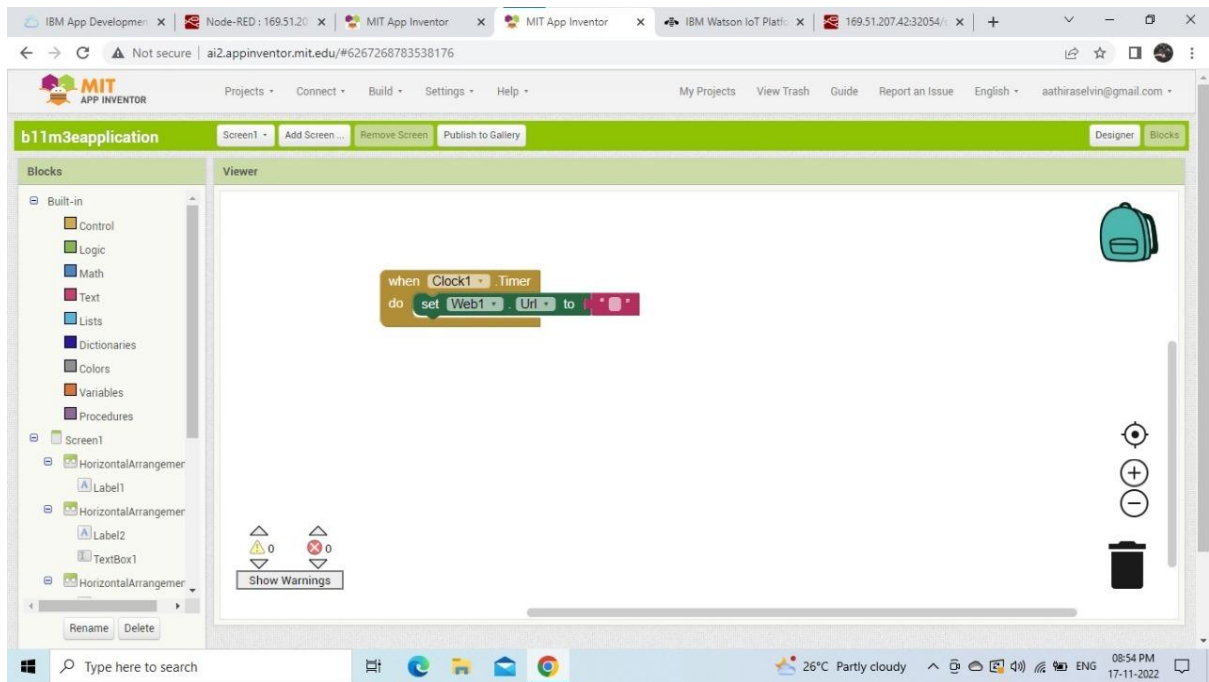
PROJECT DEVELOPMENT PHASE

SPRINT 4

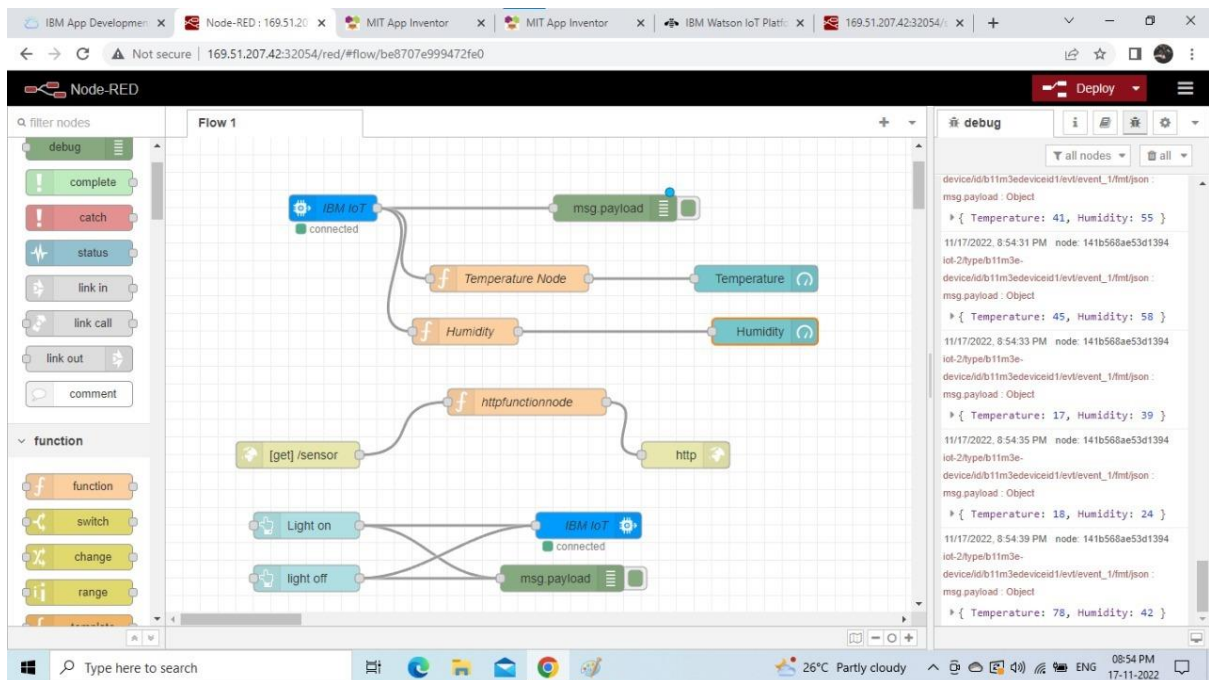
TEAM ID PROJECT	PNT2022TMID34075
TITLE	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

Developing MIT app and publishing the data





NODE-RED FLOW



PUBLISHING THE DATA TO IBM WATSON PLATFORM

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays details for a device named 'b11m3edevicid1', which is in a 'Disconnected' state. Below the device information, there is a section for 'Recent Events' with a table showing a stream of data.

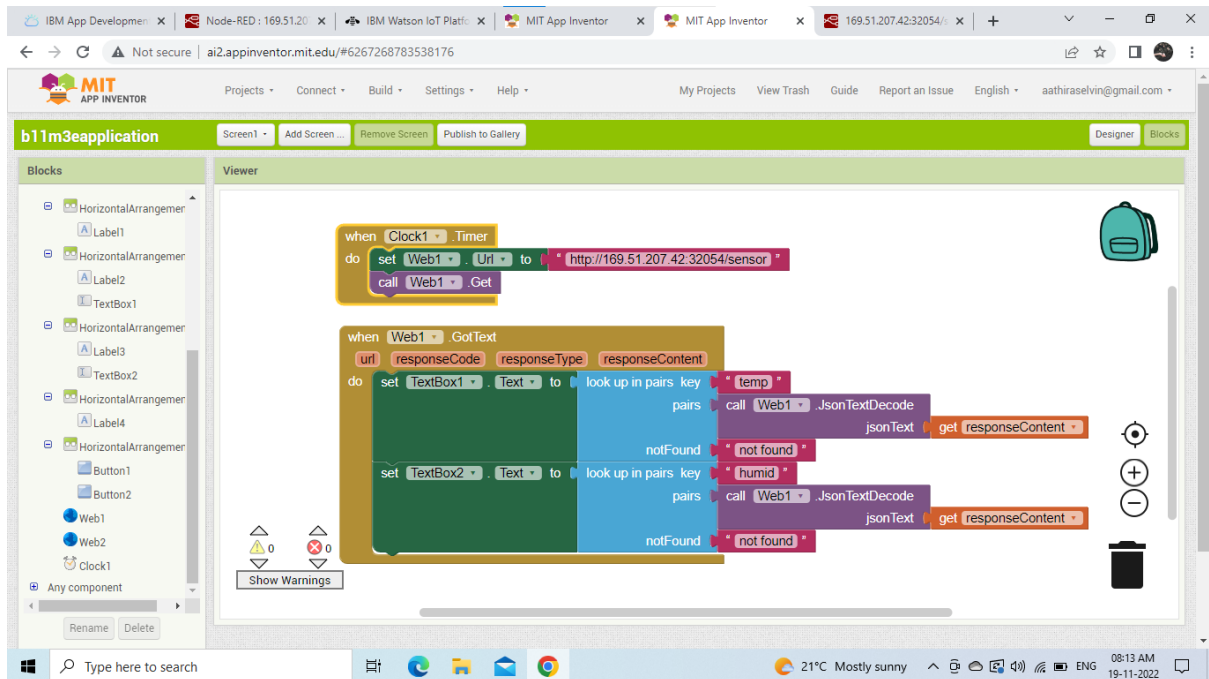
Event	Value	Format	Last Received
event_1	{"Temperature":38,"Humidity":52}	json	a few seconds ago
event_1	{"Temperature":62,"Humidity":61}	json	a few seconds ago
event_1	{"Temperature":92,"Humidity":4}	json	a few seconds ago
event_1	{"Temperature":60,"Humidity":41}	json	a few seconds ago
event_1	{"Temperature":22,"Humidity":97}	json	a few seconds ago

Below the table, it indicates '1 Simulation running'.

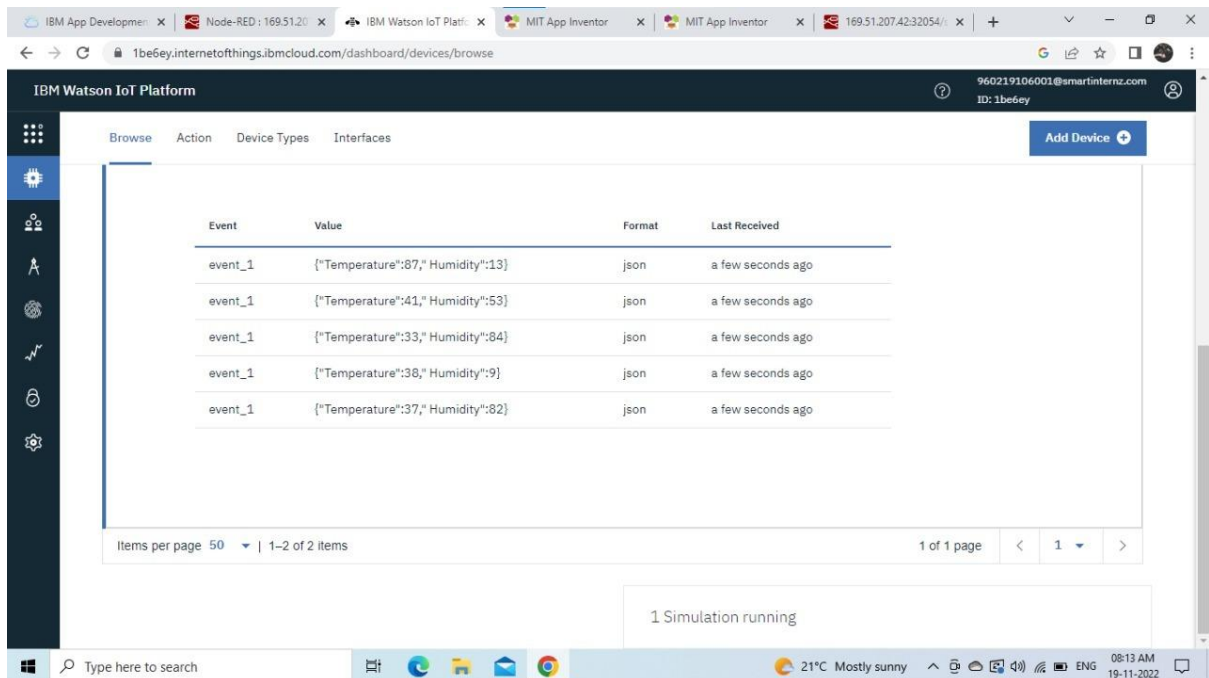
NODE-RED SENSOR DATA

The screenshot shows a web browser window displaying a Node-RED interface. The address bar indicates the URL '169.51.207.42:32054/sensor'. The main content area shows a JSON object representing sensor data: `{"temp":94,"humid":85}`. The browser's taskbar at the bottom shows the system time as 08:55 PM on 17-11-2022.

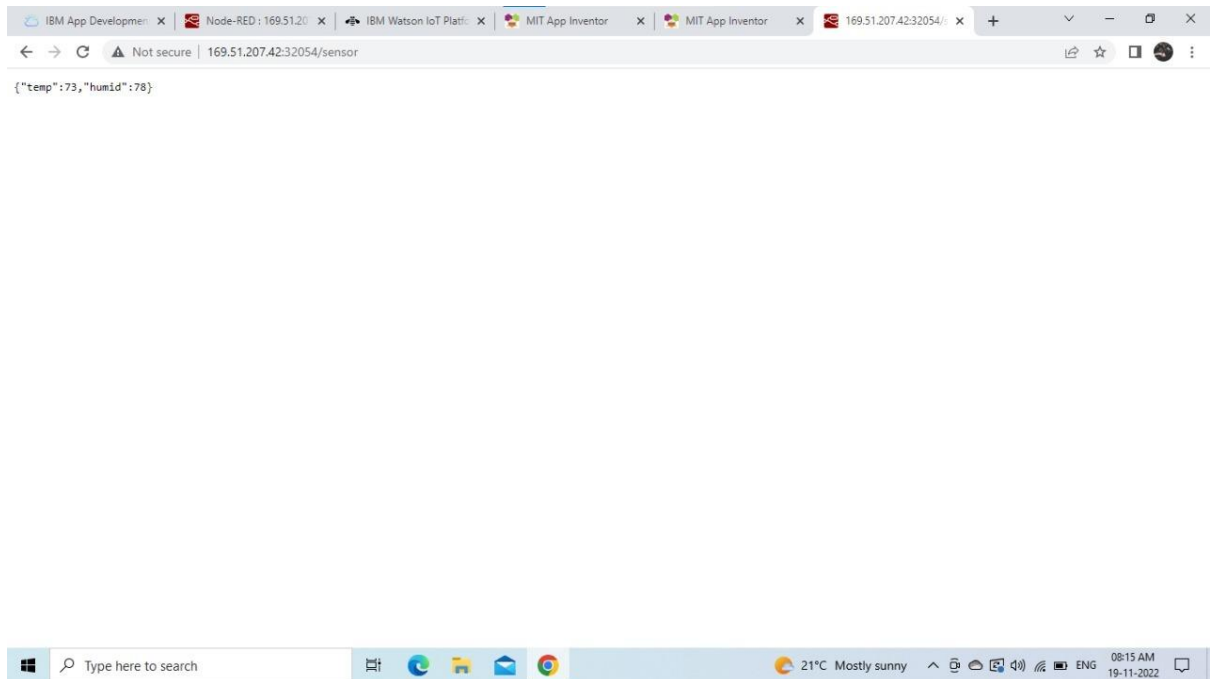
MIT APP DEVELOPMENT



OUTPUT IN IBM WATSON



NODE-RED SENSOR OUTPUT



MOBILE APP

Screen1

Weather Monitoring

Temperature 44

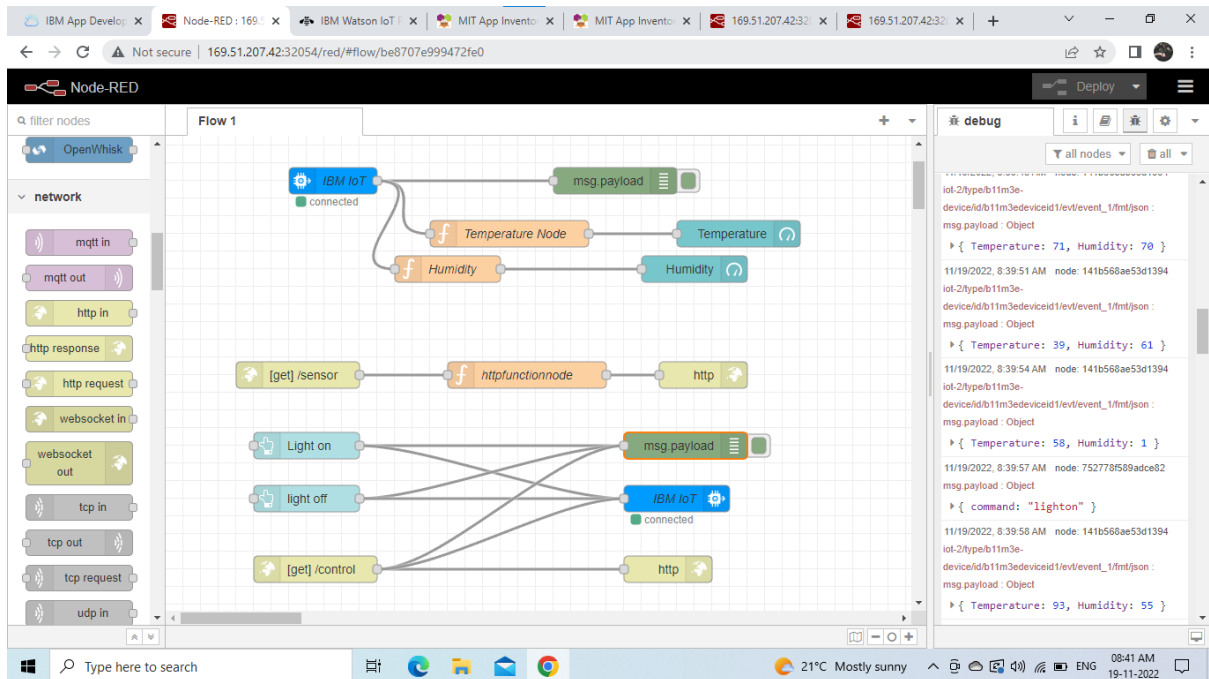
Humidity 68

Control

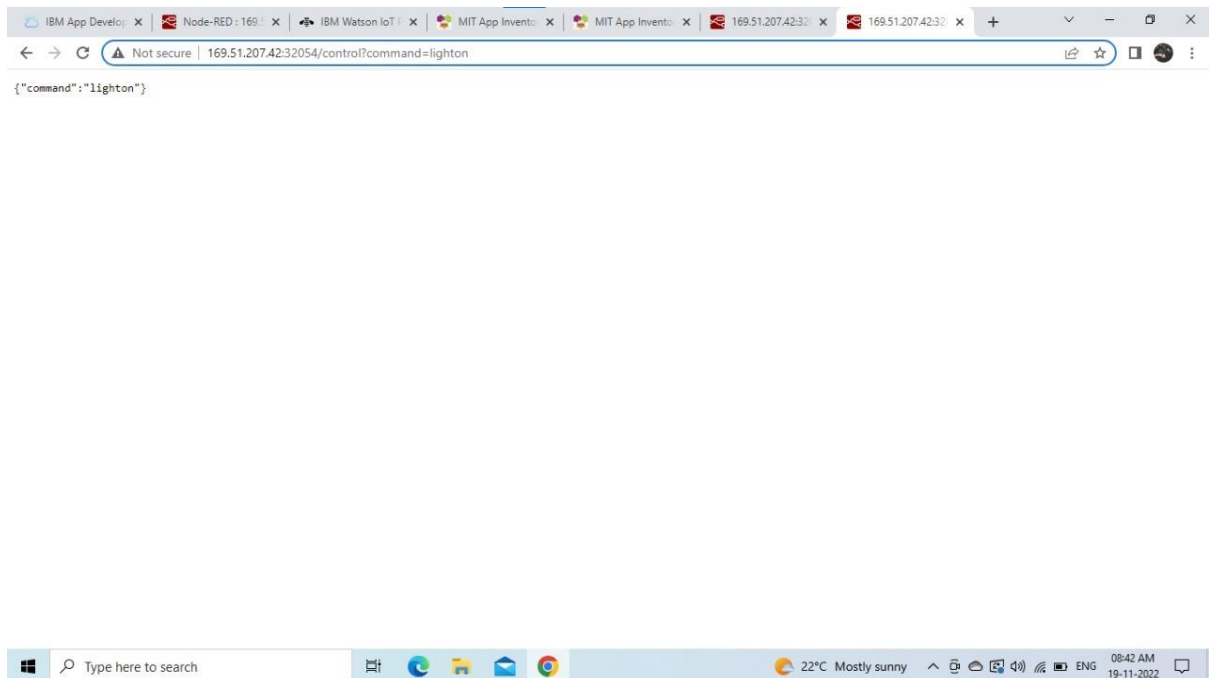
LIGHT ON

LIGHT OFF

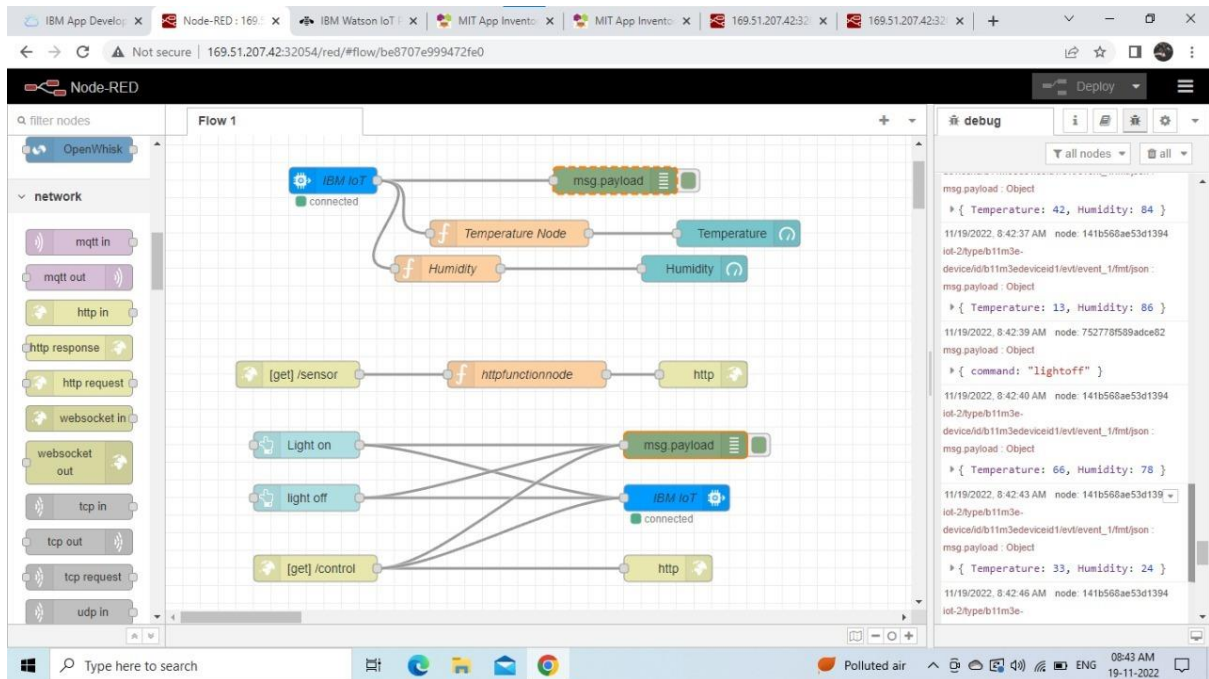
FLOW FOR LIGHT ON



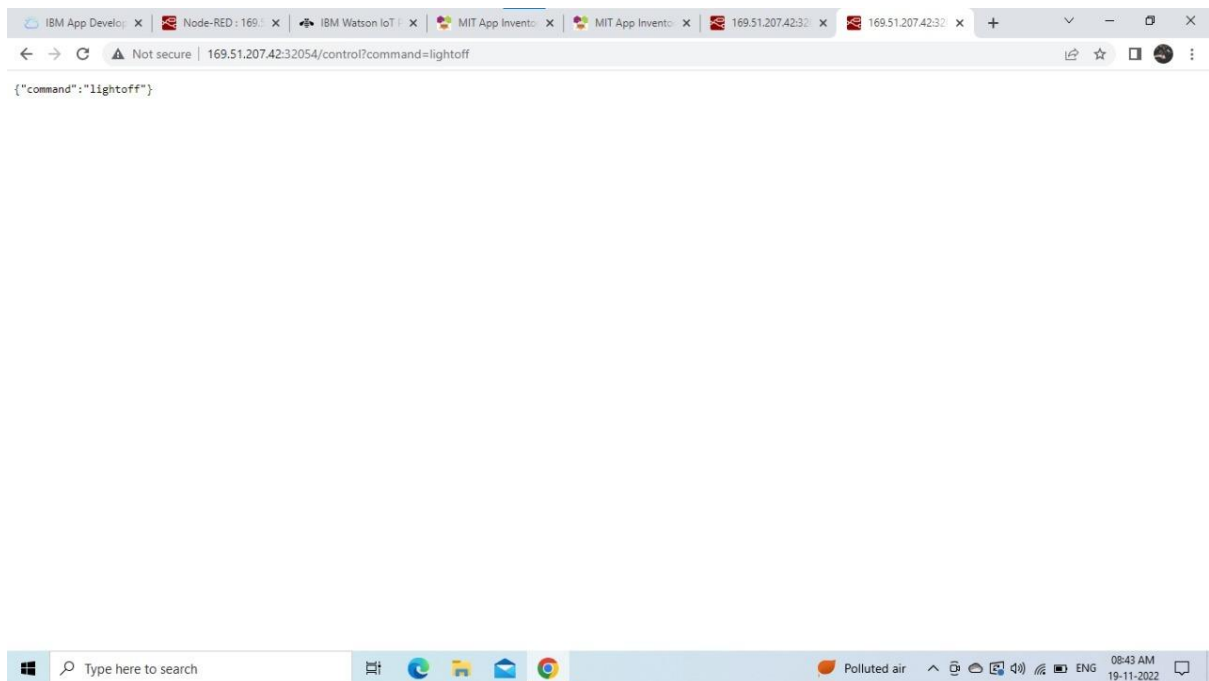
NODE-RED SENSOR OUTPUT



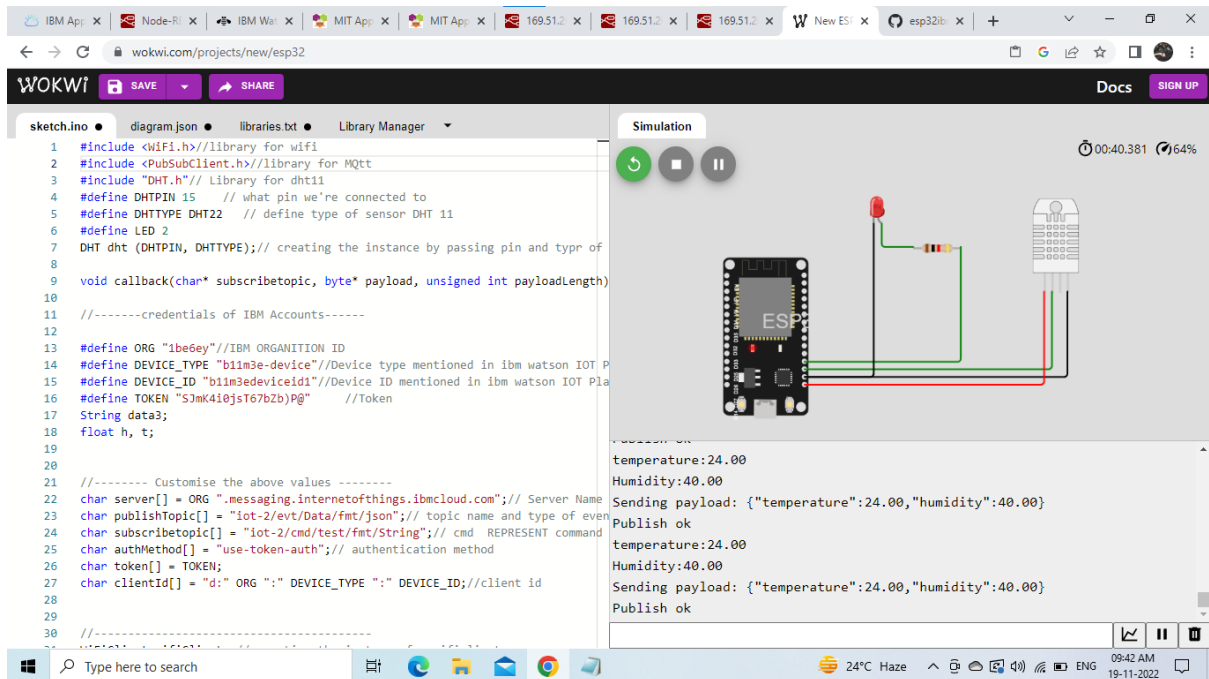
FLOW FOR LIGHT OFF



NODE-RED SENSOR OUTPUT



WOWKI SIMULATION



The screenshot shows the WOKWI web interface for simulating an ESP32 microcontroller. The sketch.ino file is open, displaying the following code:

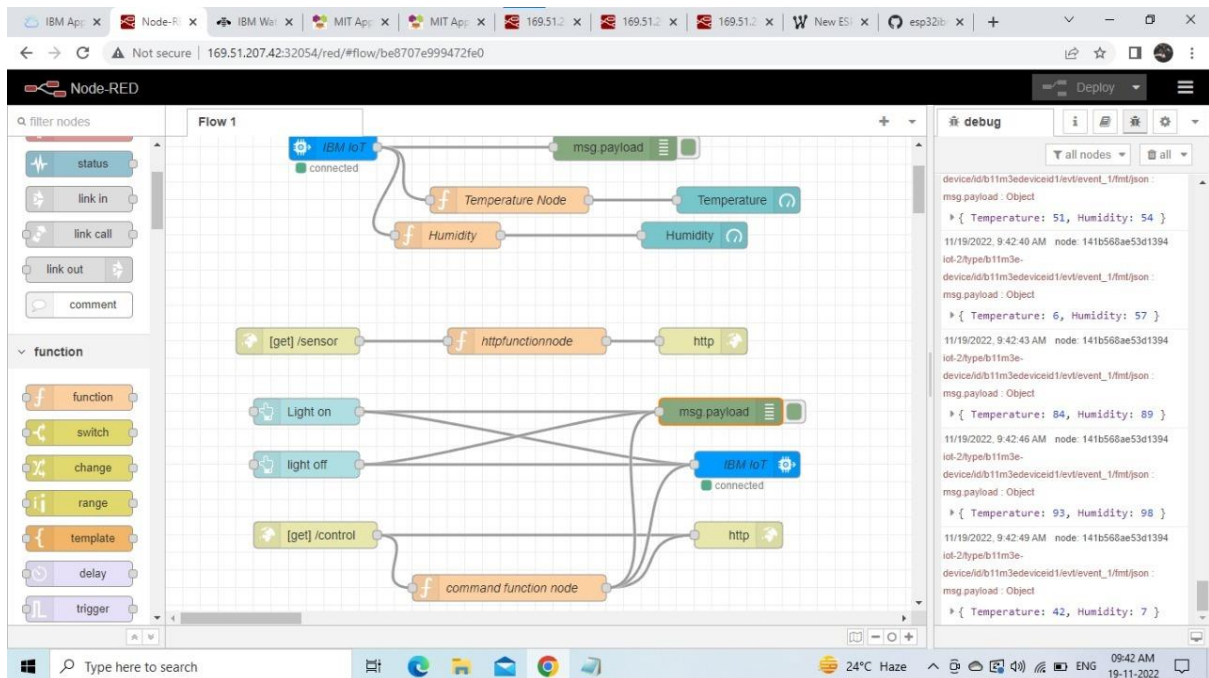
```
1 #include <WiFi.h> //library for wifi
2 #include <PubSubClient.h> //library for MQTT
3 #include "DHT.h" // Library for dht11
4 #define DHTPIN 15 // what pin we're connected to
5 #define DHTTYPE DHT22 // define type of sensor DHT 11
6 #define LED 2
7 DHT dht (DHTPIN, DHTTYPE); // creating the instance by passing pin and type of
8
9 void callback(char* topic, byte* payload, unsigned int payloadLength)
10
11 //-----credentials of IBM Accounts-----
12
13 #define ORG "1be6ey" //IBM ORGANIZATION ID
14 #define DEVICE_TYPE "b1m3e-device" //Device type mentioned in ibm watson IOT P
15 #define DEVICE_ID "b1m3edeviceld1" //Device ID mentioned in ibm watson IOT Pla
16 #define TOKEN "SJMk410jsT67b2b)Pg" //Token
17 String data3;
18 float h, t;
19
20
21 //-----Customize 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 even
24 char subscribTopic[] = "iot-2/cmd/test/fmt/String"; // cmd REPRESENT command
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 //-----
```

The simulation window on the right shows the ESP32 board connected to a DHT22 sensor. The sensor readings are displayed as follows:

temperature:24.00
Humidity:40.00

Sending payload: {"temperature":24.00,"humidity":40.00}
Publish ok
temperature:24.00
Humidity:40.00
Sending payload: {"temperature":24.00,"humidity":40.00}
Publish ok

NODE-RED FLOW



The screenshot shows the Node-RED web interface for creating a flow. The flow is titled "Flow 1" and includes the following nodes:

- IBM IoT (connected)
- msg payload
- Temperature Node
- Humidity
- [get]/sensor
- httpfunctionnode
- http
- Light on
- light off
- [get]/control
- command function node
- msg payload

The debug console on the right shows the data being received from the sensor:

```
device/Id/b1m3edeviceld1/evt/event_1/fmt/json :
msg.payload : Object
{
  Temperature: 51,
  Humidity: 54
}
11/19/2022, 9:42:40 AM node:141b568ae53d1394
iot-2/type/b1m3e-
device/Id/b1m3edeviceld1/evt/event_1/fmt/json :
msg.payload : Object
{
  Temperature: 6,
  Humidity: 57
}
11/19/2022, 9:42:43 AM node:141b568ae53d1394
iot-2/type/b1m3e-
device/Id/b1m3edeviceld1/evt/event_1/fmt/json :
msg.payload : Object
{
  Temperature: 84,
  Humidity: 89
}
11/19/2022, 9:42:46 AM node:141b568ae53d1394
iot-2/type/b1m3e-
device/Id/b1m3edeviceld1/evt/event_1/fmt/json :
msg.payload : Object
{
  Temperature: 93,
  Humidity: 96
}
11/19/2022, 9:42:49 AM node:141b568ae53d1394
iot-2/type/b1m3e-
device/Id/b1m3edeviceld1/evt/event_1/fmt/json :
msg.payload : Object
{
  Temperature: 42,
  Humidity: 7
}
```

OUTPUT IN IBM WATSON PLATFORM

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various platform features. The main content area is titled 'Recent Events' and shows a table of live data streams. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. It lists five events, each with a unique ID and a JSON value containing temperature and humidity data. The 'Format' column indicates that all events are in 'json' format, and the 'Last Received' column shows they were received 'a few seconds ago'. A status message at the bottom right of the table area states '1 Simulation running'. The bottom of the screen shows a Windows taskbar with the search bar, application icons, and system tray information including the date and time.

Event	Value	Format	Last Received
event_1	{"Temperature":58,"Humidity":63}	json	a few seconds ago
event_1	{"Temperature":33,"Humidity":67}	json	a few seconds ago
event_1	{"Temperature":100,"Humidity":50}	json	a few seconds ago
event_1	{"Temperature":13,"Humidity":14}	json	a few seconds ago
event_1	{"Temperature":3,"Humidity":43}	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

1 Simulation running

09:43 AM 19-11-2022