Develop A Web application using Node-RED Service

Develop the web application using Node Red

Team ID	PNT2022TMID44448
Project Name	Project Name – Signs with Smart Connectivity for better road safety
Mentor	D.Nivethini

Requirements:

Software:

1. Arduino IDE

2. IBM cloud Account

3. IBM IOT Platform

4. Node Red

5. MIT App Inventor

Hardware

1. Node MCU ESP8266 2.
2. DHT11 - To measure the Temperature and Humidity
3. LED’S - To Turn ON/OFF the light

Activities :

1. Create a device in IBM Watson IoT platform

2. Create Node-red application

3. Install DHT, PubSub libraries

4. Develop the code snippet for sending the indoor weather parameters to the cloud and receiving the commands for controlling

5. Develop the code snippet for connecting Node MCU to the Wi-Fi

6. Connecting the sensor and LEDs to the microcontroller

7. Create the Web UI to visualize the indoor weather parameters

8. Create a mobile application visualizing the sensor reading and buttons to control the LED

1. Create a device in IBM Watson IOT platform

https://w5704q.internetofthings.ibmcloud.com/dashboard/security

2. Create Node-red application

https://node-red-mhjxs-2022-11-06.eu gb.mybluemix.net/red/#flow/1a625650c137b576

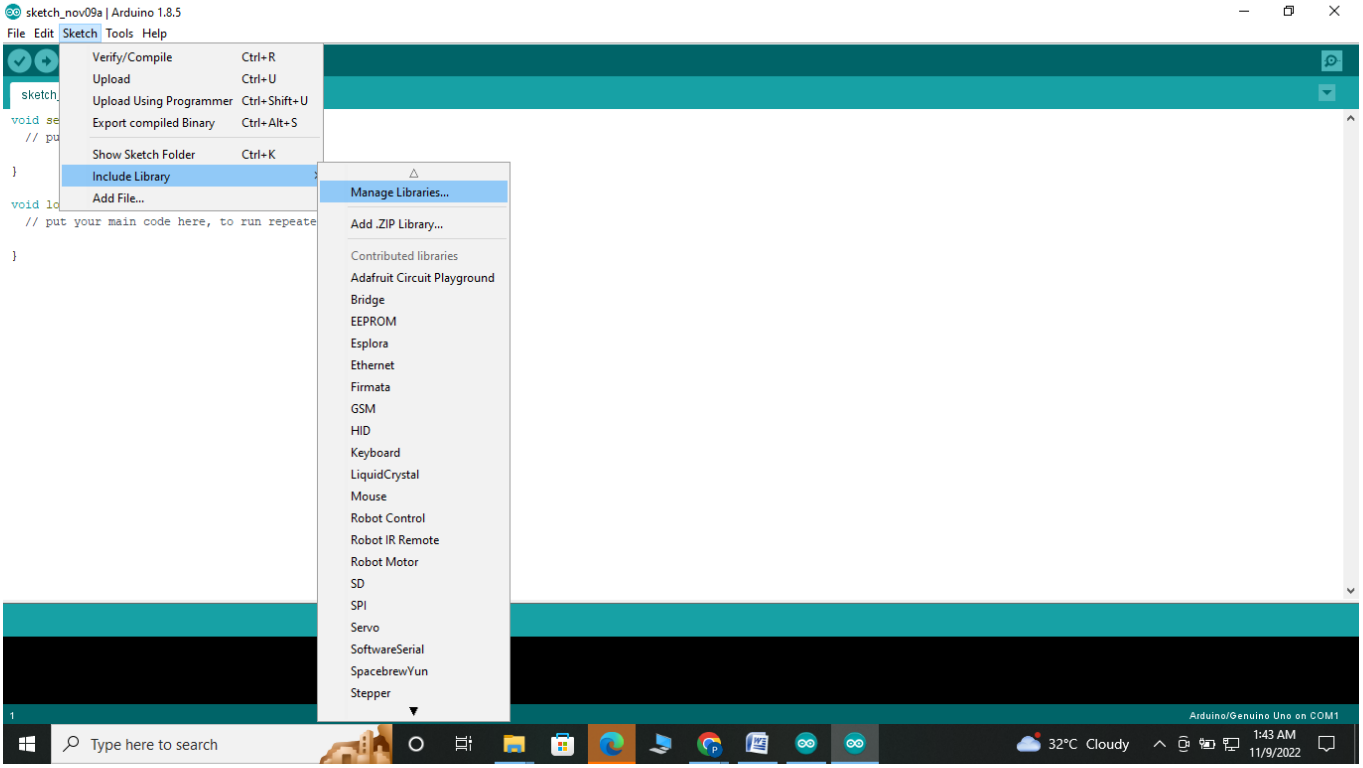
4. Install the libraries

a. DHT library

https://github.com/adafruit/DHT-sensor-library

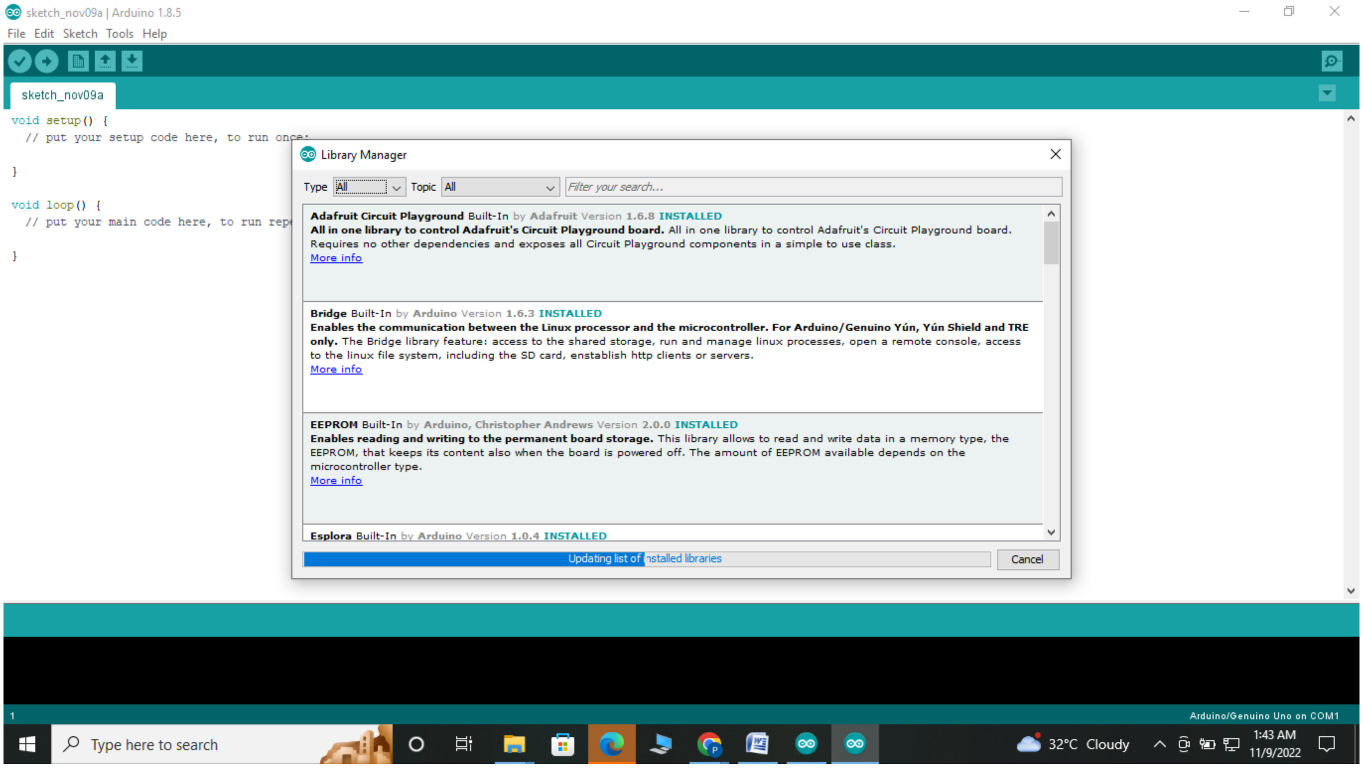
Download the library from above link and you can include zip file in Include library directly or follow the below steps to include library

Step 1: Open Arduino IDE. Click on Sketch -> Include Library -> Manage Libraries



```
void se
  // pu

}

void lo
  // put your main code here, to run repeate

}
```

Step 2: A pop-up library manager window appears. Search for DHT library by typing "dht" in the search bar



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repe

}
```

Adafruit Circuit Playground Built-In by **Adafruit** Version **1.6.8** **INSTALLED**
All in one library to control Adafruit's Circuit Playground board. All in one library to control Adafruit's Circuit Playground board. Requires no other dependencies and exposes all Circuit Playground components in a simple to use class.
More info

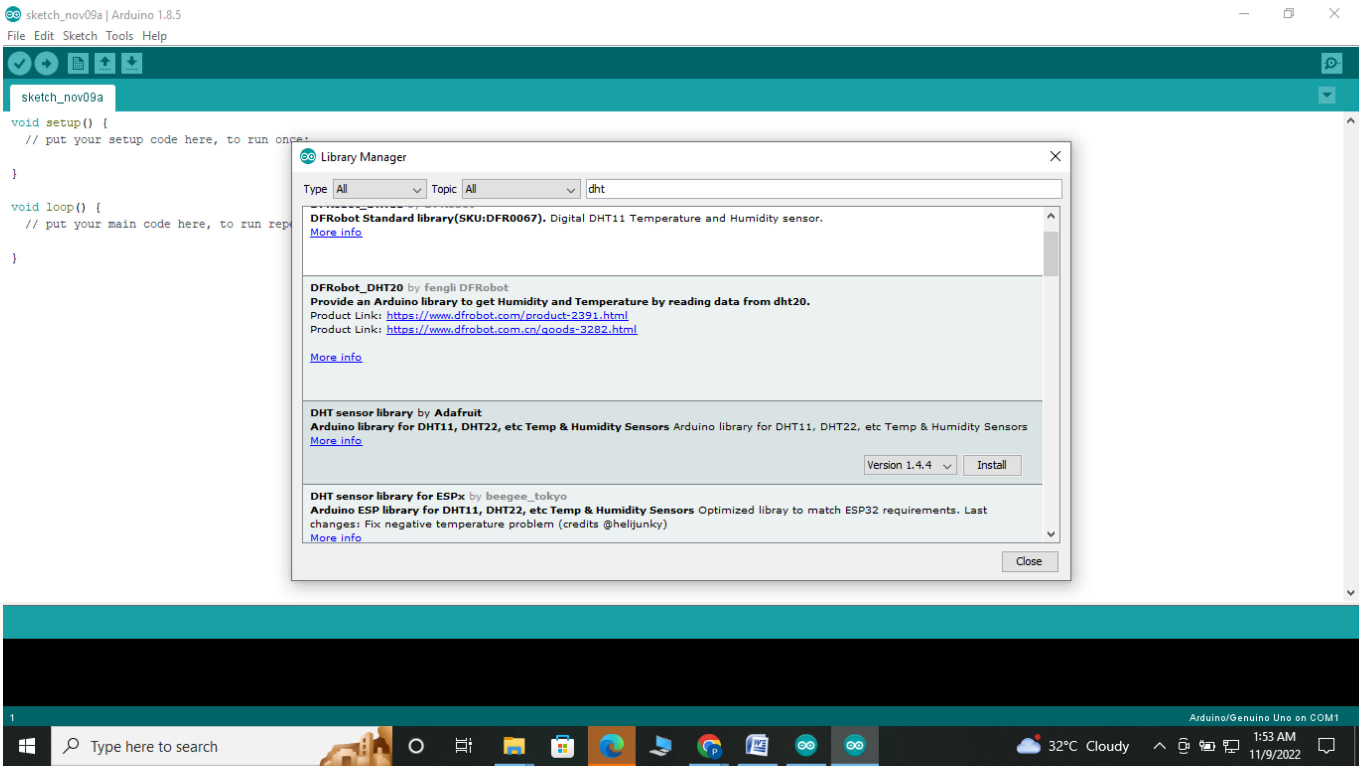
Bridge Built-In by **Arduino** Version **1.6.3** **INSTALLED**
Enables the communication between the Linux processor and the microcontroller. For Arduino/Genuino Yún, Yún Shield and TRE only. The Bridge library feature: access to the shared storage, run and manage linux processes, open a remote console, access to the linux file system, including the SD card, enstablish http clients or servers.
More info

EEPROM Built-In by **Arduino, Christopher Andrews** Version **2.0.0** **INSTALLED**
Enables reading and writing to the permanent board storage. This library allows to read and write data in a memory type, the EEPROM, that keeps its content also when the board is powered off. The amount of EEPROM available depends on the microcontroller type.
More info

Esplora Built-In by **Arduino** Version **1.0.4** **INSTALLED**

Updating list of installed libraries

Step 3: Click on dht sensor library and click install to include dht library in your Arduino IDE**.**



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run rep
}
```

dht

DFRobot Standard library(SKU:DFR0067). Digital DHT11 Temperature and Humidity sensor.
More info

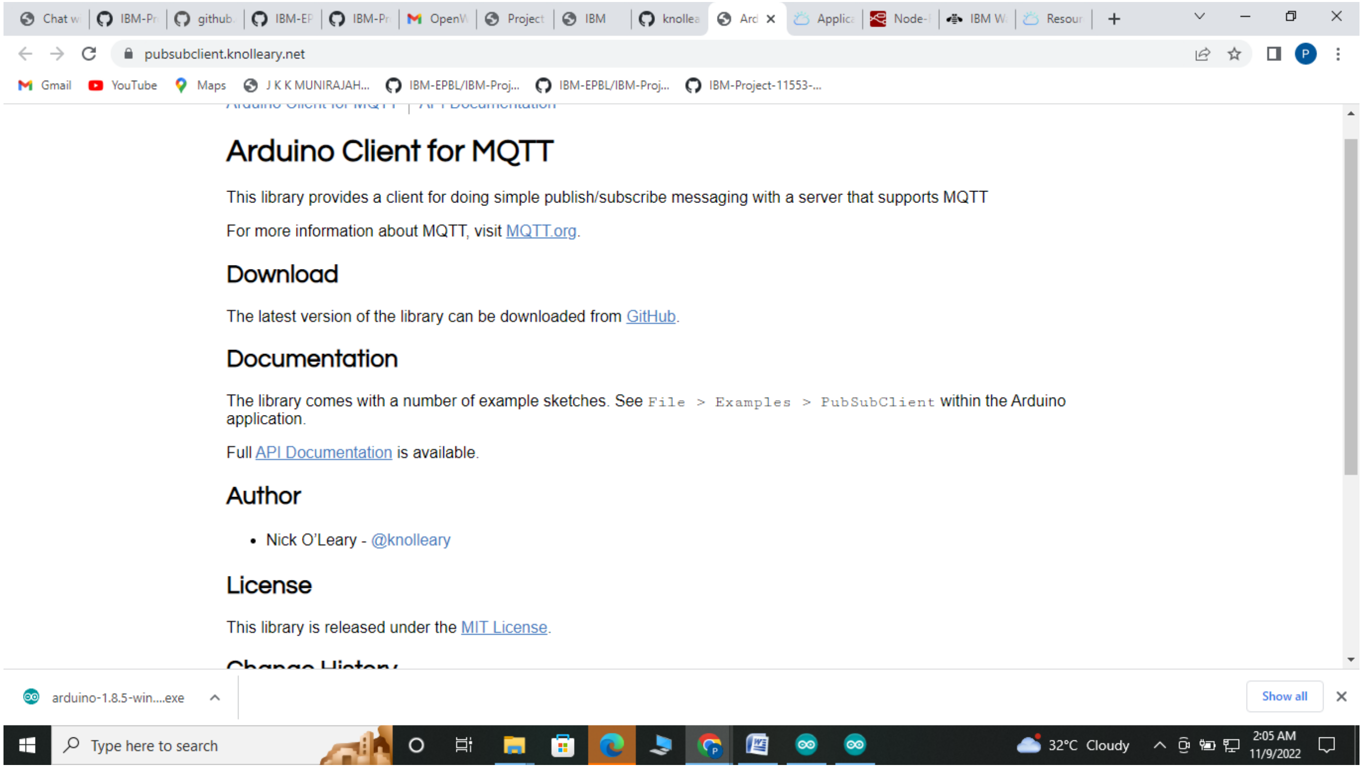
DFRobot_DHT20 by fengli DFRobot
Provide an Arduino library to get Humidity and Temperature by reading data from dht20.
Product Link: https://www.dfrobot.com/product-2391.html
Product Link: https://www.dfrobot.com.cn/goods-3282.html
More info

DHT sensor library by **Adafruit**
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
More info

DHT sensor library for ESPx by beegee_tokyo
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last changes: Fix negative temperature problem (credits @helijunky)
More info

b. Pubsub library for communicating with the cloud
https://github.com/knolleary/pubsubclient
Download the Library file from the above link and add zip file in include library from tools menu in Arduino IDE

Step 1: Download the zip file



Arduino Client for MQTT

This library provides a client for doing simple publish/subscribe messaging with a server that supports MQTT

For more information about MQTT, visit MQTT.org.

Download

The latest version of the library can be downloaded from GitHub.

Documentation

The library comes with a number of example sketches. See `File > Examples > PubSubClient` within the Arduino application.

Full API Documentation is available.

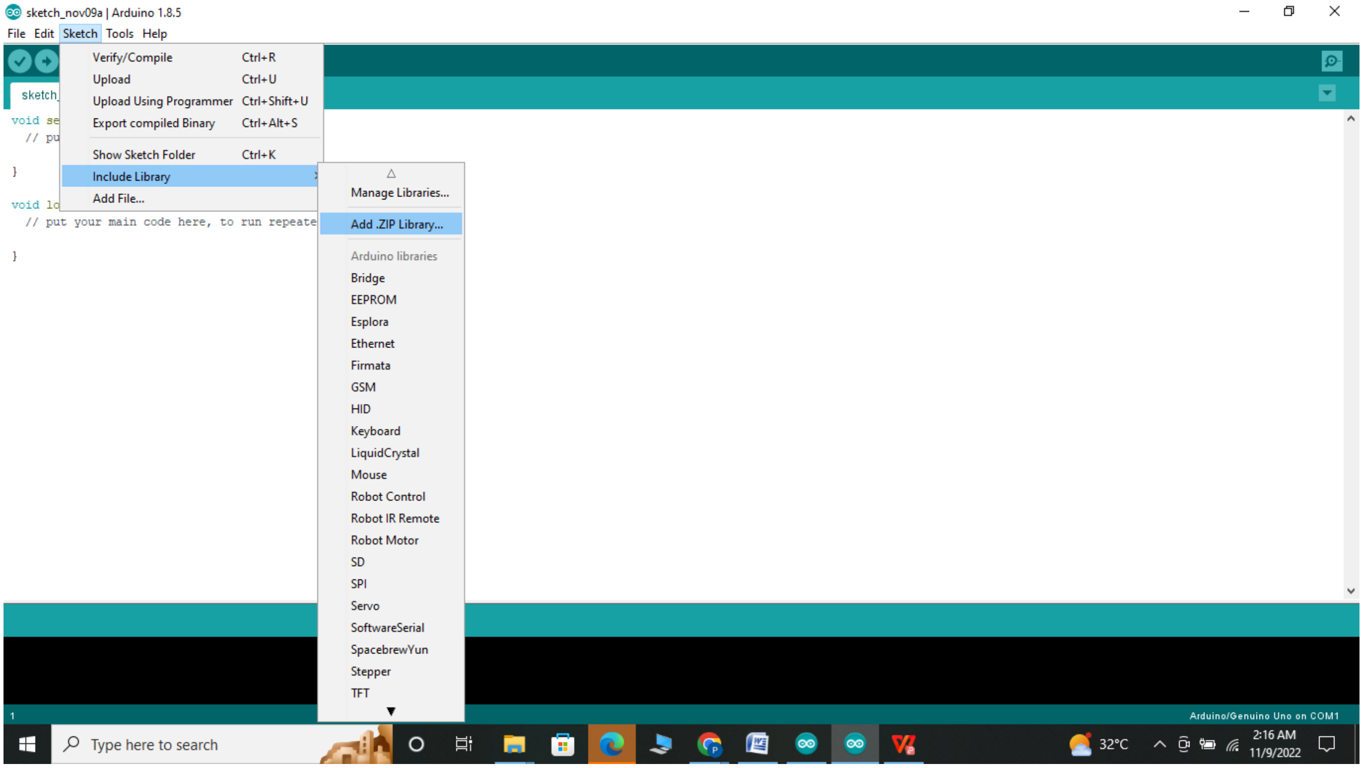
Author

- Nick O'Leary - @knolleary

License

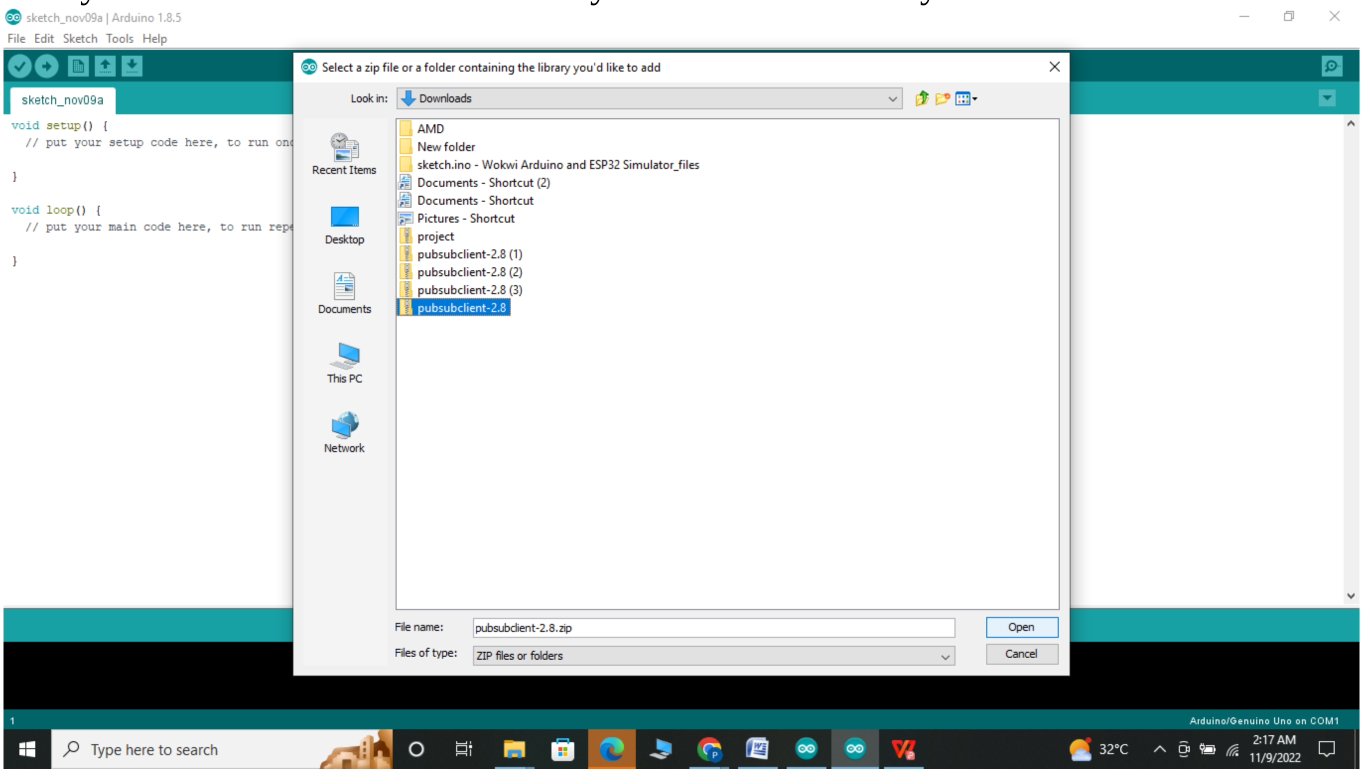
This library is released under the MIT License.

Step 2: Include zip folder into Arduino IDE. Click on Sketch -> Include Library -> Add .Zip library



```
void se
  // pu

}

void lo
  // put your main code here, to run repeate

}
```

Step 3: Select the file or folder containing the library to add. Click on open to include the library into Arduino IDE. Pubsub library has been successfully added



```
void setup() {
  // put your setup code here, to run onc

}

void loop() {
  // put your main code here, to run repe

}
```

Select a zip file or a folder containing the library you'd like to add

Look in: Downloads

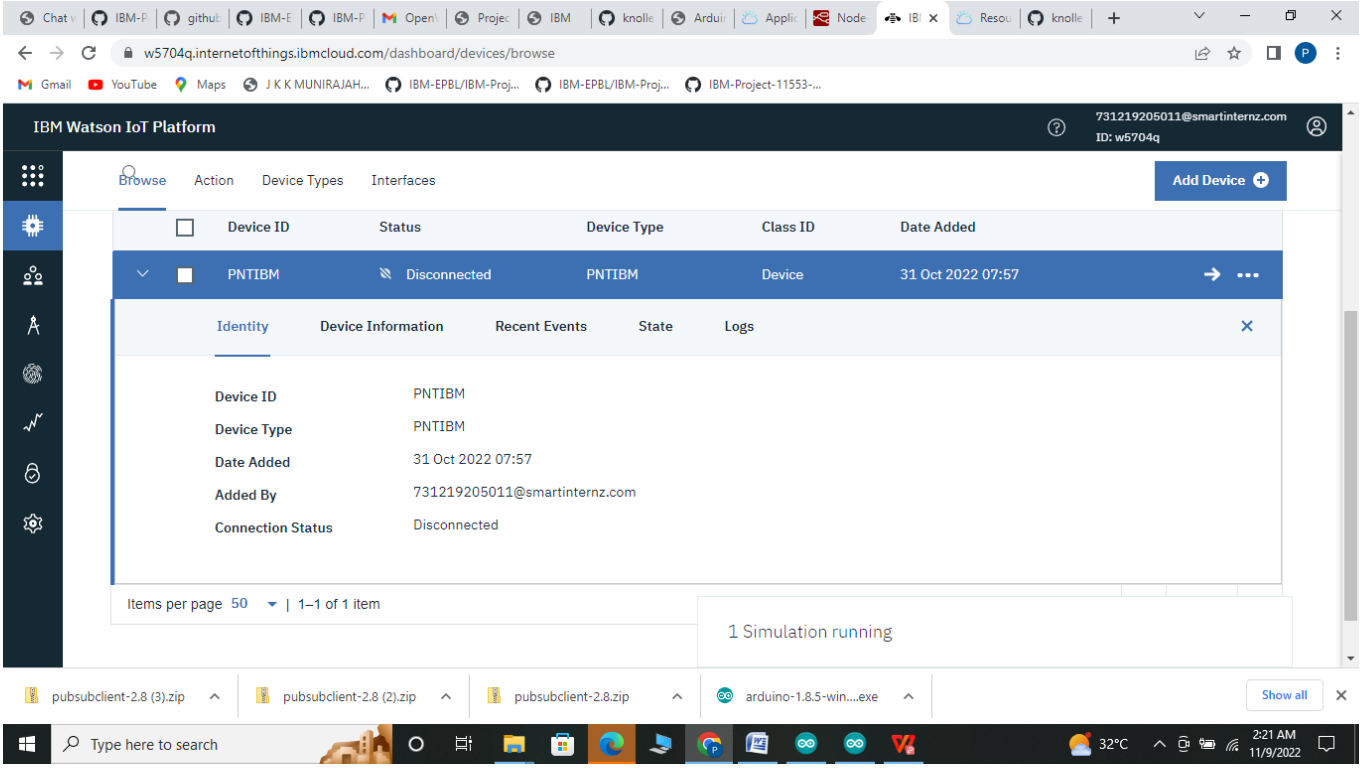
- AMD
- New folder
- sketch.ino - Wokwi Arduino and ESP32 Simulator_files
- Documents - Shortcut (2)
- Documents - Shortcut
- Pictures - Shortcut
- project
- pubsubclient-2.8 (1)
- pubsubclient-2.8 (2)
- pubsubclient-2.8 (3)
- pubsubclient-2.8

File name: pubsubclient-2.8.zip

Files of type: ZIP files or folders

4.Develop the code snippet for connecting Node MCU to the Wi-Fi

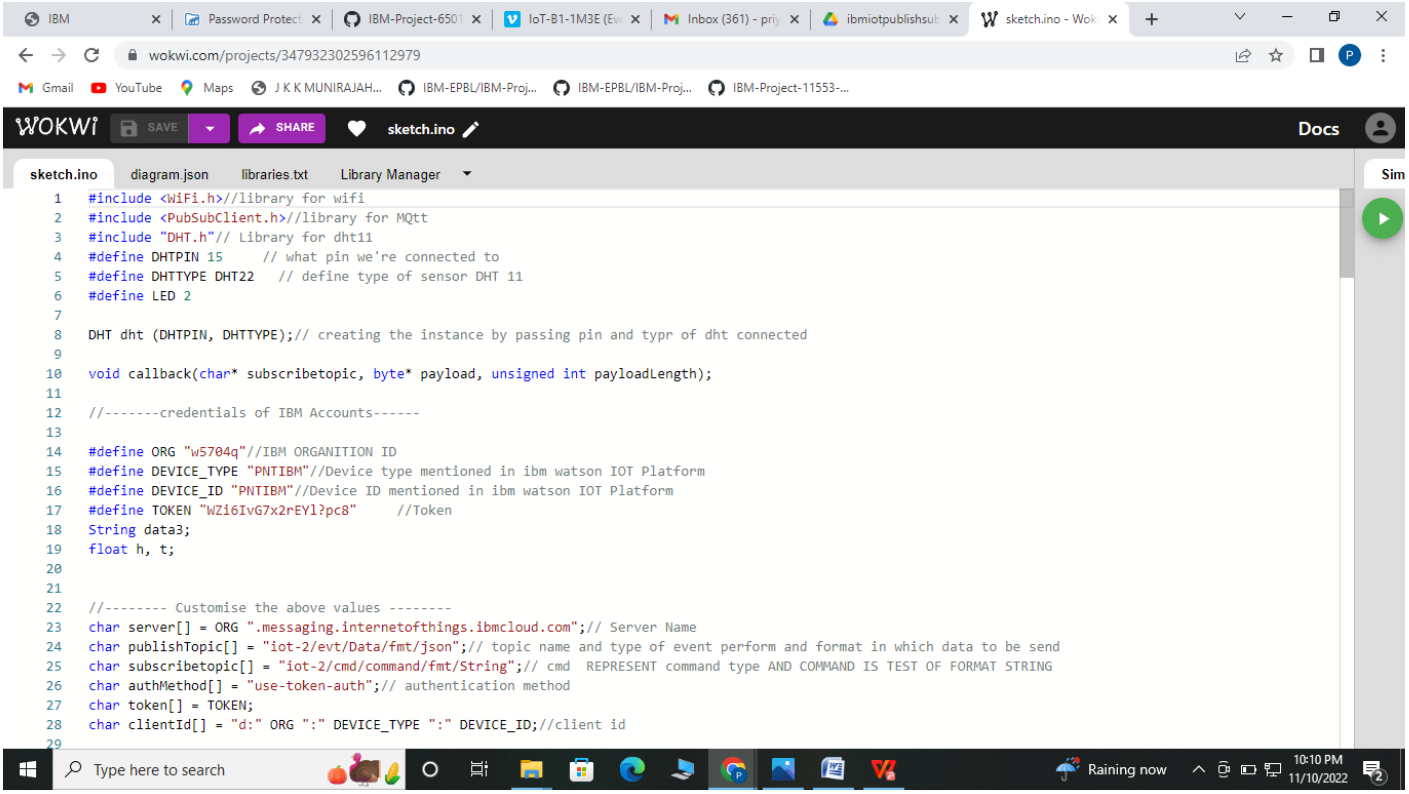
Step 1: Copy your Organization ID, Device Type, Device ID, Authentication Type, Authentication
Token to clipboard from IBM Watson IBM IoT Platform while creating a device.



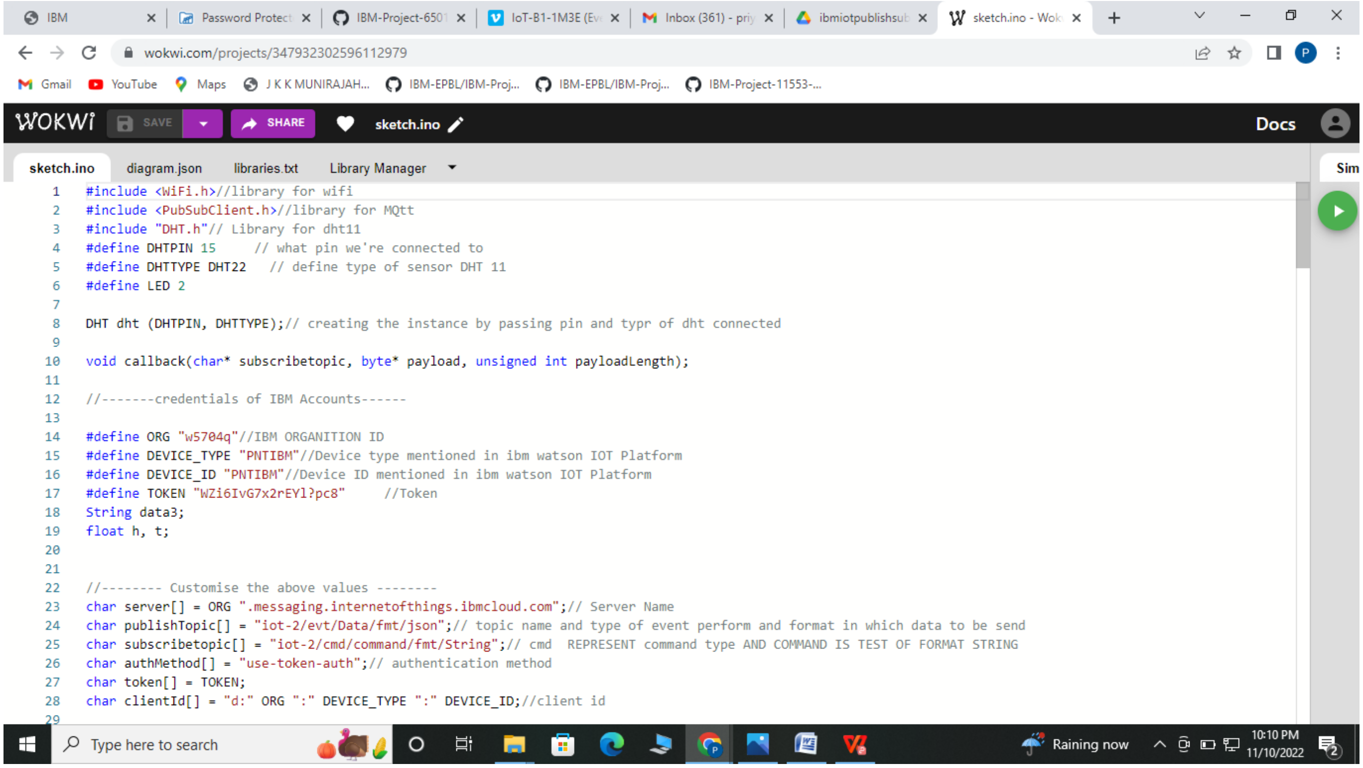
☐	Device ID	Status	Device Type	Class ID	Date Added
☐	PNTIBM	Disconnected	PNTIBM	Device	31 Oct 2022 07:57

Device ID	PNTIBM
Device Type	PNTIBM
Date Added	31 Oct 2022 07:57
Added By	731219205011@smartinternz.com
Connection Status	Disconnected

Step2: Edit your Wi-Fi credentials ,ORG ID, device type, device id, authentication type, authentication token in the code .

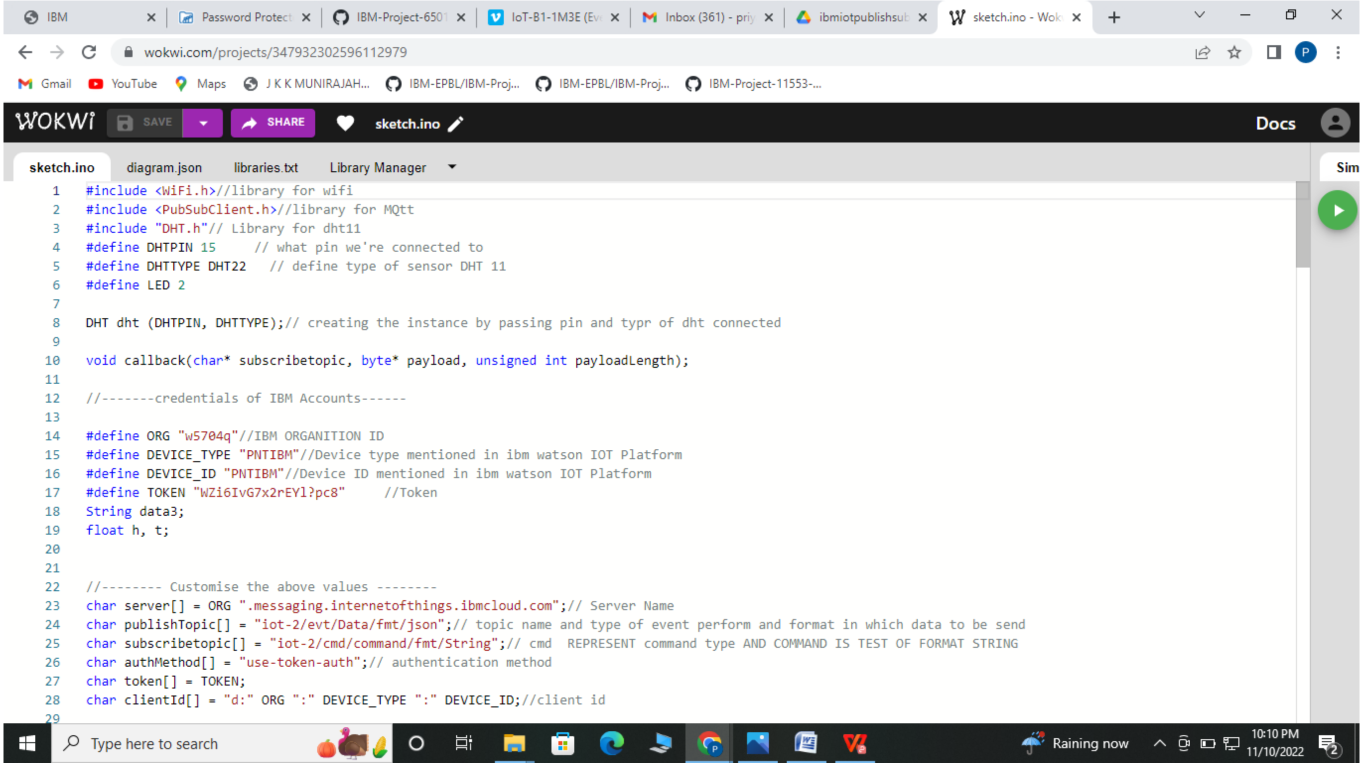


```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "w5704q"//IBM ORGANITION ID
#define DEVICE_TYPE "PNTIBM"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "PNTIBM"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "WZi6IvG7x2rEYl?pc8"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
```

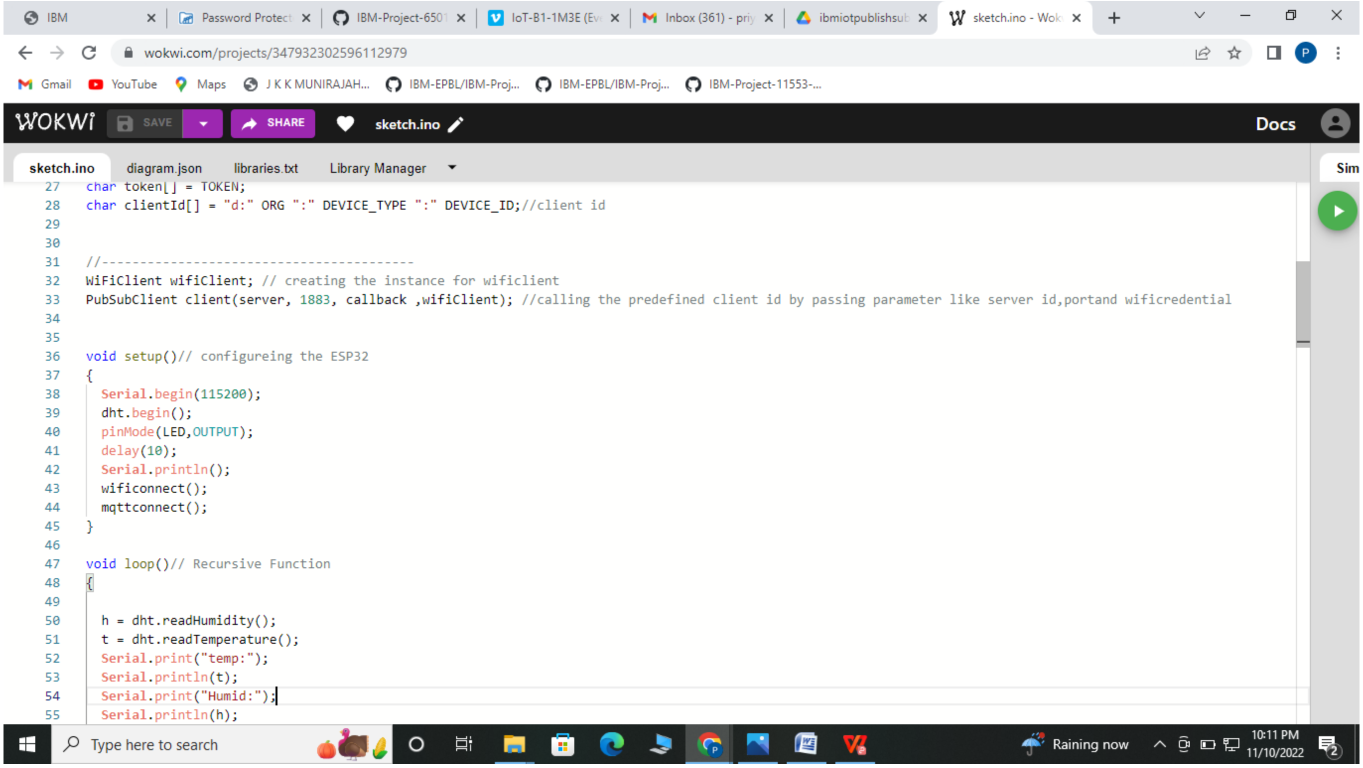


```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "w5704q"//IBM ORGANITION ID
#define DEVICE_TYPE "PNTIBM"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "PNTIBM"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "WZi6IvG7x2rEYl?pc8"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
```

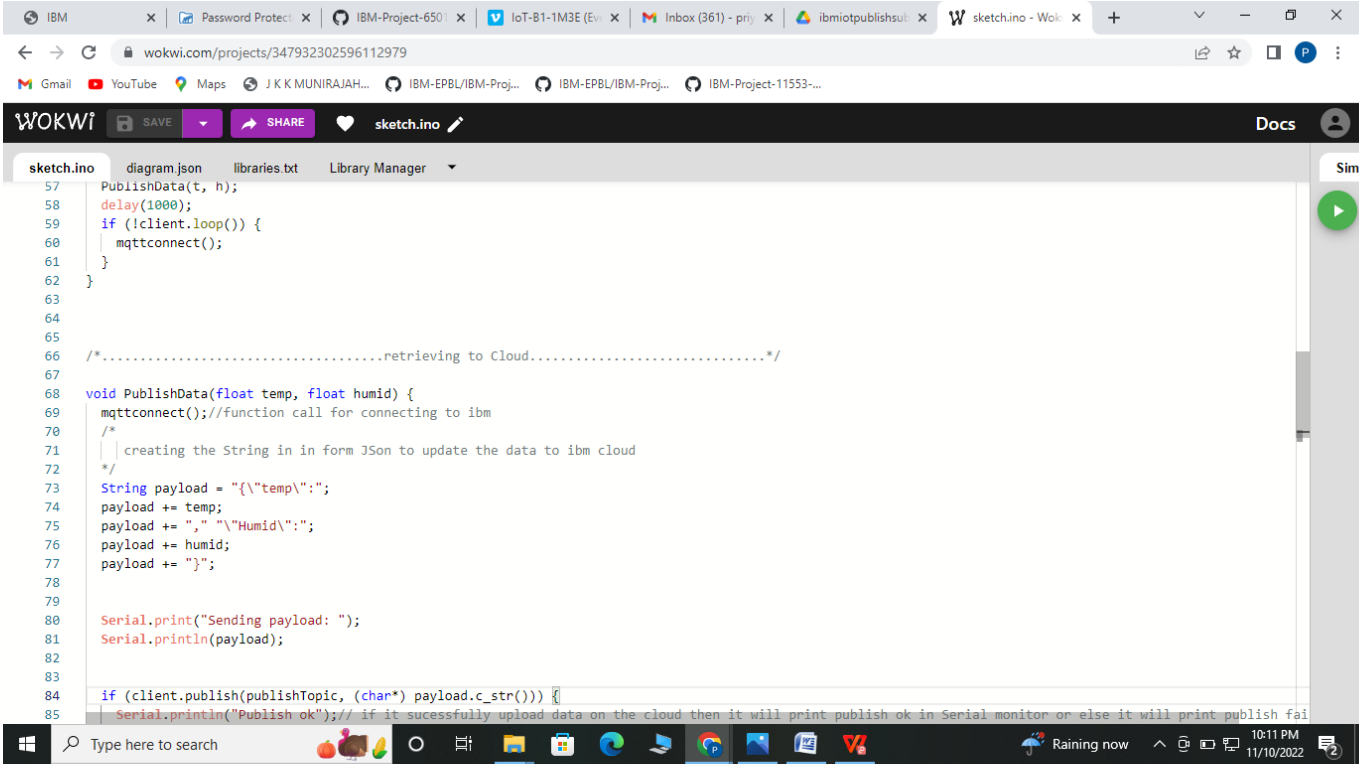
Keep note of the underline data for further reference as we need to update them.



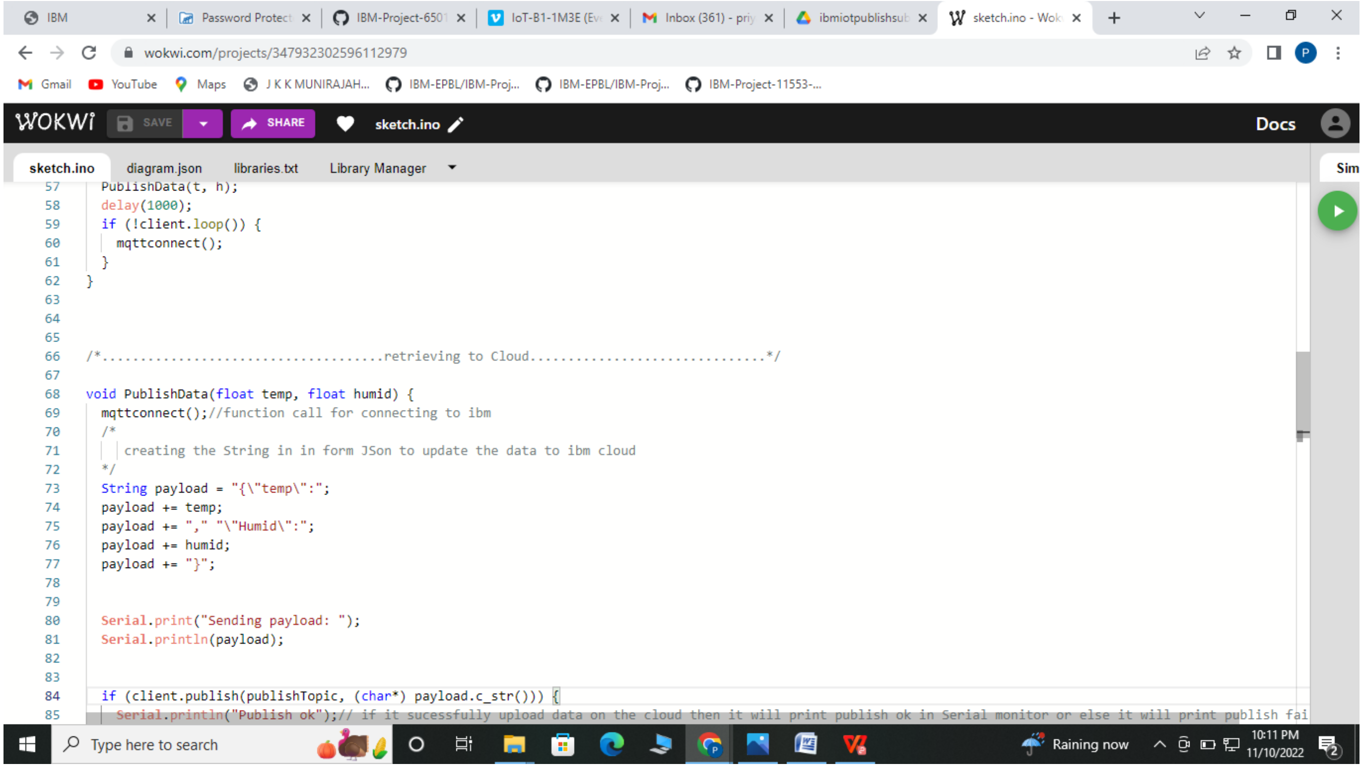
```
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN 15     // what pin we're connected to
#define DHTTYPE DHT22   // define type of sensor DHT 11
#define LED 2

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "w5704q"//IBM ORGANITION ID
#define DEVICE_TYPE "PNTIBM"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "PNTIBM"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "WZi6IvG7x2rEYl?pc8"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id
```



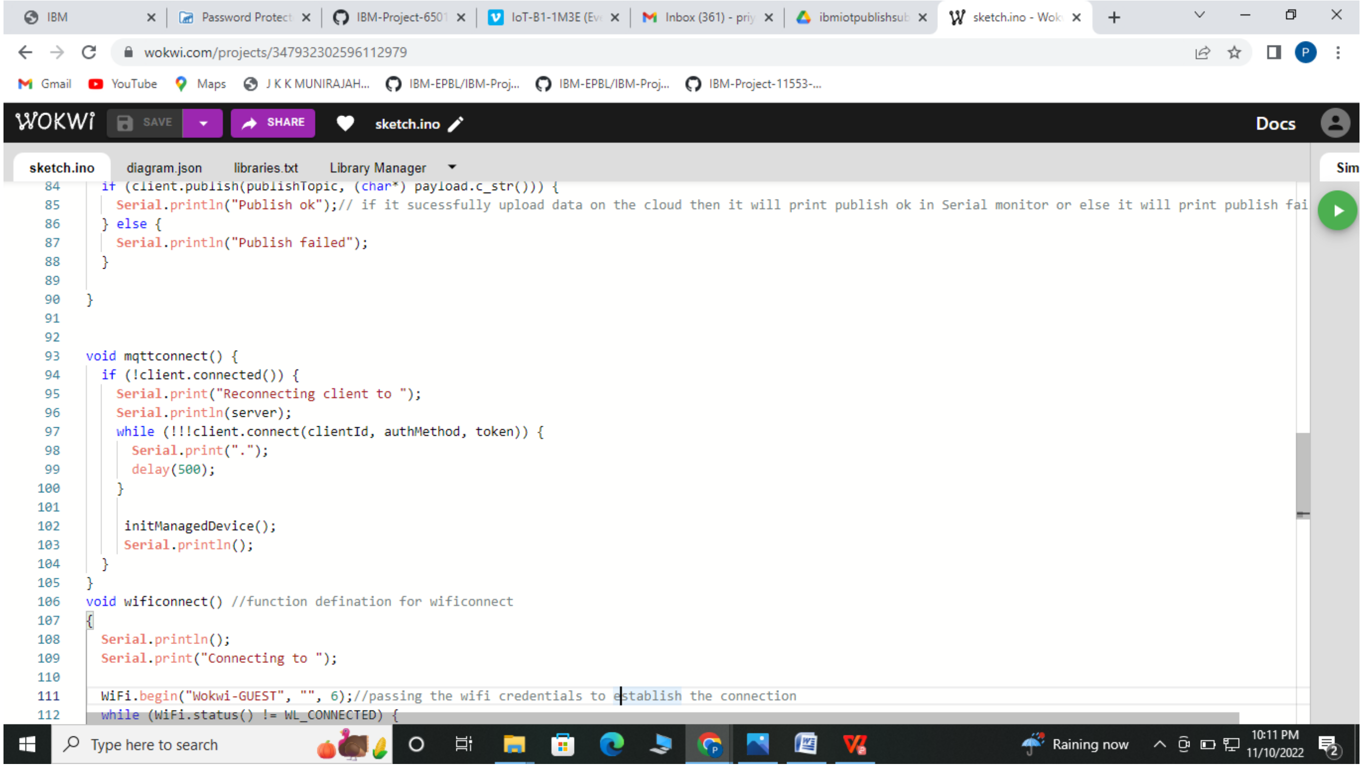
```
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//-----------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);
```



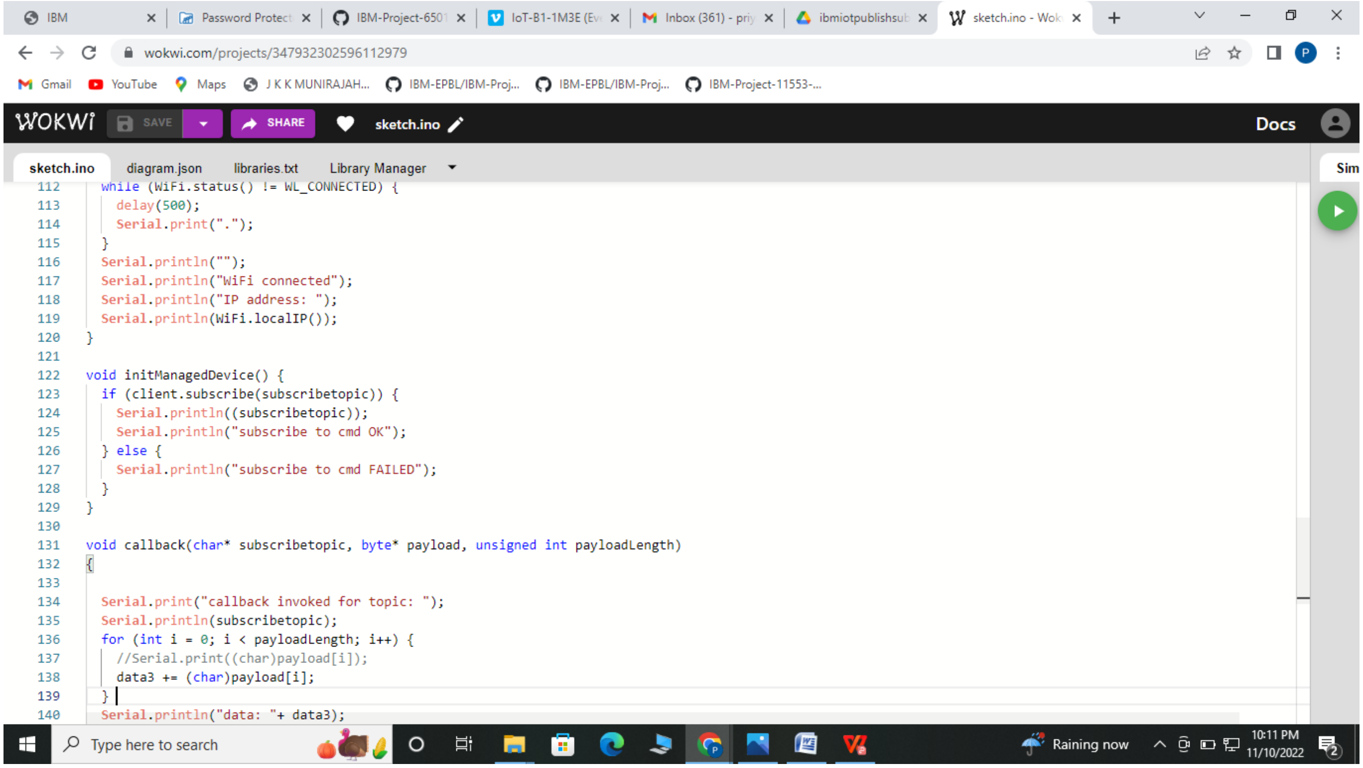
```
  PublishData(t, h);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*....................................retrieving to Cloud...............................*/

void PublishData(float temp, float humid) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"Humid\":";
  payload += humid;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish fai
```



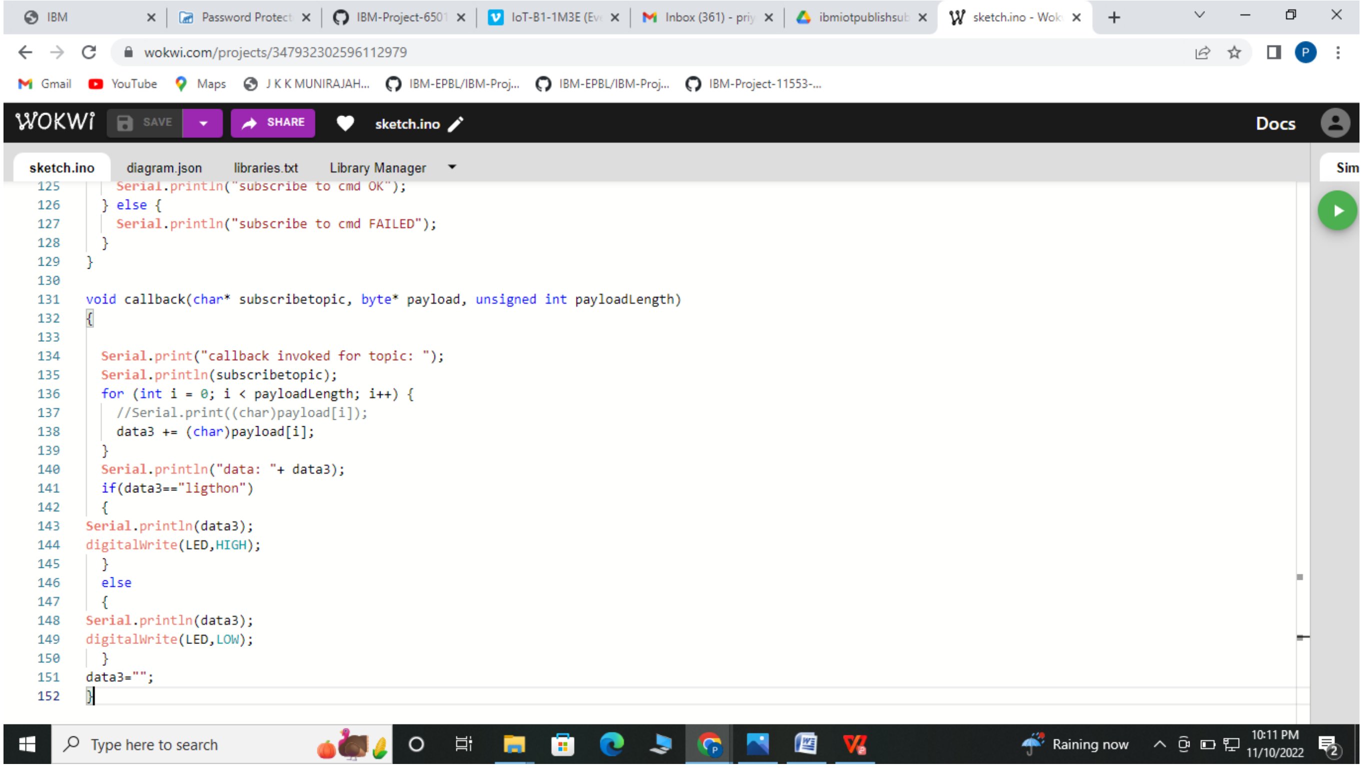
```
  PublishData(t, h);
  delay(1000);
  if (!client.loop()) {
    mqttconnect();
  }
}



/*...................................retrieving to Cloud..............................*/

void PublishData(float temp, float humid) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"Humid\":";
  payload += humid;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish fai
```



```
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will print publish ok in Serial monitor or else it will print publish fai
  } else {
    Serial.println("Publish failed");
  }

}


void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the connection
  while (WiFi.status() != WL_CONNECTED) {
```

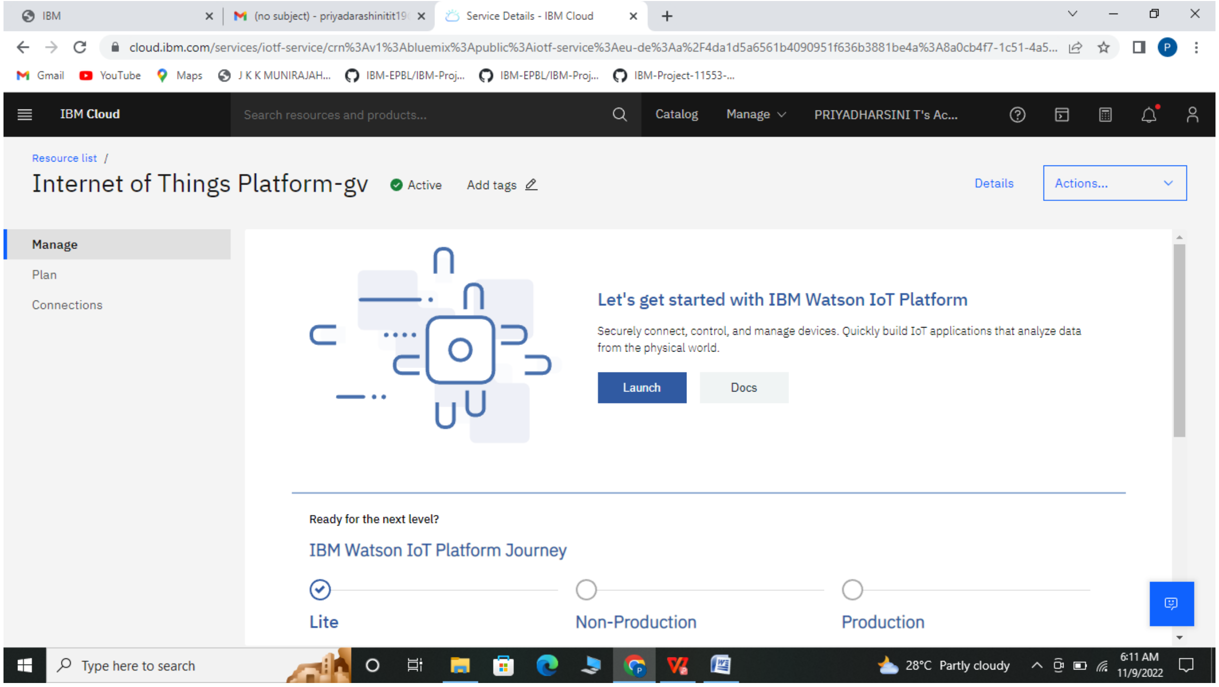


```
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
```



```
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="ligthon")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else
  {
Serial.println(data3);
digitalWrite(LED,LOW);
  }
data3="";
}
```

Step3: Compile the code and check for errors



Internet of Things Platform-gv

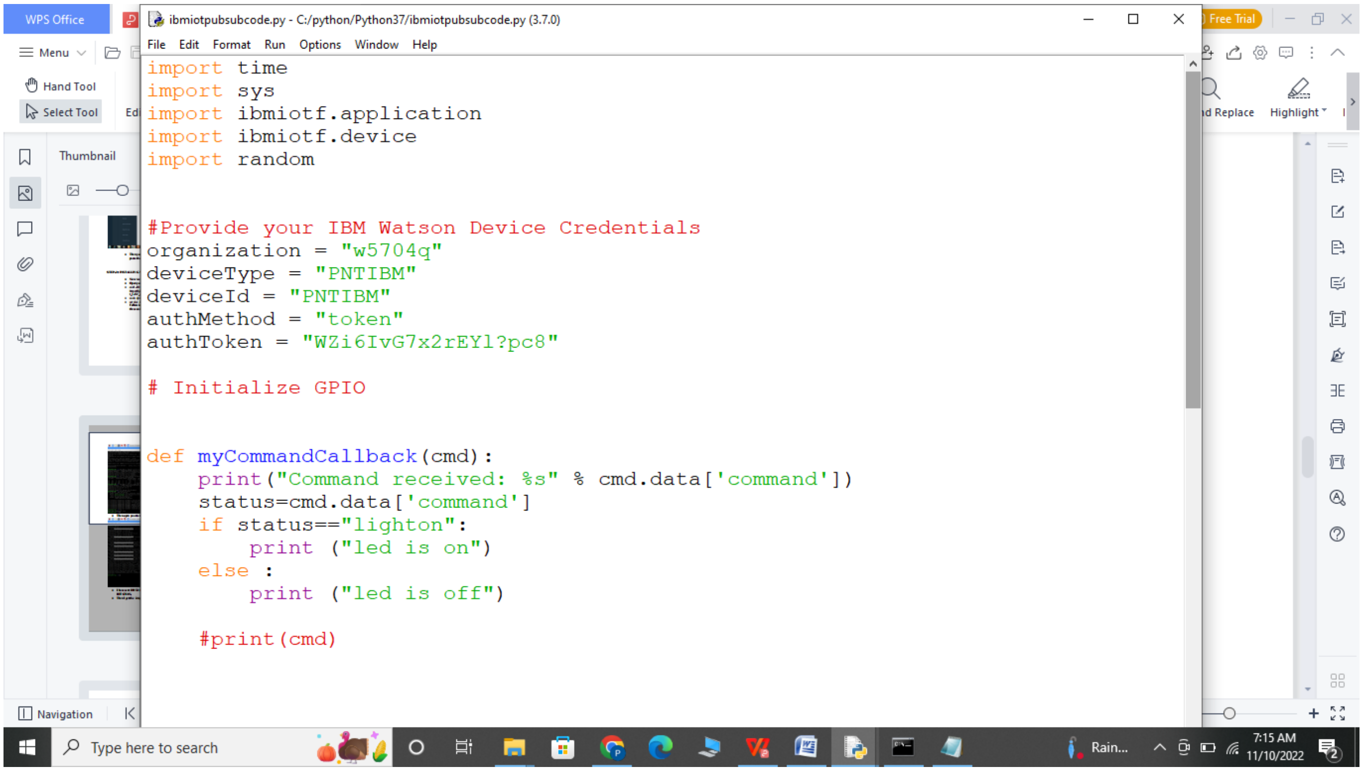
Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Ready for the next level?

IBM Watson IoT Platform Journey

Step 4: Upload the code to NodeMCU by selecting the NodeMCU 1.0 and port from tools menu



ibmiotpubsubcode.py - C:/python/Python37/ibmiotpubsubcode.py (3.7.0)

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


#Provide your IBM Watson Device Credentials
organization = "w5704q"
deviceType = "PNTIBM"
deviceId = "PNTIBM"
authMethod = "token"
authToken = "WZi6IvG7x2rEYl?pc8"

# Initialize GPIO


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    else :
        print ("led is off")

    #print(cmd)
```

Step 5: Connecting the sensor and LEDs to the microcontroller

o Connect VCC of basic shield to 3.3V of NodeMCU (Input Power Supply)

o Connect GND of Basic shield to GND of NodeMCU

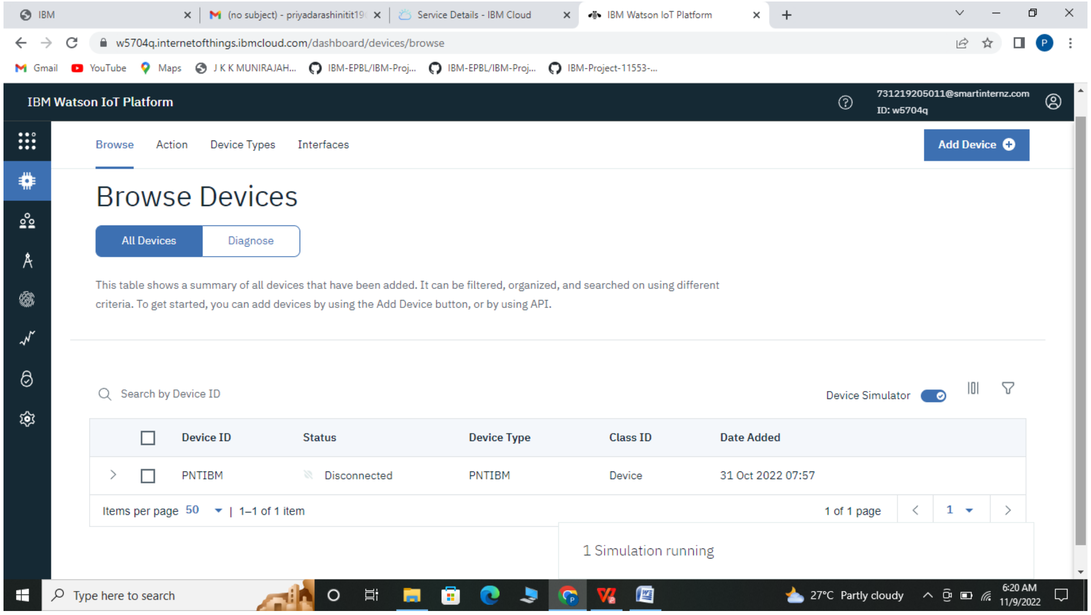
o Connect LED (L1) of basic shield to digital pin(D0) of NodeMCU

o Connect LED (L2) of basic shield to digital pin(D1) of NodeMCU

o Connect VCC of DHT sensorto 3.3V of NodeMCU

o Connect GND of DHT sensor to GND of NodeMCU

o Connect Data pin of DHT sensor to digital pin(D2) of NodeMCU



Browse Devices

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.

	Device ID	Status	Device Type	Class ID	Date Added
□	PNTIBM	Disconnected	PNTIBM	Device	31 Oct 2022 07:57

Items per page 50 | 1–1 of 1 item

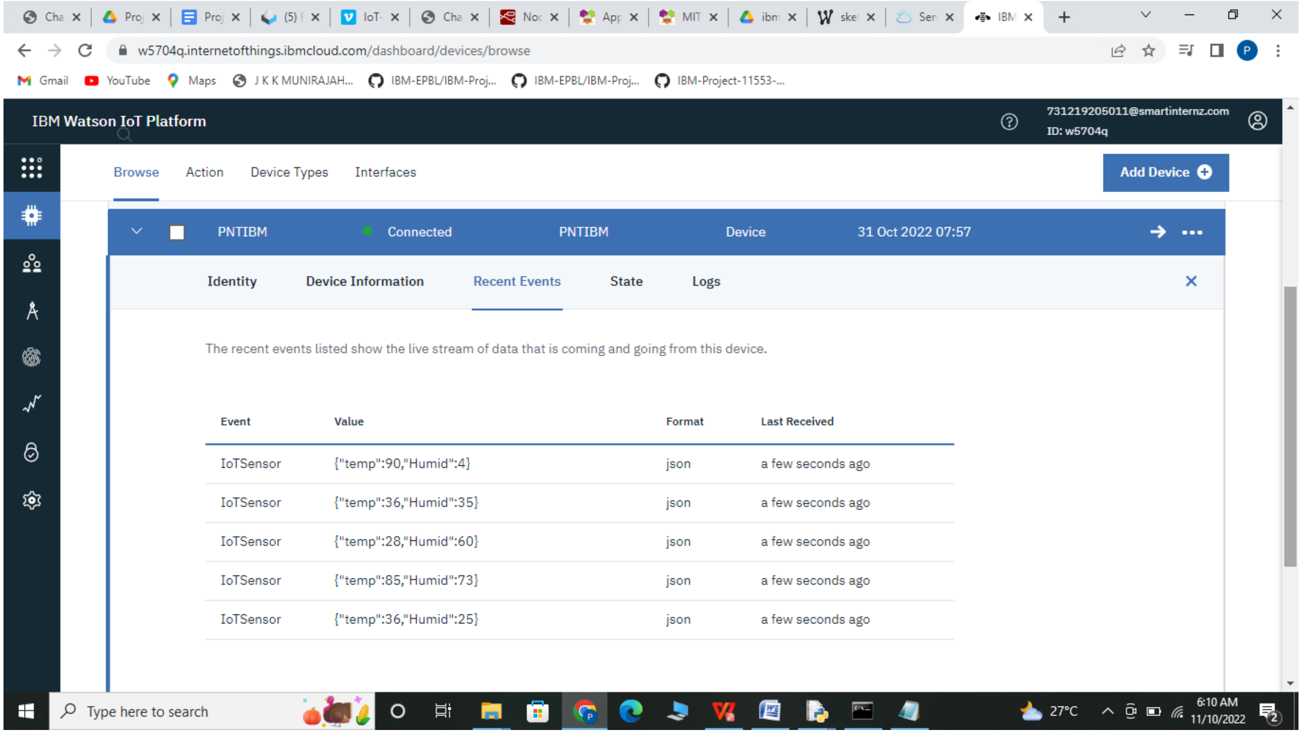
1 of 1 page

1 Simulation running

Step 6: Create the Web UI to visualize the indoor weather parameters

TASK 1: Connecting device to cloud to see the data in the cards section

Step 1: Login to your IBM cloud account and click on services.

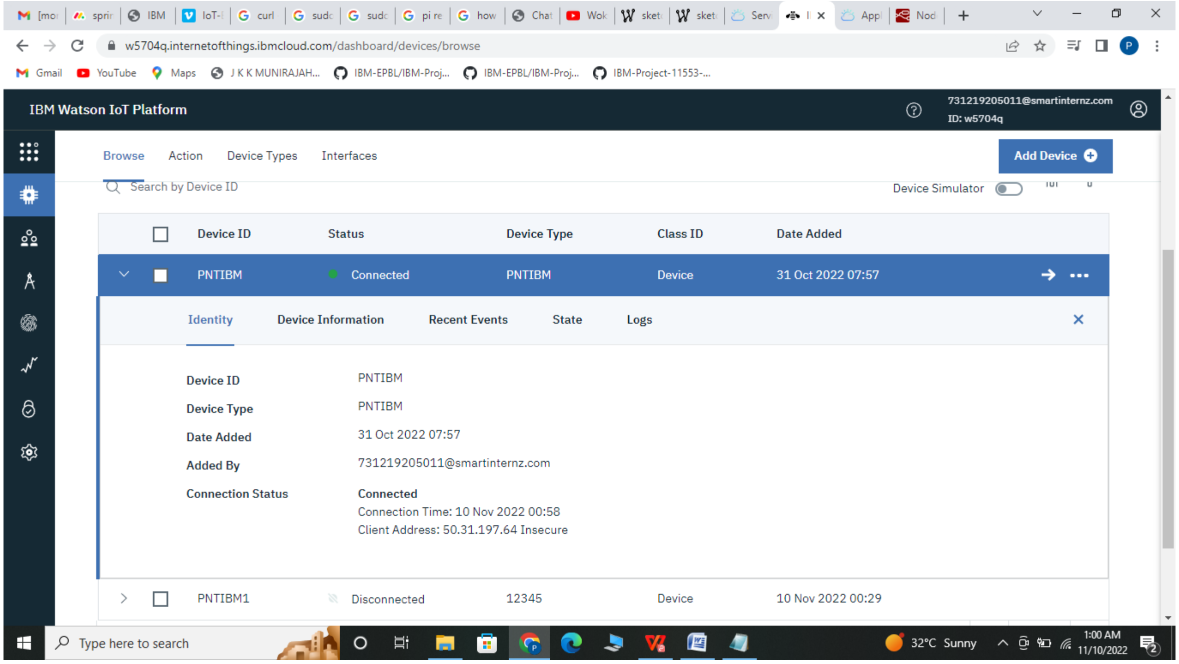


PNTIBM	Connected	PNTIBM	Device	31 Oct 2022 07:57

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
IoTSensor	{"temp":90,"Humid":4}	json	a few seconds ago
IoTSensor	{"temp":36,"Humid":35}	json	a few seconds ago
IoTSensor	{"temp":28,"Humid":60}	json	a few seconds ago
IoTSensor	{"temp":85,"Humid":73}	json	a few seconds ago
IoTSensor	{"temp":36,"Humid":25}	json	a few seconds ago

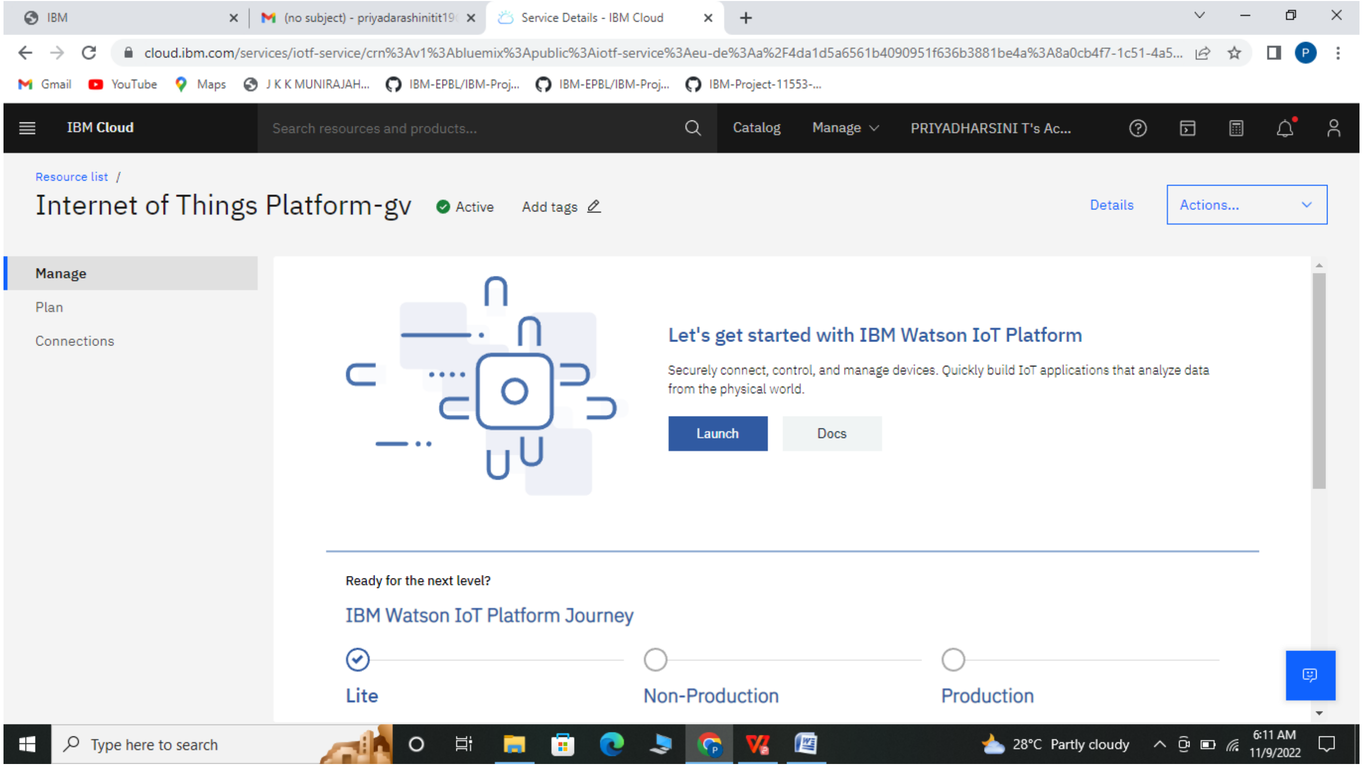
Step 2: In services section click on the IoT platform you have created



Device ID	Status	Device Type	Class ID	Date Added
PNTIBM	Connected	PNTIBM	Device	31 Oct 2022 07:57
PNTIBM1	Disconnected	12345	Device	10 Nov 2022 00:29

Device ID	PNTIBM
Device Type	PNTIBM
Date Added	31 Oct 2022 07:57
Added By	731219205011@smartinternz.com
Connection Status	Connected Connection Time: 10 Nov 2022 00:58 Client Address: 50.31.197.64 Insecure

Step 3: In IoT platform service tab click on Launch to lauch the IoT platform service



IBM
(no subject) - priyadarashinitit19
Service Details - IBM Cloud
cloud.ibm.com/services/iotf-service/crn%3Av1%3Abluemix%3Apublic%3Aiotf-service%3Aeu-de%3Aa%2F4da1d5a6561b4090951f636b3881be4a%3A8a0cb4f7-1c51-4a5...
Gmail
YouTube
Maps
J K K MUNIRAJAH...
IBM-EPBL/IBM-Proj...
IBM-EPBL/IBM-Proj...
IBM-Project-11553-...
IBM Cloud
Search resources and products...
Catalog
Manage
PRIYADHARSINI T's Ac...
Resource list /

Internet of Things Platform-gv ● Active Add tags

Let's get started with IBM Watson IoT Platform

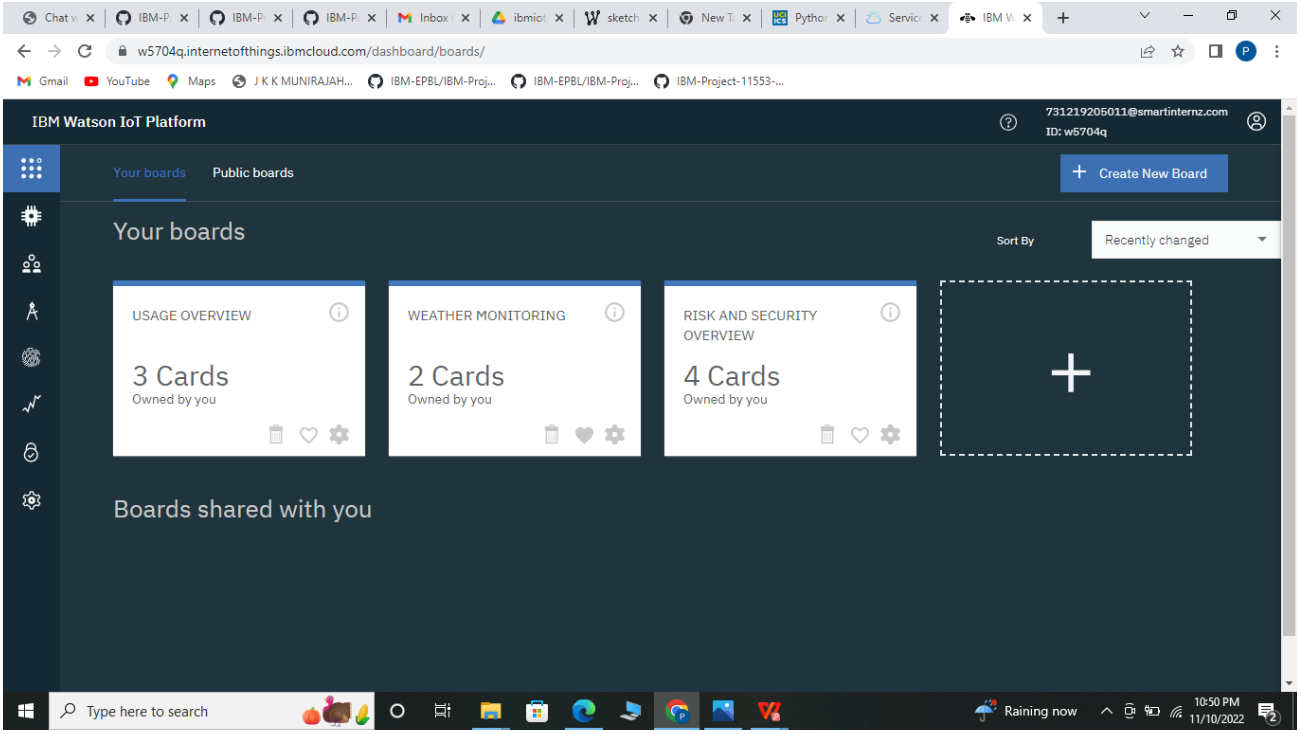
Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Ready for the next level?

IBM Watson IoT Platform Journey

Step 4: Click on the Device you have created

Step 5 : You can see your informat here



Your boards

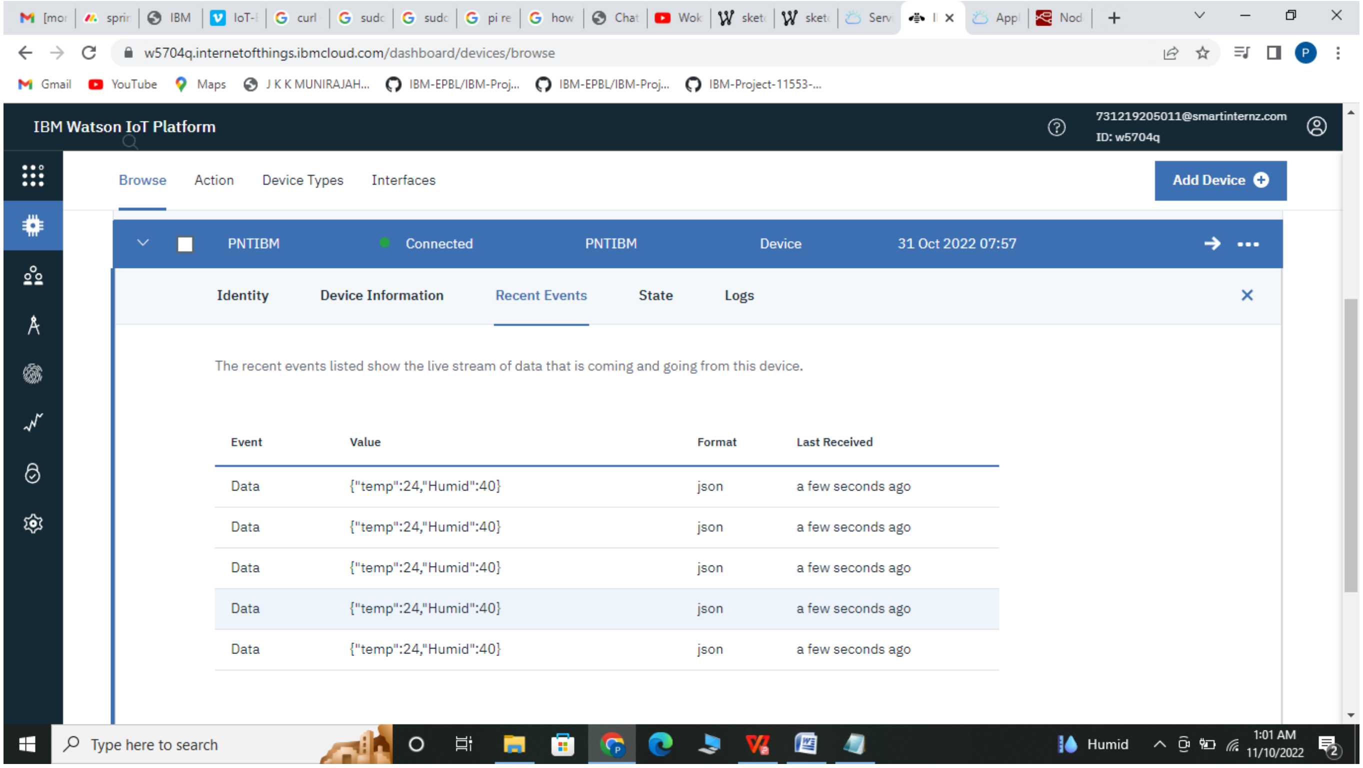
USAGE OVERVIEW
3 Cards
Owned by you

WEATHER MONITORING
2 Cards
Owned by you

RISK AND SECURITY OVERVIEW
4 Cards
Owned by you

Boards shared with you

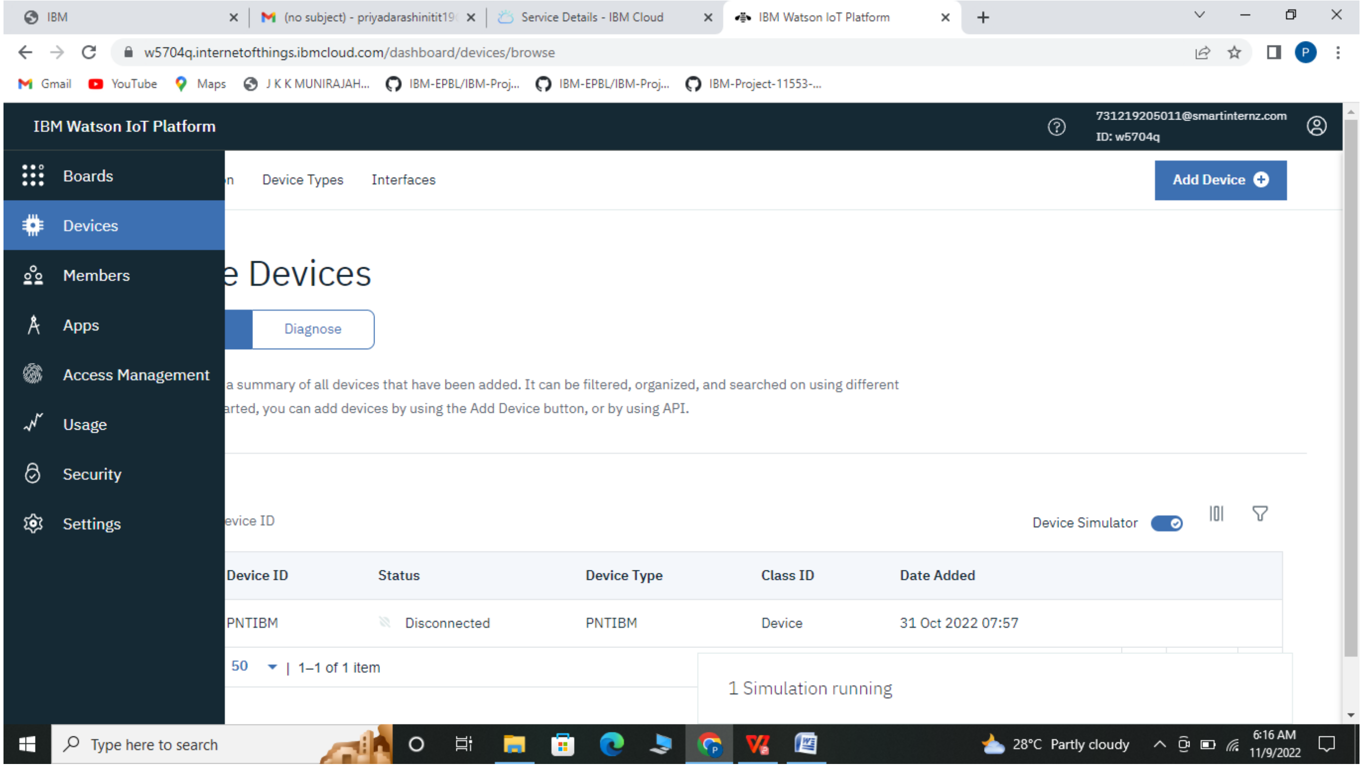
step 6: Now connect your NodeMCU to your system, if your sensor data is uploading to cloud goto your cloud plaform and check the recent events where data from NodeMCU is sent to IBM Cloud platform



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago
Data	{"temp":24,"Humid":40}	json	a few seconds ago

Step 7: To see your data in graphical representation in cloud, click on boards in the left menu

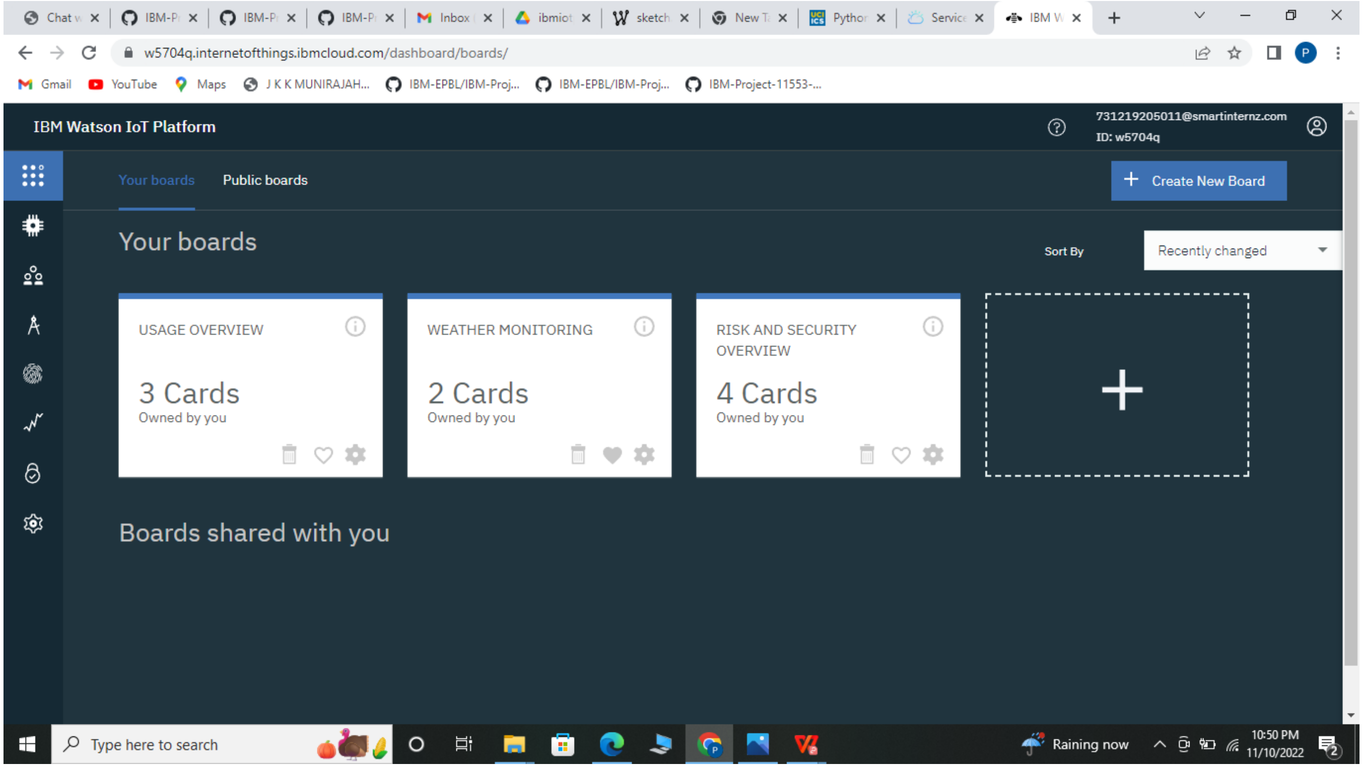


a summary of all devices that have been added. It can be filtered, organized, and searched on using different
arted, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
PNTIBM	Disconnected	PNTIBM	Device	31 Oct 2022 07:57

50 | 1–1 of 1 item

Step 8: Click on create new board which is on the top-right corner of the platform



Step 9: In create new board, give board name and click on next and submit the board.

Step 10: You board has been created and open the board

Step 11: Here you can add multiply cards with your required design specification such as line, bar graph etc.

Step12: Select a card type, then a pop-up appears where you need to select your device. After selecting your device click on Next.

Step 13: You need to connect a data set to view the incoming data on the graph. Here you need to connect your data sets by selecting data, property, Name, type, min and max value. After selecting click on NEXT to continue.

Step 14: You can select different sizes for you chart or graph here then click on Next and submit.

Step 15: Also create data set for humidity. Now you can see the graphs in the cards section.

TASK 2: Creating a Node-red UI to view data in graphical form

Step 1: In IBM cloud dashboard, click on Cloud Foundry apps.

Step 2: A new window appears where we need to NODE-RED SELDZ app created before.

Step 3: Click on Visit App URL in Node RED SELDZ service dashboard.

Step 4: click on your Node-RED flow editor where you will be redirected to the Node-RED flow editor

Step 5: To install IBM nodes in Node-red flow editor click on manage palette in the menu option which is on the top-right of the screen.

Step 6: In install section search for ibmiot and install the ibm nodes to flow editor.

Step 7: Search for IBM nodes in the filter nodes section.

Step 8: To Retrieve the data from the IBM IoT platform by using Node-RED IBM IoT Input node and double click on the IBM IoT input node.

Step 8: Select API Key from Authentication in properties.

Step 9: In API Key paste API Key, API Token and server name and update it.

To generate API Key go to IBM IoT platform

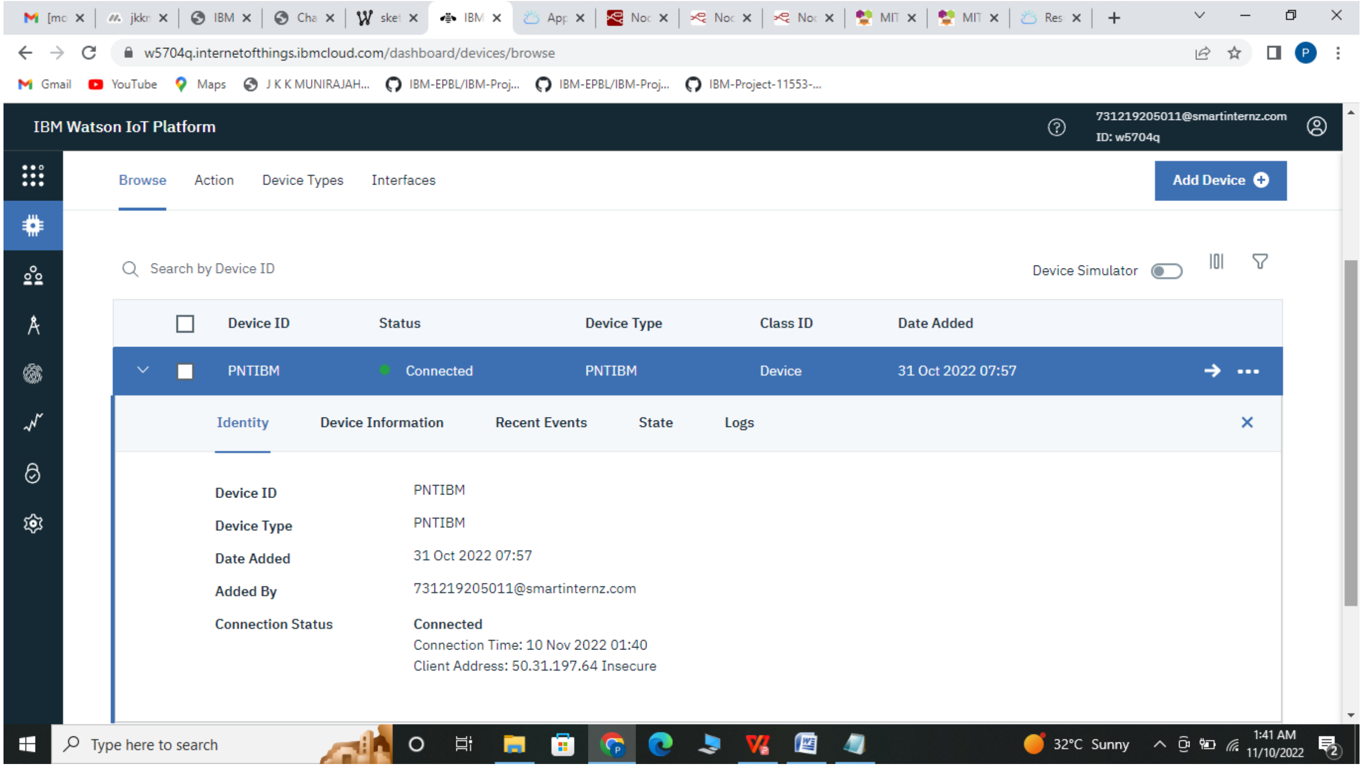
- In Apps Section -> Click onGenerate API Key.
- Click Next for Information. In Permissions select Standard Application as Role and click on

  Generate api key

- Copy your API Key and Authentication token to note them in IBM input node.

**** API token is generated only once copy it to your notepad.**

Step 10: Also update your input type as event, Device type, Device ID, command and format in.the propertiees section and click on Done.



	Device ID	Status	Device Type	Class ID	Date Added
□	PNTIBM	● Connected	PNTIBM	Device	31 Oct 2022 07:57

Device ID PNTIBM
Device Type PNTIBM
Date Added 31 Oct 2022 07:57
Added By 731219205011@smartinternz.com
Connection Status Connected
Connection Time: 10 Nov 2022 01:40
Client Address: 50.31.197.64 Insecure

Step 11: Click on Deploy option to check the connection status. If the status is disconnected check for IBM IoT properties and try again.

Step 12: Place the debug node in the flow editor and click on deploy to see the temperature. and humidity value in the debug tab

Step 13: Drag and Place the function node in the flow editor to separate the temperature and humidity value

Type msg.payload=msg.payload.d.temperature in one function and type

msg.payload=msg.payload.d.humidity in another function to separate the humidity and

temperature values from payload and click deploy

Humidity and temperature values appear seperately.

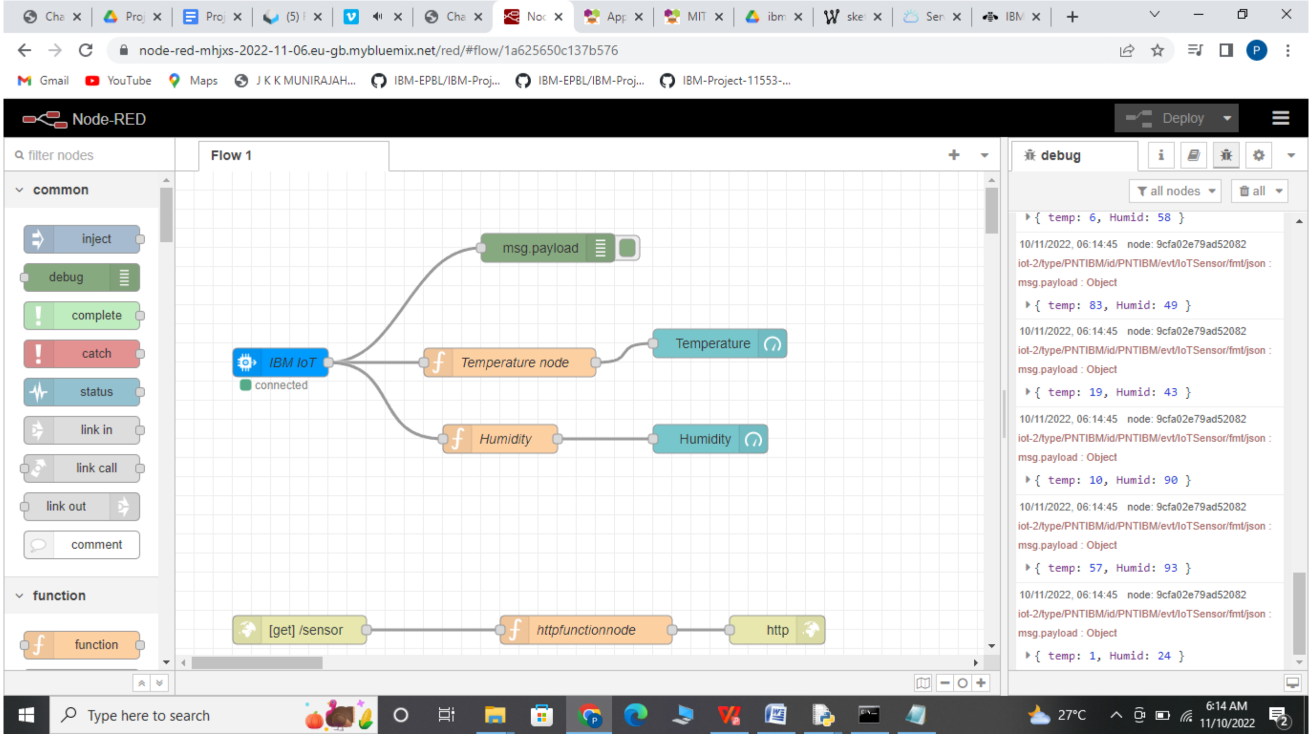
Step 14: Install the dashboard node from the manage pallet to create a UI to display

temperature and humidity values in the Dashboard.

Select gauge function and these nodes to temperature and humidity functions

Step 15: Edit both temperature and humidity nodes and deploy it.

Step 16: After editing the two nodes, deploy it.



```
▸{ temp: 6, Humid: 58 }
10/11/2022, 06:14:45   node: 9cfa02e79ad52082
iot-2/type/PNTIBM/id/PNTIBM/evt/IoTSensor/fmt/json :
msg.payload : Object
▸{ temp: 83, Humid: 49 }
10/11/2022, 06:14:45   node: 9cfa02e79ad52082
iot-2/type/PNTIBM/id/PNTIBM/evt/IoTSensor/fmt/json :
msg.payload : Object
▸{ temp: 19, Humid: 43 }
10/11/2022, 06:14:45   node: 9cfa02e79ad52082
iot-2/type/PNTIBM/id/PNTIBM/evt/IoTSensor/fmt/json :
msg.payload : Object
▸{ temp: 10, Humid: 90 }
10/11/2022, 06:14:45   node: 9cfa02e79ad52082
iot-2/type/PNTIBM/id/PNTIBM/evt/IoTSensor/fmt/json :
msg.payload : Object
▸{ temp: 57, Humid: 93 }
10/11/2022, 06:14:45   node: 9cfa02e79ad52082
iot-2/type/PNTIBM/id/PNTIBM/evt/IoTSensor/fmt/json :
msg.payload : Object
▸{ temp: 1, Humid: 24 }
```

Step 17: Display the temperature and humidity value in the Dashboard by copying and pasting the URL of the NodeRed in the new tab.

Creating an HTTP request for getting the temperature and humidity values.

Step 1: Display the temperature and humidity value in the webpage by dragging and placing the

HTTP Input and HTTP Response node in the NodeRed flow editor.

- Place the HTTP Input and HTTP Response node in the flow editor

- Edit the HTTP Input node - Select the GET method and create a URL name by typing /hom

  - Edit the Function node connected to IBM IoT input node - Edit the above function node which is connected to the IBM IOT Input node by globally setting the temperature and humidity values, the values that are set globally can be accessed by the other function

- Place and edit the function node which is placed in between the HTTP nodes - Get the global temperature and humidity values by editing the function nodes, which is basically the response whenever you hit that URL.

Step 2: Place the HTTP Input Function and HTTP Response node in the flow editor and deploy it.

Step 3: Edit the HTTP Input node - Select the GET method and create a URL name by typing/data and deploy it.

Step 4: Edit the Function node connected to IBM IoT input node - Edit the above function node which is connected to the IBM IoT Input node by globally setting the temperature and humidity value and deploy it.

Step 5: Edit the function node which is placed in between the HTTP Input and HTTP Response. node for getting the temperature and humidity values deploy it.

Step 6: Copy the URL in the NodeRed flow till .net and paste in the new tab by appending "/data" along with the URL and press Enter. Both the temperature and humidity values will be displayed on the webpage.

CONTROLLING THE LIGHT APPLIANCES ON/OFF BY GIVING COMMAND TO THE DEVICE

Step 1: Drag and Place the IBM IoT Output Node in the flow editor

Step 2: Give the device credentials and API Key in the IBM IoT Output node and deploy it so that the status of the IBM IoT Output Node will be in connected status

1. Select the API Key in the Authentication

2. Select the option in the Output Type as Device Command and fill the device credentials

3. Now select the pencil type icon which is near the API Key and fill the API Key and API Token credentials and click Update

4.Connected status shows for IBM IoT out node

Step 3: Drag and place two Button nodes from the Dashboard node

Step 4: Configure the button node for LIGHT ON and LIGHT OFF

Double click on the Button for LIGHT ON and type the group name by clicking on the pencil icon near the group tab

1. Type the group name as light on and click on the pencil icon which is near the tab

name and type the tab name as smart home and update it

2. Double click on the 2nd Button for LIGHT OFF and type the group name by clicking on the pencil icon near the group tab

3. Deploy it

Step 5: Copy the NodeRed URL till .net and paste in the new tab by typing /ui along with the NodeRed URL and press ENTER which will display the UI for controlling the Light ON/OFF

You can check the controls in IBM IoT Platform in Recent events and also by connecting the Arduino to LED

7. Create a mobile application visualizing the sensor reading and buttons to control the LED

Step 1: MIT App login

Type MIT App inventor in google search and press Enter, select the first link in the search engine

Step 2: Click on the first link you will be redirected to MIT App Inventor dashboard.

Step 3: MIT App inventor dashboard

Step 4: Click on Create Apps! It will redirect to the Gmail login page. Through Gmail account by typing your Username and Password, you can log in to the MIT App flow editor.

Step 6: By agreeing with the terms and conditions you will be redirected to the Dashboard and click on Start new project

Step 7: Type the project name and click ok....you will be redirected to the app-building flow editor

Step 8: Displaying the temperature and humidity value and controlling the Lights ON/OFF using the

Mobile app

1. Create a UI to display the Temperature and Humidity value in the Mobile App

2. Drag and Place two horizontal Arrangement in the mobile UI from the layout which is present in the palette

3. Change the height as 50 and width as fill parent in the properties

Step 9: Drag and place two Labels from User Interface which is present in the Palette and change the properties of the label

● Tick the box to display the text inside the Label as Bold and Italic in the properties

● Change the font size as 30 and set the height as fill parent and width as Automatic in the
properties, change the Text of the label as Temperature, Text Alignment as Centerand Text Color as Blue in the properties for temperature and
● change the Text of the label as Humidity, Text Alignment as Center and TextColor as Blue in the properties for humidity

Step 10: Drag and place two TextBox from User Interface which is present in the Palette and change the properties of the TextBox

● Now change the properties of the textbox by changing the height as fill parent

Step 11: Drag and Place two buttons to turn ON/OFF the lights from the palette ● Drag two Buttons from the User Interface which is present in the palette and place the on UI ●Change the properties of the LIGHT ON Button

● Change the properties of the LIGHT ON Button by modifying Fontsize as 25, Text as LIGHT ON and TextColor as green from the properties

● Change the properties of the LIGHT OFF Button by modifying Fontsize as 25, Text as LIGHT OFF and TextColor as RED from the properties

Step 12: After designing UI, go to Blocks which is present in the top right corner near the Designer and design the blocks to retrieve the data from the IBM cloud platform.

Step 13: Get the Temperature and Humidity value from the url which is created with the help of NodeRed

Step 14: To get the data from the webpage, go to a Designer part to drag and drop the WEB component from the Connectivity which is present in the Palette ● As it is a non-visible component it will not display on the UI instead, it will be displayed down to the screen

Step 15: To get the temperature and humidity values for every time intervals, and to get the recent data, go to Designer to drag and drop the CLOCK from the Sensors which is present in the Palette on to screen1

●As it is a non-visible component it will not display on the UI instead, it will be displayed down to the screen

Step 16: Now go to Blocks which is present in the top right corner and design the blocks to display the Temperature and Humidity values

Step 17: Click on the Clock1 which is present under the blocks, Drag (when Clock1.timer do)block and place it on the editor to reload the webpage

Step 18: Click on the Web1 which is present under the blocks, Drag (Set web1.Url to) block and

place it on the editor to set the URL through which we are getting temperature and humidity values.

Step 19: Click on the Text which is present under the Built-in blocks, Drag and place the (TextString) block along with the (Set Web1.Url to) block and paste the URL which gives the temperature and humidity values in the Text block

Step 20: Click on the Web1 which is present under the blocks, Drag (call Web1.Get) block and place it on the editor so that the URL is called to retrieve the temperature and humidity values which are uploaded to cloud.

Step 21: As Values in the webpage are in the JSON format we have to decode the JSON format and display the temperature and humidity values in the TextBox of the mobile app UI

- Click on the Web1 present under blocks, Drag and Place (When Web1.GotText) bloc on the edi

- Click on the (responseContent) block from the (When Web1.GotText) block and place it aside to get the response from the url and display that value in the textbox

- As the response (temperature and humidity values) are in the JSON format we have to decode the JSON data. Click on the Web1 which is present under the Blocks and select (Call Web1.JsonTextDecode jsonText) and connect it with the (response content) block

- Click on Lists which is present under Built-in and select (Lookup in pairs key) Pairs block and connect it with the (Call Web1.JsonTextDecode jsonText) block. This block connection will check for the temperature and humidity Key values. If the Key value is the same it will display the value of that particular

- drag one text block from the built in blocks and connect it with the (Lookup in pairs key) block. Enter the Key value in the Text block so that it will check for the key value in the response.

- Now display the decoded JSON data in the text box. Click on TextBox1 which is present under Screen1 and select (Set TextBox1.Text to) block and connect it with the (Lookup in pairs key) block and connect the entire block with (When Web1.GotText do) block.

- Repeat the above procedures to display the humidity value in the textbox2,make sure that you give the key value as humidity

Step 22: Open NodeRed flow editor to create an HTTP request to get command from the mobile app to device

- Drag and Place HTTP Input node and HTTP Response in the flow editor and edit the HTTP Input Node by typing “/command” for URL and click Done. So that when the LIGHT ON/LIGHT OFF Button is pressed from the mobile device” /command” URL will be hitted and the command is given to the cloud and the command, in turn, is given to the device to turn on/off the light

● Drag and place the function node in the flow editor. Edit the function node and Deploy it.

● Now copy the URL of the NodeRed flow till .net and paste in the new tab and write [/command? command=light on] for light on and [/command ?command=light off] for light off to check whether URL is working properly or not

● Connect the function node to the IBM IoT Output node to give the command to the device and connect the debug node Now whenever you are hitting the url by appending light on / light off command you need to check whether you are getting that command in the debug node. If you are getting the commands in the debug node and if your function node is connected to the ibm iot output node you will get the same data at your device also.

Step 23: Controlling the Lights ON/OFF through Mobile APP

Click on Button1 for Light on which is present under Screen1 and select (When Button1. Click do) block and place it on the editor

Step 24: Click on the Web1 which is present under the blocks, Select (Set web1.Url to) block and place it under the (When Button1.Click do) block

Step 25: Click on the Text which is present under the Build-in blocks, Drag and place the text block along with the (Set Web1.Url to) block and paste the URL in the Text block to control the

Step 26: Click on the Web1 which is present under the blocks, Drag (call Web1.Get) block and place it on the editor so that the URL is called. So now whenever the button is clicked the light on command will be sent to cloud and from cloud it will be sent to device

Step 27: Repeat the above procedure to TURN OFF the light by placing Button2

Step 28: Download the MIT AI2 Companion from the play store in the Mobile APP

Step 29: Scan the QR Code from the MIT APP Inventor - Click on Connect which is present in the top left corner and select AI Companion from the drop-down which will show a QR code.

Step 20: Now scan the QR code using Mobile app and it will display the temperature and humidity value and we can also control light on/light off using mobile app