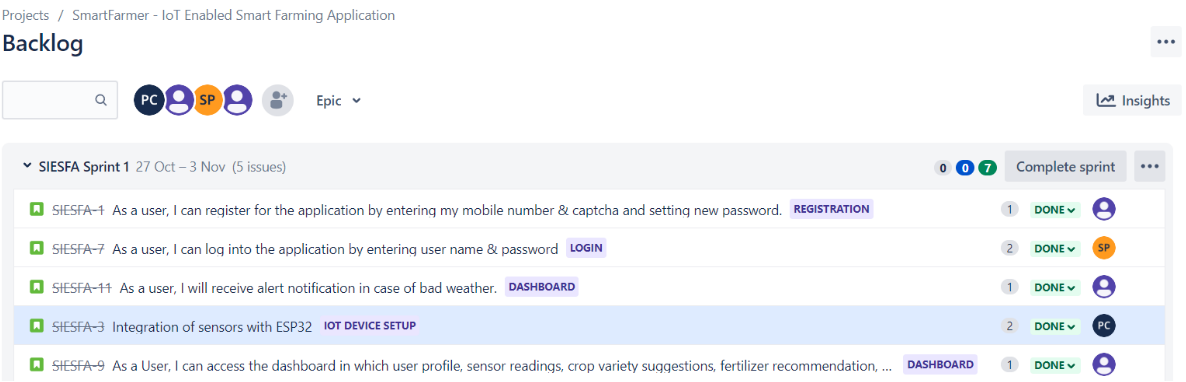
PROJECT DEVELOPMENT PHASE

SPRINT DELIVERY - 1

Team ID	PNT2022TMID52856
Project Name	SmartFarmer - IoT Enabled Smart Farming Application

BACKLOG

Release	User Story Number	User Story / Task	Story points	Priority	Team members
Sprint-1	USN-1	As a user, I can register for the application by entering my mobile number & captcha and setting new password.	1	High	Ranga Krishna Prasadh H Sathish P
Sprint-1	USN-3	As a user, I can access my sensor data in dashboard	2	High	Priya Dharshini C Vishalini AJ
Sprint-1	USN-7	As a user, I can log into the application by entering user name & password	2	High	Priya Dharshini C Vishalini AJ
Sprint-1	USN-9	As a User, I can access the dashboard in which user profile, sensor readings, crop variety suggestions, fertilizer recommendation, weather report and waterlevel are displayed.	1	High	Ranga Krishna Prasadh H Sathish P
Sprint-1	USN-11	As a user, I will receive alert notification incase of bad weather.	1	High	Priya Dharshini C Vishalini AJ


Projects / SmartFarmer - IoT Enabled Smart Farming Application

Backlog

SIESFA Sprint 1 27 Oct – 3 Nov (5 issues) 0 0 7 Complete sprint

~~SIESFA-1~~ As a user, I can register for the application by entering my mobile number & captcha and setting new password. REGISTRATION 1 DONE

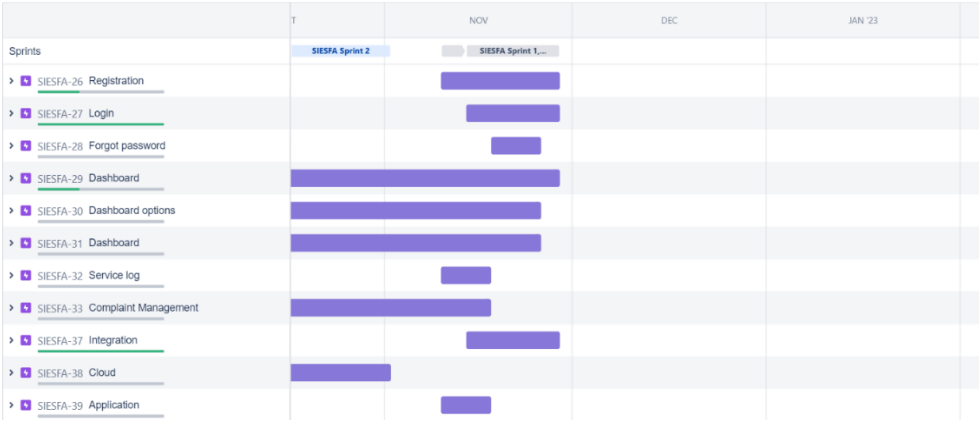
~~SIESFA-7~~ As a user, I can log into the application by entering user name & password LOGIN 2 DONE SP

~~SIESFA-11~~ As a user, I will receive alert notification in case of bad weather. DASHBOARD 1 DONE

~~SIESFA-3~~ Integration of sensors with ESP32 IOT DEVICE SETUP 2 DONE PC

~~SIESFA-9~~ As a User, I can access the dashboard in which user profile, sensor readings, crop variety suggestions, fertilizer recommendation, ... DASHBOARD 1 DONE

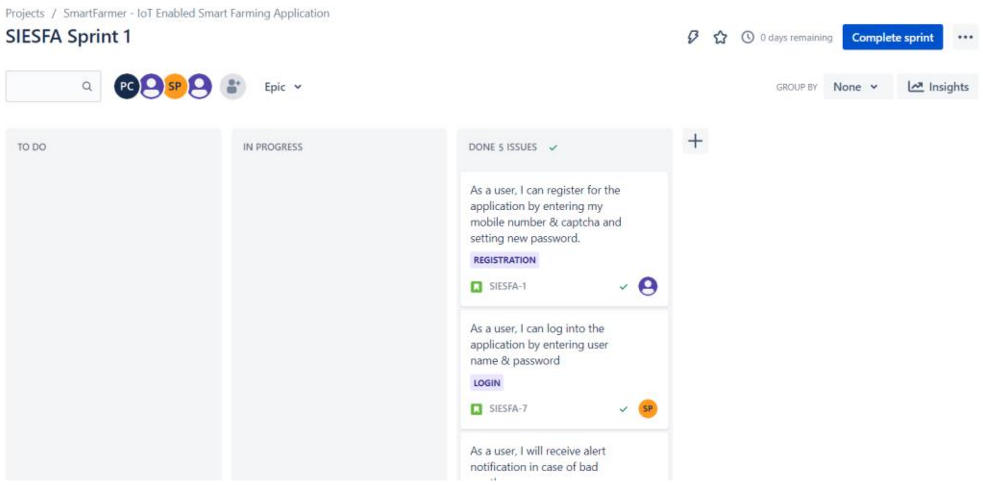
ROAD MAP



Sprints

- SIESFA-26 Registration
- SIESFA-27 Login
- SIESFA-28 Forgot password
- SIESFA-29 Dashboard
- SIESFA-30 Dashboard options
- SIESFA-31 Dashboard
- SIESFA-32 Service log
- SIESFA-33 Complaint Management
- SIESFA-37 Integration
- SIESFA-38 Cloud
- SIESFA-39 Application

BOARD



Projects / SmartFarmer - IoT Enabled Smart Farming Application

SIESFA Sprint 1

0 days remaining

TO DO

IN PROGRESS

DONE 5 ISSUES ✓

As a user, I can register for the application by entering my mobile number & captcha and setting new password.

REGISTRATION

SIESFA-1 ✓

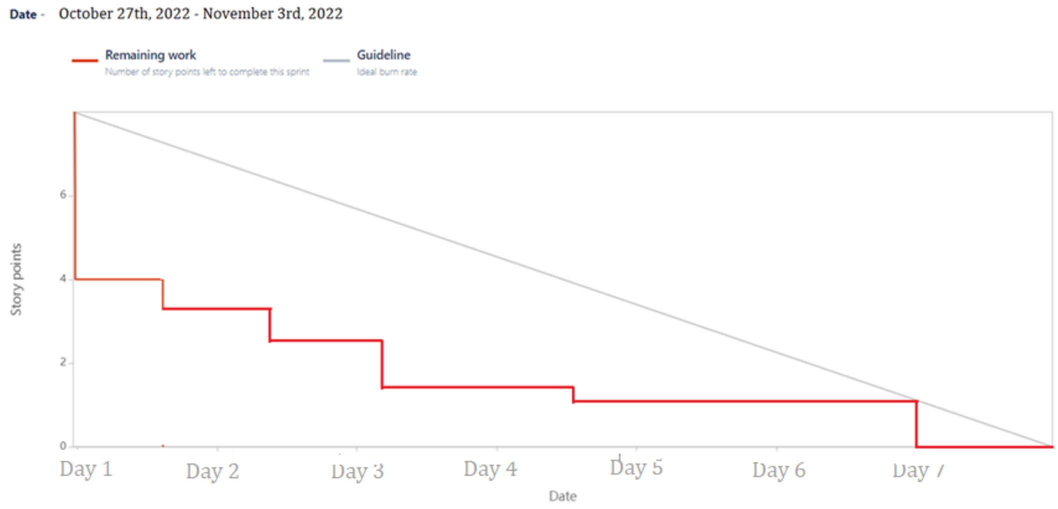
As a user, I can log into the application by entering user name & password

LOGIN

SIESFA-7 ✓ SP

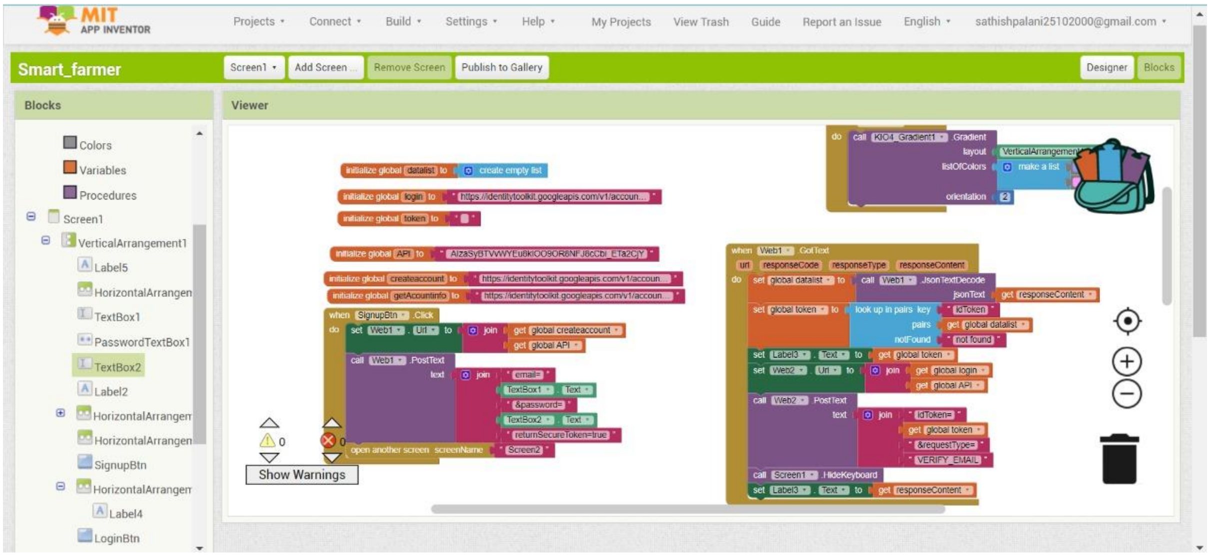
As a user, I will receive alert notification in case of bad

SPRINT BURNDOWN CHART



USN -1, USN-7

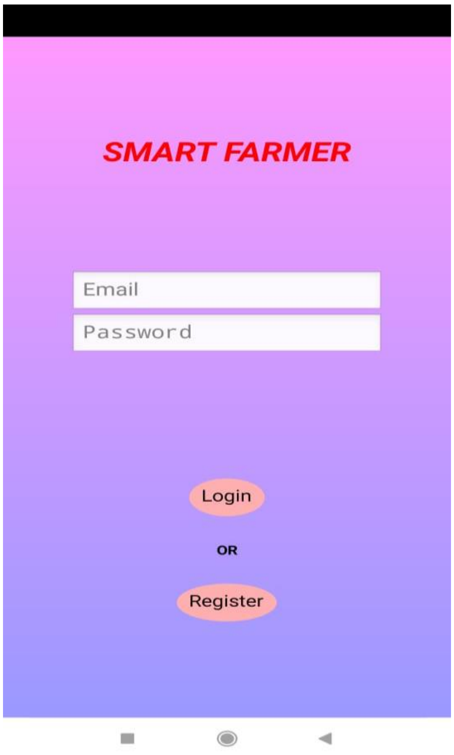
MIT APP

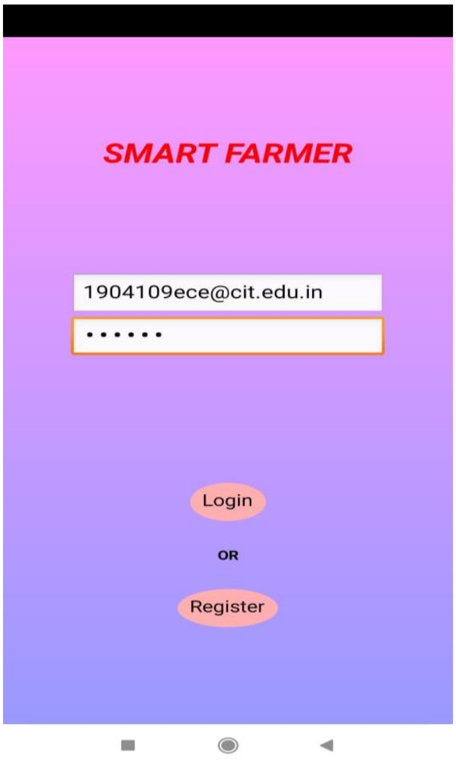


```
initialize global datalist to create empty list
initialize global login to "https://identitytoolkit.googleapis.com/v1/accoun..."
initialize global token to " "
initialize global API to "AIzaSyBTVWWYEu8kIOO9OR8NFJ8cCbl_ETa2CjY"
initialize global createaccount to "https://identitytoolkit.googleapis.com/v1/accoun..."
initialize global getAcountinfo to "https://identitytoolkit.googleapis.com/v1/accoun..."

when SignupBtn .Click
do  set Web1 . Url to join  get global createaccount
                            get global API
    call Web1 .PostText
              text  join  "email="
                          TextBox1 . Text
                          "&password="
                          TextBox2 . Text
                          "returnSecureToken=true"
    open another screen screenName "Screen2"

do  call KIO4_Gradient1 .Gradient
              layout  VerticalArrangement
        listOfColors  make a list
         orientation  2

when Web1 .GotText
 url  responseCode  responseType  responseContent
do  set global datalist to call Web1 .JsonTextDecode
                                jsonText  get responseContent
    set global token to look up in pairs  key  "idToken"
                                        pairs  get global datalist
                                     notFound  "not found"
    set Label3 . Text to get global token
    set Web2 . Url to join  get global login
                            get global API
    call Web2 .PostText
              text  join  "idToken="
                          get global token
                          "&requestType="
                          "VERIFY_EMAIL"
    call Screen1 .HideKeyboard
    set Label3 . Text to get responseContent
```

MIT Application is being developed for user. In this application, user can access the dashboard in which user profile, sensor readings, crop variety suggestions, fertilizer recommendation, weather report and waterlevel are displayed.

SMART FARMER APP:





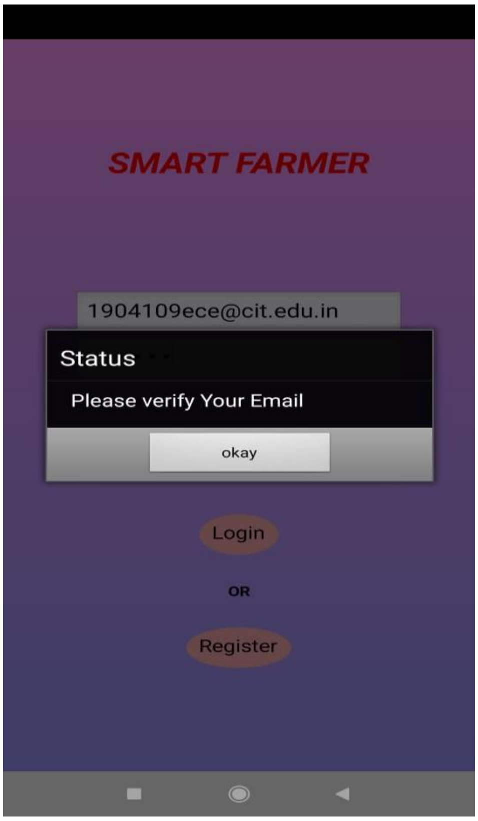
1904109ece@cit.edu.in

••••••

Registration/login screen

User can register for the application by entering mail id and setting new password. Then user can login into the application by entering user name & password

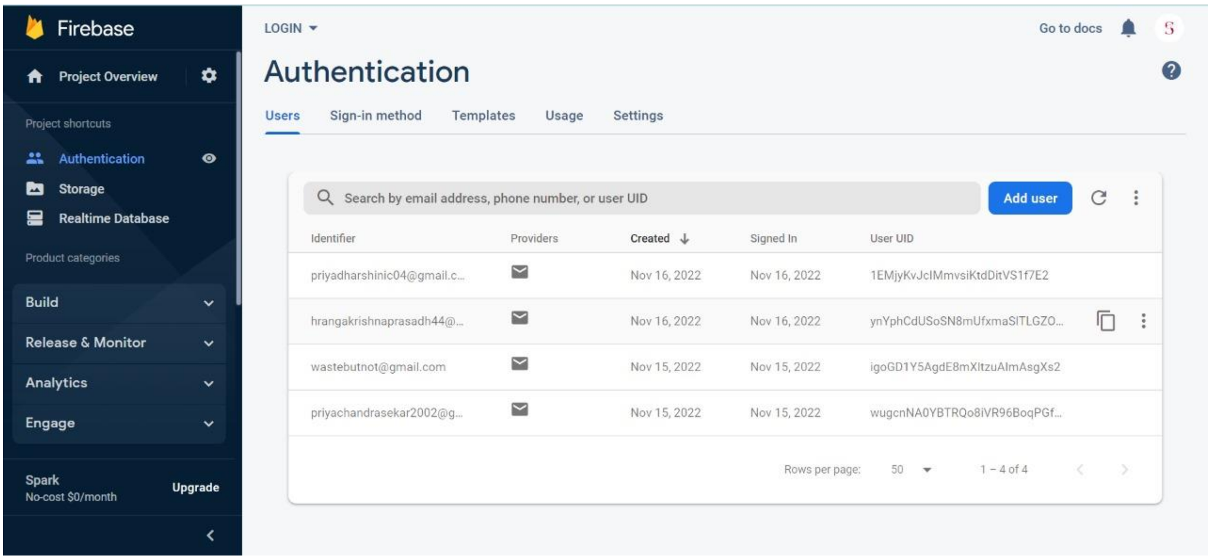
VERIFICATION MAIL GENERATED USING FIREBASE AUTHENTICATION:



1904109ece@cit.edu.in

Status

Please verify Your Email



Authentication

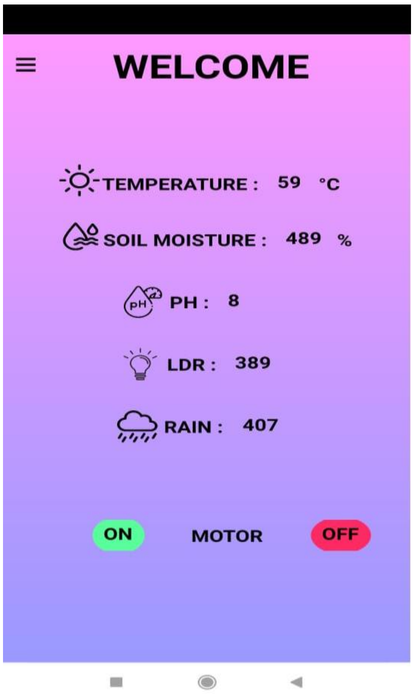
Search by email address, phone number, or user UID

Identifier	Providers	Created ↓	Signed In	User UID
priyadharshinic04@gmail.c...	✉	Nov 16, 2022	Nov 16, 2022	1EMjyKvJcIMmvsiKtdDitVS1f7E2
hrangakrishnaprasadh44@...	✉	Nov 16, 2022	Nov 16, 2022	ynYphCdUSoSN8mUfxmaSITLGZO...
wastebutnot@gmail.com	✉	Nov 15, 2022	Nov 15, 2022	igoGD1Y5AgdE8mXltzuAlmAsgXs2
priyachandrasekar2002@g...	✉	Nov 15, 2022	Nov 15, 2022	wugcnNA0YBTRQo8iVR96BoqPGf...

Rows per page: 50 1 – 4 of 4

USN-3, USN-9

HOME DASHBOARD SCREEN



WELCOME

TEMPERATURE : 59 °C

SOIL MOISTURE : 489 %

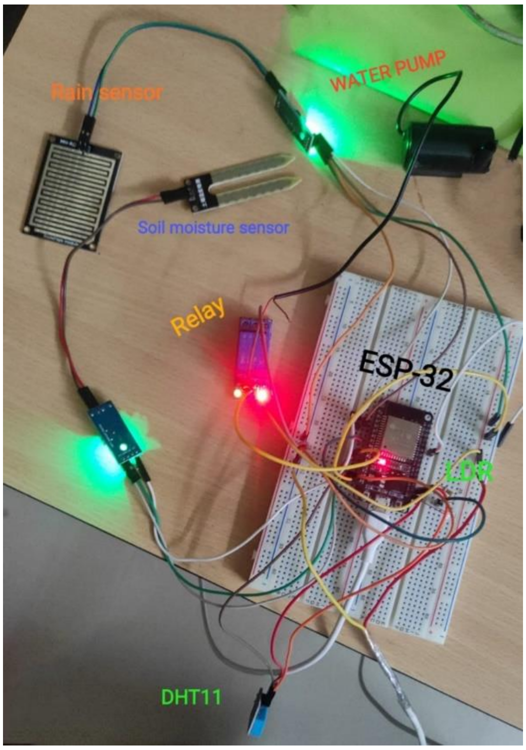
PH : 8

LDR : 389

RAIN : 407

User can access the sensor data in dashboard. This dashboard shows temperature of the field , moisture of the soil ,pH of soil and status of climate.

HARDWARE SETUP



This shows sensor ,relay and water pump integrated with ESP32.

CODE:

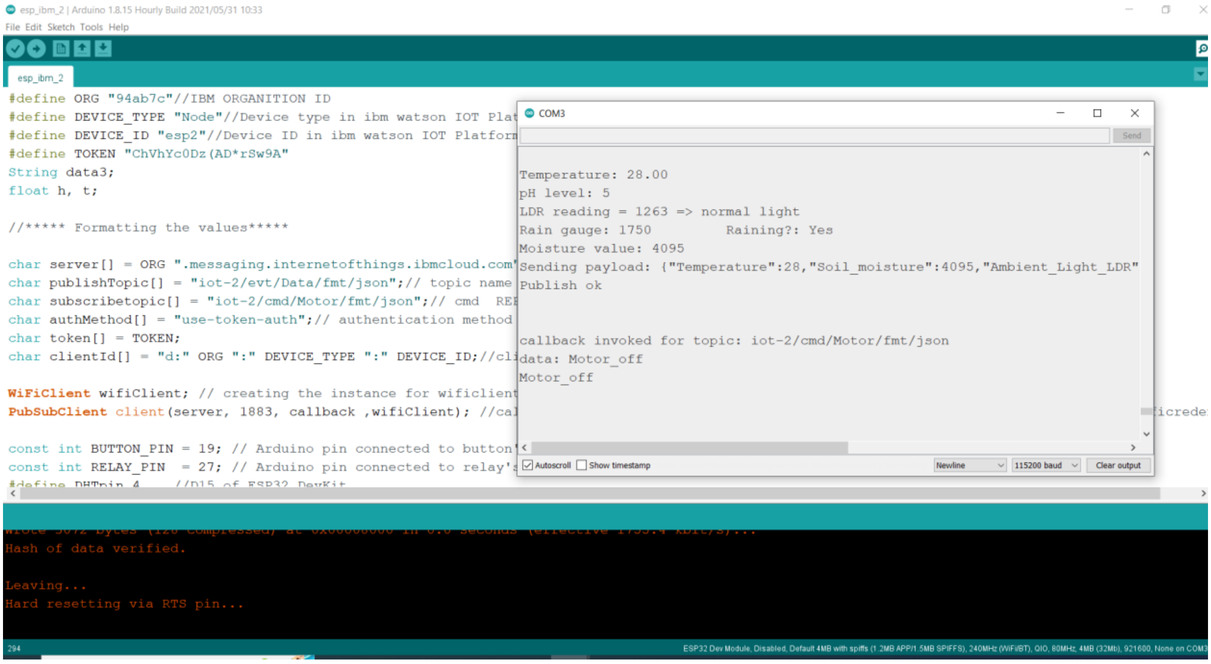
```
#include <string.h>
#include <WiFi.h>
#include <WiFiClient.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>
#include "DHT.h"

float distance=44;
#define sound_speed 0.034
int trigpin=18;
int echopin=19;
int led=5;
int LED=9;
long duration;
String message;
int ph;
int temp;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//***IBM Account**

#define ORG "94ab7c"//IBM ORGANITION ID
#define DEVICE_TYPE "Node"//Device type in ibm watson IOT Platform
#define DEVICE_ID "esp2"//Device ID in ibm watson IOT Platform
#define TOKEN "ChVhYc0Dz(AD*rSw9A"
String data3;
float h, t;

//** Formatting the 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/Motor/fmt/json";// 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


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


const int BUTTON_PIN = 19; // Arduino pin connected to button's pin

const int RELAY_PIN  = 27; // Arduino pin connected to relay's pin

#define DHTpin 4    //D15 of ESP32 DevKit

#define DHTTYPE DHT11

DHT dht(DHTpin,DHTTYPE);



#define LIGHT_SENSOR_PIN 33 // ESP32 pin GIOP36 (ADC0)


#define rainAnalog 35

#define rainDigital 34


#define AOUT_PIN 39 // ESP32 pin GIOP36 (ADC0) that connects to AOUT pin of moisture sensor

void setup()

{

  Serial.begin(115200);

  wificonnect();

  mqttconnect();

  client.subscribe(subscribetopic);

  client.setCallback(callback);
  randomSeed(42);
  pinMode(BUTTON_PIN, INPUT_PULLUP); // set arduino pin to input pull-up mode
  pinMode(RELAY_PIN, OUTPUT);        // set arduino pin to output mode

  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)\tHeatIndex (C)\t(F)");
  dht.begin();
  //dht.setup(DHTpin, DHTesp::DHT11); //for DHT11 Connect DHT sensor to GPIO 17
  //dht.setup(DHTpin, DHTesp::DHT22); //for DHT22 Connect DHT sensor to GPIO 17

  pinMode(rainDigital,INPUT);

  delay(10);
  Serial.println();
}

void loop()
{

/*
Serial.println("distance"+String(distance)+"cm");
 if(distance<100)
 {
   message="Alert";
   digitalWrite(led,HIGH);
   } else
{
  message="Normal";
  digitalWrite(led,LOW);
  }
  delay(1000);
  PublishData(distance,message);
  if (BUTTON_PIN == LOW) {
    Serial.println("The button is being pressed");
    digitalWrite(RELAY_PIN, HIGH); // turn on
  }
  else
  if (BUTTON_PIN == HIGH) {
    Serial.println("The button is unpressed");
    digitalWrite(RELAY_PIN, LOW);  // turn off
  }
*/
/*
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();
  // Read temperature as Fahrenheit (isFahrenheit = true)
  float f = dht.readTemperature(true);

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  // Compute heat index in Fahrenheit (the default)
  float hif = dht.computeHeatIndex(f, h);
  // Compute heat index in Celsius (isFahreheit = false)
  float hic = dht.computeHeatIndex(t, h, false);

  Serial.print(F("Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("°C "));
  Serial.print(f);
  Serial.print(F("°F  Heat index: "));
  Serial.print(hic);
  Serial.print(F("°C "));
  Serial.print(hif);
  Serial.println(F("°F"));
*/
  client.loop();
  Serial.println("\n\n");
  float temp = dht.readTemperature();
  Serial.print("Temperature: ");
  Serial.println(temp);
  ph = random(5,9);//random value as sensor not available
  Serial.print("pH level: ");
  Serial.print(ph);

 // reads the input on analog pin (value between 0 and 4095)
  int analogValue = analogRead(LIGHT_SENSOR_PIN);
  Serial.print("\nLDR reading = ");
  Serial.print(analogValue);   // the raw analog reading

  // We'll have a few threshholds, qualitatively determined
  if (analogValue < 400)
  {
    Serial.print(" => Dark condition");
  } else if (analogValue < 800)
    {
      Serial.print(" => Dim light");
    } else if (analogValue < 2000)
    {
      Serial.print(" => normal light");
    } else if (analogValue < 3200)
      {
        Serial.print(" => Bright");
      } else
      {
        Serial.print(" => Very bright");
      }

  int rainAnalogVal = analogRead(rainAnalog);
  int rainDigitalVal = digitalRead(rainDigital);
  Serial.print("\nRain gauge: ");
  Serial.print(rainAnalogVal);
  Serial.print("\t Raining?: ");
  if(rainDigitalVal==1)
  {
    Serial.print("No");
  }
  else
  {
    Serial.print("Yes");
  }

  int value = analogRead(AOUT_PIN); // read the analog value from sensor
  Serial.print("\nMoisture value: ");
  Serial.println(value);

  message="Normal";

  PublishData(temp,value,analogValue,rainAnalogVal,ph,message);
  callback;
  delay(1000);
}
```

```
//**Publish***

void PublishData(int temp,int sm,int ldr,int rain,int ph, String a)
{
  mqttconnect();

  //creating the String in in form JSon to update the data to ibm cloud

  DynamicJsonDocument doc(1024);
  String payload;
  doc["Temperature"]=temp;
  doc["Soil_moisture"]=sm;
  doc["Ambient_Light_LDR"]=ldr;
  doc["pH_sensor"]=ph;
  doc["Rain_sensor"]=rain;
  doc["message"]=a;
  serializeJson(doc, payload);
  Serial.print("Sending payload: ");
  Serial.println(payload);

  if (client.publish(publishTopic,(char*)  payload.c_str()))
  {
    Serial.println("Publish ok");// if upload sucessful
    client.subscribe(subscribetopic);
  }
  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();
  }
}

const char* ssid     = "POCO";
const char* password = "hpranga44";
const char* host = "192.168.6.129";
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin(ssid, password);//passing the wifi credentials to establish the connection
  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)
{
 String comdata="";
 Serial.println("\n");
 Serial.print("callback invoked for topic: ");
 Serial.println(subscribetopic);
 for (int i = 12; i < payloadLength-2; i++)
 {
   //Serial.print((char)payload[i]);
   comdata += (char)payload[i];
 }
 Serial.println("data: "+ comdata);
 if(comdata=="Motor_on")
 {
    Serial.println(comdata);
   digitalWrite(RELAY_PIN,HIGH);
 }
 else
 {
   Serial.println(comdata);
   digitalWrite(RELAY_PIN,LOW);
 }
comdata="";
}
```

OUTPUT


esp_ibm_2 | Arduino 1.8.15 Hourly Build 2021/05/31 10:33
File Edit Sketch Tools Help
esp_ibm_2
COM3
Send
☑ Autoscroll □ Show timestamp
Newline
115200 baud
Clear output

```
#define ORG "94ab7c"//IBM ORGANITION ID
#define DEVICE_TYPE "Node"//Device type in ibm watson IOT Plat
#define DEVICE_ID "esp2"//Device ID in ibm watson IOT Platform
#define TOKEN "ChVhYc0Dz(AD*rSw9A"
String data3;
float h, t;

//***** Formatting the values*****

char server[] = ORG ".messaging.internetofthings.ibmcloud.com"
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name
char subscribetopic[] = "iot-2/cmd/Motor/fmt/json";// cmd  REI
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//cli

WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //cal                icrede

const int BUTTON_PIN = 19; // Arduino pin connected to button'
const int RELAY_PIN  = 27; // Arduino pin connected to relay's
#define DHTpin 4     //D15 of ESP32 DevKit
```

```
Temperature: 28.00
pH level: 5
LDR reading = 1263 => normal light
Rain gauge: 1750          Raining?: Yes
Moisture value: 4095
Sending payload: {"Temperature":28,"Soil_moisture":4095,"Ambient_Light_LDR"
Publish ok


callback invoked for topic: iot-2/cmd/Motor/fmt/json
data: Motor_off
Motor_off
```

```
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1755.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

294   ESP32 Dev Module, Disabled, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None on COM3

Sensor readings are displayed in the serial monitor of arduino IDE