Smart Farmer

IoT Enabled Smart Farming ApplicationTeam ID: PNT2022TMID48510

Bachelor of Engineering

Electronics Communication & Engineering

CHETTINAD COLLEGE OF ENGINERING TECHNOLOGY – 639114

TEAM ID : PNT2022TMID48510

Team Members:

920219106020 - PRADEEP PAL T

920219106021 - PREMNATH T

920219106001 - AAKASH A

920219106003 - ARAVINDH J

1: INTRODUCTION

The main objective of this project is to design a IoT based Smart farming system which helps farmers to monitor their fields by monitoring the field parameters such as soil moisture, temperature and humidity etc. Monitoring systems helps to enhance the crop productivity. IoT technology is an evolving technology in recent times. Conventional farming requires manual labors to monitor the field condition which was the time-consuming process. To overcome this downside, smart farming concept was introduced. Through smart farming, farmers can automate the process of farming through the IoT based Mobile/Web application.

1:1: Project Overview

Temperature, Humidity are the important factors which affects the quality and productivity of the plant growth. Continuous monitoring of these parameters helps to provide valuable information to the farmers which in term helps to automate the irrigation process. This IoT based Smart farmer system will continuously monitor the temperature, humidity of the field update those value to the IoT based cloud application. Farmer can monitor their fields through the IoT based application. They can also control the end devices like pump motors to supply water to their fields through IoT based application.

1:2: Purpose

The purpose of smart farmer project is to help farmers in the irrigation process. The system provides various parameters like temperature, humidity etc. to monitor the condition of the fields and to protect the crops. Based on the temperature, soil moisture, water level of the field etc., and system will take necessary action and the entire operation can be controlled by the IoT application.

2: LITERATURE SURVEY

2:1 Existing Problem

The main problem in the conventional farming is wastage of water, use of fertilizers and use of human resource. To overcome these problems a smart farming system has been proposed. In this project irrigation process will be done automatically using different sensors like Temperature sensor, Humidity sensor etc. This project helps to replace the manual work. The proposed system will monitor crop-field using Temperature sensor, Humidity sensor, Soil moisture sensor etc. By monitoring these parameters the irrigation process can be automated.

2:2:References

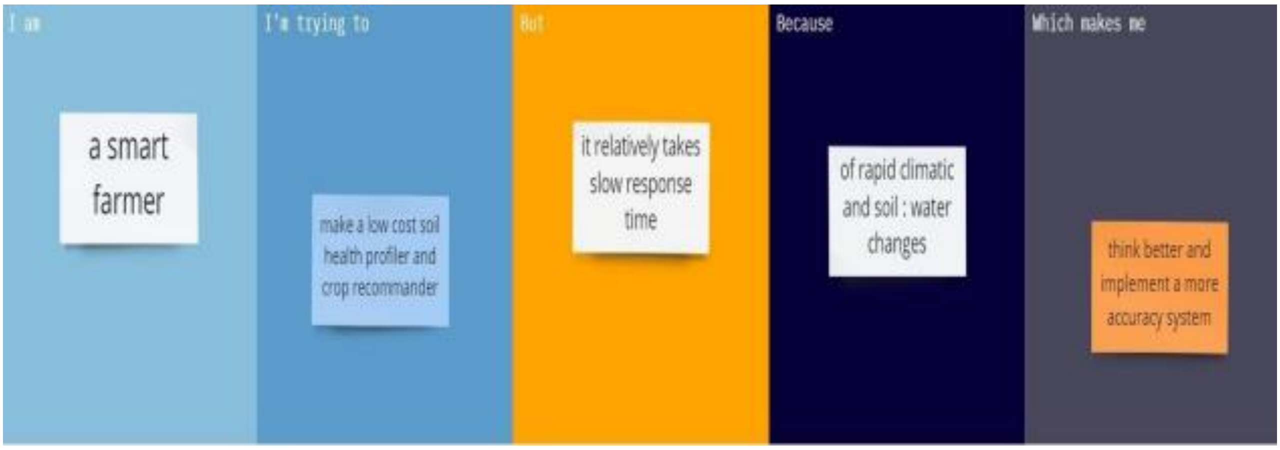
S.NO	Paper Title	Author	Journal name& year of publication	Description
1	IoT Enabled Smart Farming And Irrigation System	M. Rohith, R Sainivedhana, Dr. N. Sabiyath Fatima	IEEE 2021	In this paper, authors have demonstrated a IoT enabled smart farming and irrigation system to automate the process of watering to plants. This system helps to measure the values of various parameters such as humidity, moisture and temperature of plants and water them accordingly. This system consists of three sensors which will sense the values of humidity, moisture and temperature of plants. If any of the sensor values decreases the motor automatically turns on the water for plants. The ultimate significance of the paper is that most of the manual work is reduced and watering process is automated with the help of IoT enabled devices as a result of which healthy plants can be grown.

2	A Multi-collective, IoT-enabled, Adaptive Smart Farming Architecture	G.Kakamoukas, P. Sariciannidis, G.Livanos, M.Zervakis, D.Ramnalis, V.Polychrnos, T.Karamitsou, A.Folinas, N. Tsitsiokas	IEEE 2019	In this paper, authors have proposed a precision architecture for Smart Farming in order to use precise and efficient approaches for monitoring and processing information from farms, crops, forestry, and livestock aiming at more productive and sustainable rural development. This proposed architecture encloses wireless sensor networks, meteorological stations and unmanned aerial vehicles along with an information processing system that leverages machine learning and computing technologies. The innovation of the proposed architecture lies in the creation of an integrated monitoring and decision support system for efficient allocation of resources and protection of plant capital from the diseases.
3	A Systematic Review of IoT Solutions for Smart Farming	Emerson Navarro, Nuno Costa, and Antonio Pereira	MDPI 2020	In this work, authors have presented a systematic review of the state-of-the-art of IoT adoption in smart agriculture and identified the main components and applicability of IoT solutions. In this particular work it was observed that the use of artificial intelligence and image processing techniques has become more common to improve the management of smart farming. From the identified applications of IoT for smart farming it was observed that the most common application

				is the monitoring of crops. Here, authors showed that different network protocols may be simultaneously used in IoT solutions for smart farming.
4	Internet of Things and LoRaWAN–Enabled Future Smart Farming	Bruno Citoni, Francesco Fioranelli, Muhammad A. Imran,Qammer H. Abbasi		In this paper authors have explained about LoRaWAN which is been under the spotlight in recent years due to its suitability to be the standard communication protocol for IoT deployments. It provides long communication range and low energy consumption by drastically reducing the available data rate. They also explained about the development of LoRaWAN enabled smart agriculture test to improve the understanding about the impact of the limitations using experimental test data, and moving towards building predictive models and adaptive network management algorithms for smart farming using the data collected.
5	A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming	Muhammad Shoaib Farooq, Shamyla Riaz, Adnan Abid, Kamran Abid, Muhammad Azhar Naeem	IEEE 2019	In this paper, authors have explained the aspects of technologies involved in the domain of IoT in agriculture. They explained about the major components and technologies, network architecture, network layers, network topologies and protocols involved in developing IoT based smart farming system.

2:3: Problem Statement Definition

Customer Problem Statement: Agriculture is representing an essential element in the developing countries .In the agriculture, there is problem for farmers to making a suitable crop for the soil to get the better yield. Soil analysis is a valuable tool for the problem .Our system providing a smart technology for soil analyzing. Therefore, it results in helps the farmers to making a suitable crop and improves the yield A handheld Soil health profiler can solve these issues by giving the soil nutrient details and recommend suitable crops.This project involved an NodeMCU which talk to the internet. Soil NPK sensor to measure level of the macro nutrients present in the soil. The sensor data is send to the cloud server. Then the digitally generated soil fertility and crop prediction e-report is get in the specified link. It is very useful for farmers to cultivate suitable crop to the field. It improves soil quality and soil fertility. This gives more yields for the farmers. The designed system can be an easy alternative for the lengthy laboratory process. The designed system also takes the inputs like the geographical location, season of the year, etc to make the recommender system.

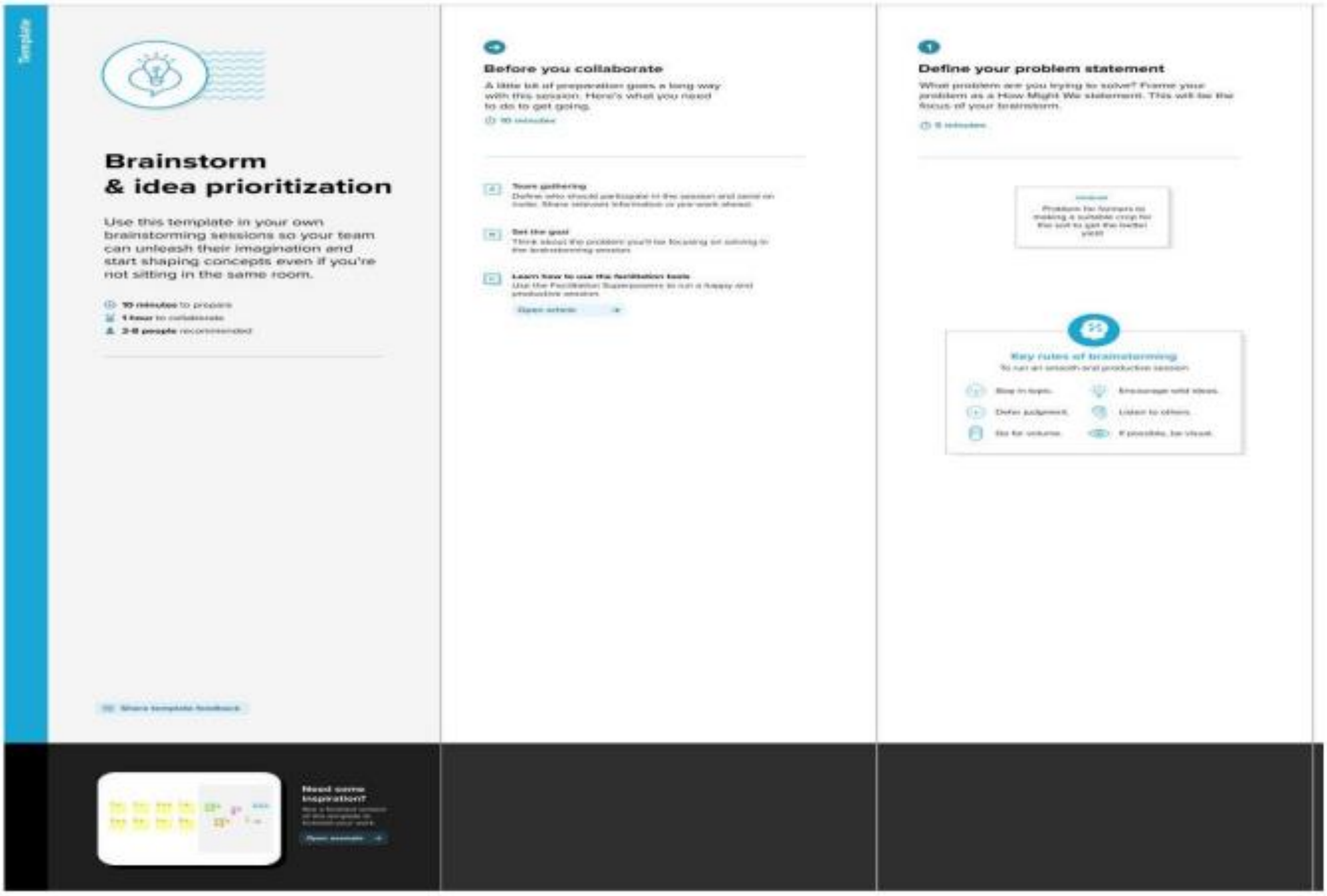


3: IDEATION AND PROPOSED SOLUTION

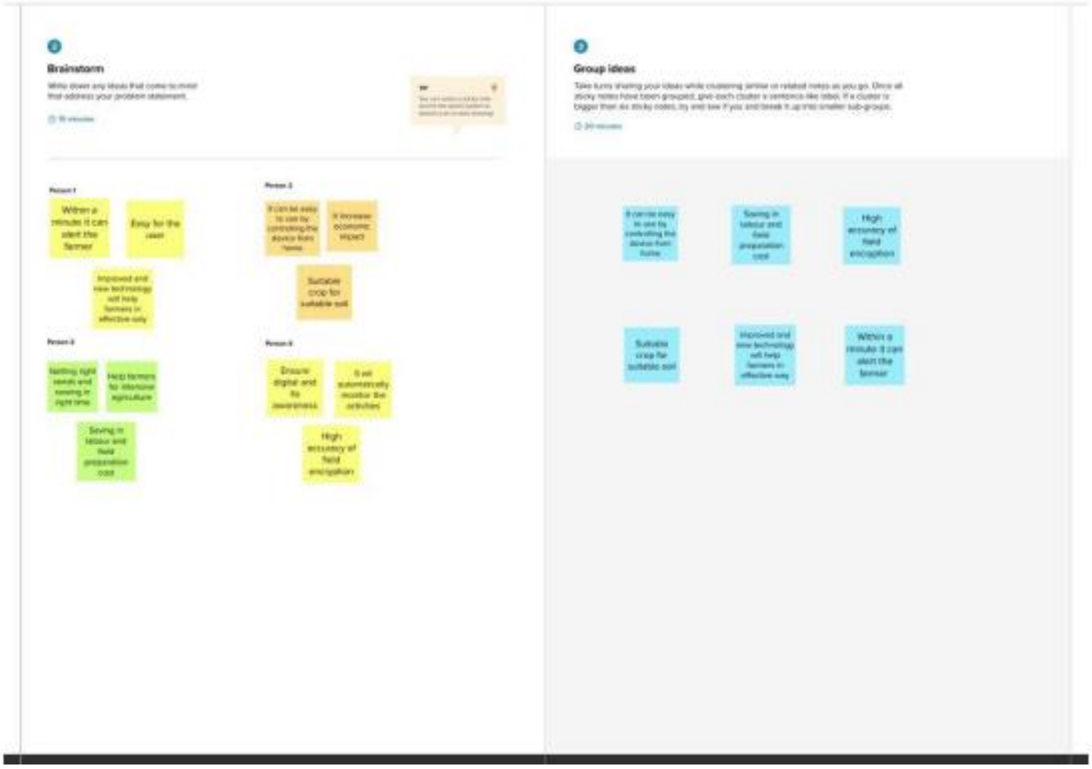
3:1: Empathy Map Canvas

3:2: Ideation And Brainstorming

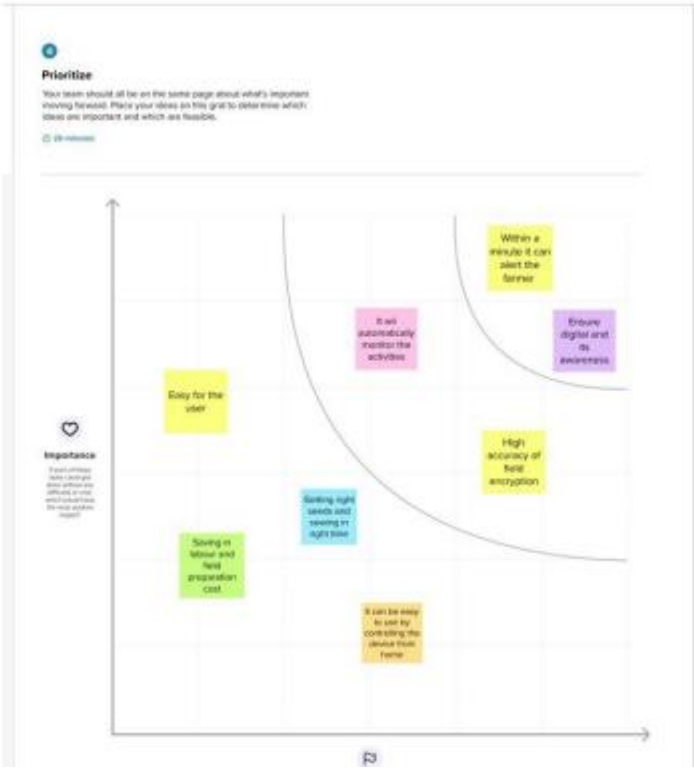
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3:3: Proposed Solution

S.No.	Parameter	Description
1.	1. Problem Statement (Problem to be solved) Difficulty in selecting suitable crop for the soil.	• Difficulty in selecting suitable crop for the soil
2.	Idea / Solution description	A handheld Soil health profiler can solve these issues by giving the soil nutrient details and recommend suitable crops
3.	Novelty/Uniqueness	It takes the inputs like the geographical location, season of the year to make the recommender system.
4.	Social Impact/Customer Satisfaction	It takes the inputs like the geographical location, season of the year to make the recommender system.
5.	Scalability of the Solution	Based on all the inputs from the system , it recommends the required crops and fertilizer to the soil.

3:4:Problem Solution Fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

This product is for farmer who can cultivate the crops.
Our main is to help farmer.

6. CUSTOMER CONSTRAINTS CC

- May have confusions on deciding the crops.
- Insufficient knowledge about the soil.

5. AVAILABLE SOLUTIONS AS

- Carrying a guidebook to everywhere we go.
- Accompanying an experienced people.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Difficulty in finding suitable crop for the soil.
- No knowledge or experience about the soil nutrients as the user is just a learner.

9. PROBLEM ROOT CAUSE RC

- Humans are incapable of memorising large number of datas.
- Eventhough books are available, it is difficult to identify the species.

7. BEHAVIOUR: BE

- Due to mismatch cropping, the field loses its fertility and heavy loss may occur.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

During an excursion or hiking or some trip to the forest, if a nearby person uses the app, then the person also will go for a try to know about the plants and animals what they are seeing through their eyes.

4. EMOTIONS: BEFORE / AFTER EM

- At first,farmers find it difficult to choosethe crop that grows well with respect tothe nutrients present in the particular soil.
- After, it becomes easy task for them to identify suitable crop for the soil.

10. YOUR SOLUTION SL

- The designed system can be an easy alternative for the lengthy laboratory process .
- The designed system also takes the inputs like the geographical location, season of the year, etc to make the recommender system.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

- Searching through the internet to know about the soil specifications.

8.2 OFFLINE

- Get help from experienced people.
- Carrying guide.

4: REQUIREMENT ANALYSIS

4:1:Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Log in to system	Check Credentials Check Roles of Access.
FR-4	Manage Modules	Manage System Admins Manage Roles of User Manage User permission
FR-5	Check Weather details	Temperature details Humidity details
FR-6	Log out	Exit

4:2:Non-Functional Requirements

Following are the non-functional requirements of the proposed solution.

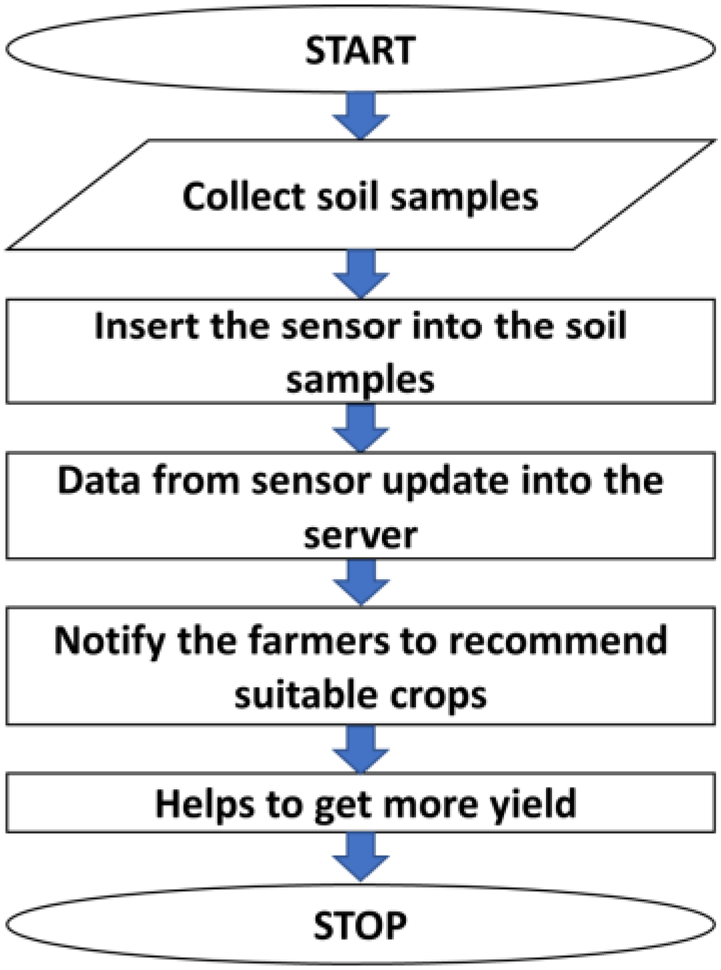
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability refers to efficiency in use, remember ability, lack of errors in operation and subjective pleasure.
NFR-2	Reliability	The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages...
NFR-3	Scalability	Scalability is a major concern for IoT platforms. It has shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making isfeasible in an environment.
NFR-4	Security	Sensitive and private data must be protected from their production until the decision making and storage stages.

NFR-5	Performance	The idea of implementing integrated sensors with sensing soil and environmental or ambient parameters in farming will be more efficient for overallmonitoring.
NFR-6	Availability	Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc

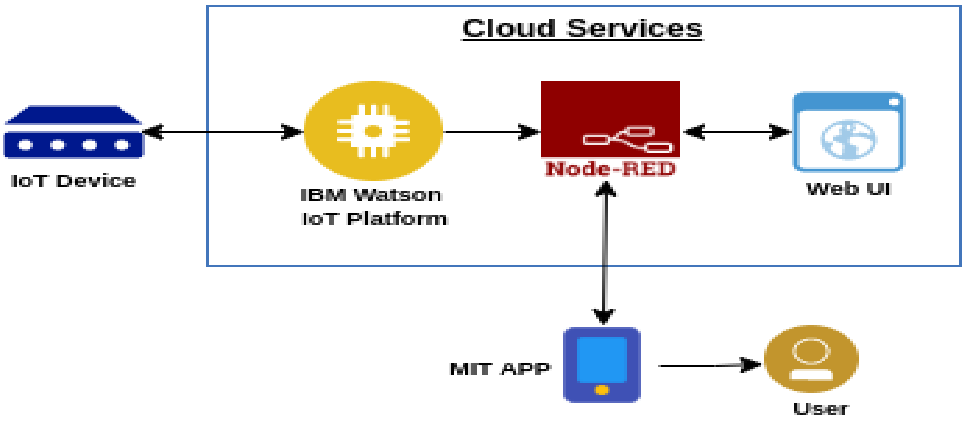
5: PROJECT DESIGN

5:1:Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirementgraphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5:2:solution & technical architechture



1. The different soil parameters (temperature, humidity, soil moisture) are sensed using different sensors, and the obtained value is stored in the Cloud.
2. NodeMCU is an open source which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO.
3. All the collected data are provided to the user through a mobile application that was developed using app inventor. The user could make a decision through an app, that what crop is suitable for the soil in the particular season by the humidity and temperature check. By using the app, they can be remotely operated by the user.

Table -1: Components and Technologies

S.No	Component	Description	Technology
1.	User Interface	User interaction with application such as UI and Mobile app	HTML, CSS, JavaScript/ Angular Js/React Jsetc.
2.	Application Logic -1	Logic for a process in the Application	Python
3.	Application Logic -2	Logic for a process in the Application	IBM Watson IoT Service
4.	Application Logic -3	Logic for a process in the Application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM Cloud
7.	File Storage	File storage requirements	IBM Block storage or other storage service or local Filesystem
8.	External API – 1	Purpose of External API used in the application	IBM Weather API, etc.,
9.	Machine Learning Model	Purpose of Machine learning model	Object recognition model, etc.,
10.	Infrastructure (Server/Cloud)	Application Deployment on Local system/Cloud local server configuration	Local, CloudFoundry, Kubernetes, etc.,

Table -2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource Framework
2.	Security Implementation	Sensitive and private data must be protected from their production until the decision- making an dstorage stages	e.g. Node-Red, Open weather AppAPI, MIT App Inventor, etc.
3.	Scalable Architecture	scalability is a major concern for IoT platforms. It has been shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousand	Technology used
4.	Availability	Automatic adjustment of farming equipment madepossible by linking information lik e crops/weather an d equipment to auto-adjust temperature, humidity, etc.	Technology used
5.	Performance	The idea of implementing integrated sensors with sensing soil and Environmental or ambient parameters in farming will be more efficient for overall monitoring.	Technology used

6: PROJECT PLANNING AND SCHEDULING

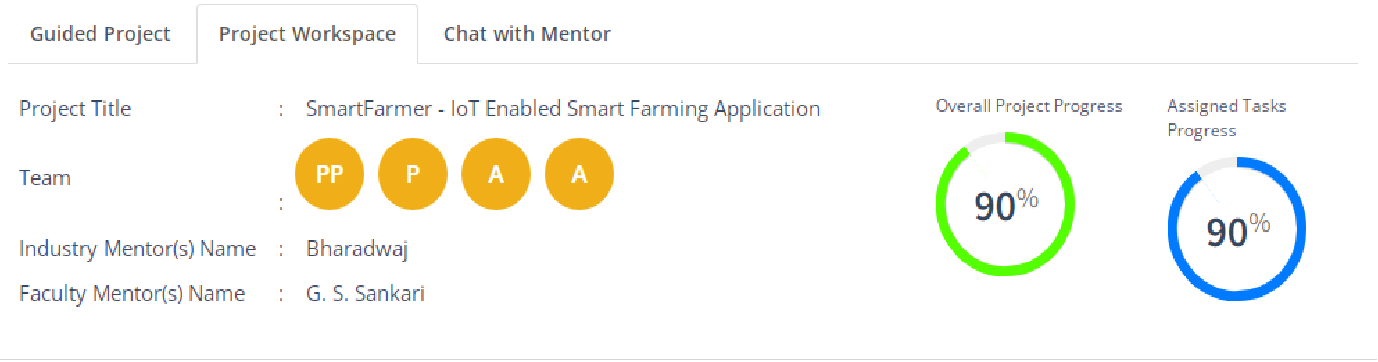
6.1 Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Point	Priority	Team Members
Sprint-1	Simulation Creation	USN-1	Connect Sensors and Arduino with python code	12	High	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	12	High	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J
Sprint-3	Registration (Mobile User MIT APP INVENTER)	USN-3	As a user, I can register for the application by entering my email and password	4	High	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J
Sprint-3	Login	USN-4	As a user, I can log into the applicationby entering username & password.	4	High	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J
Sprint-3	Dashboard	USN-5	As a User can view the dashboard, and this dashboard includes temperature, Humidity and Soil moisture values	6	High	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J
Sprint-4	Logout	USN-7	Then check the Temperature, humidity and soil moisture after logout or exit the application	6	Medium	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J
Sprint-4	Web UI	USN-8	As a user, I need to have a friendly user interface to easily view and access the resources	6	Medium	PRADEEP PAL T PREMNATH M AAKASH A ARAVINDH J

6:2:Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date	Sprint Release Date(Actual)
Sprint-1	18	6 Days	24 Oct 2022	29 Oct 2022	12	04 OCT 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	12	08 NOV 2022
Sprint-3	12	6 Days	07Nov 2022	12 Nov 2022	14	12 NOV 2022
Sprint-4	10	6 Days	14Nov 2022	19 Nov 2022	12	19 NOV 2022

6:3: Reports from JIRA

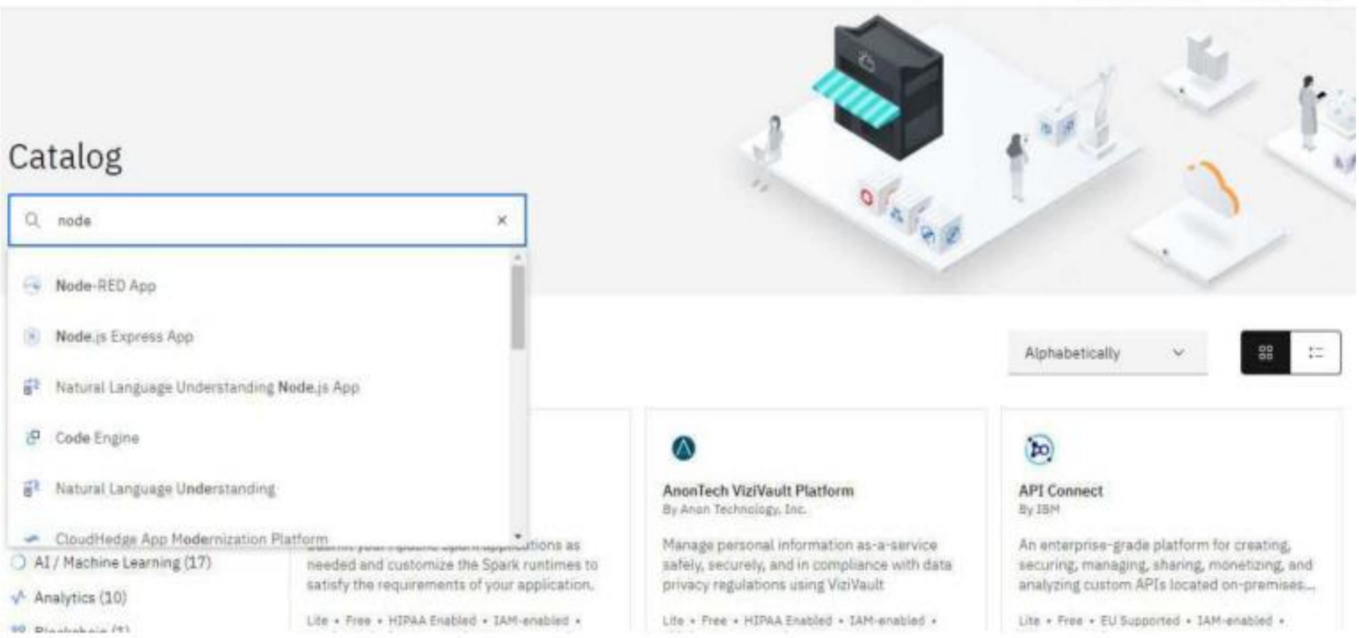


7: CODING AND SOLUTIONING

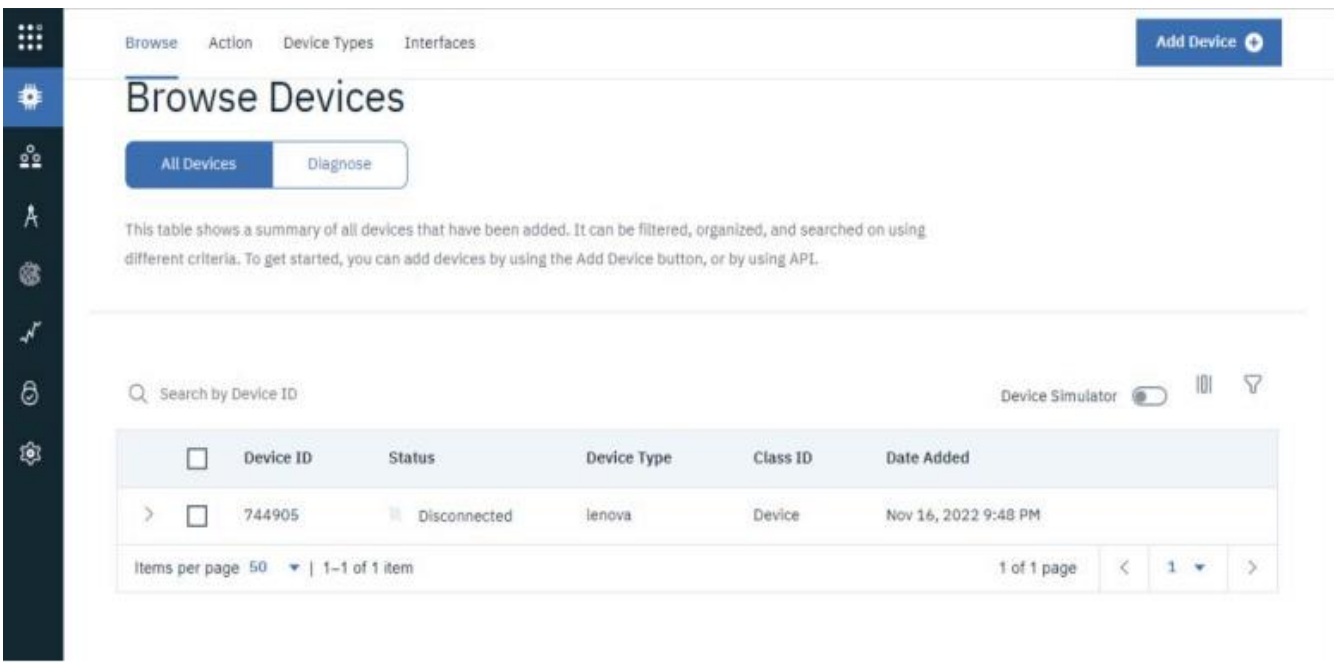
7:1: Feature 1

An account has been created on the respective platforms like IBM Cloud, IBMWatson, Node-Red, MIT App Inventor.

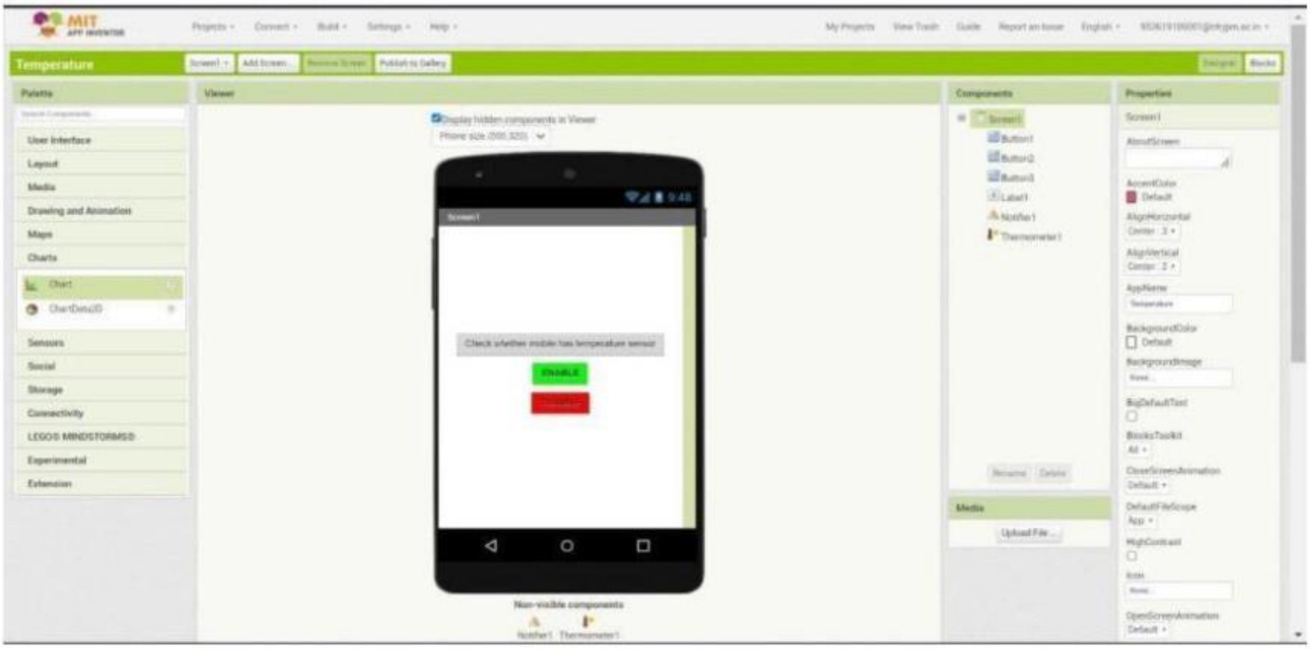
IBM CLOUD DASHBOARD:



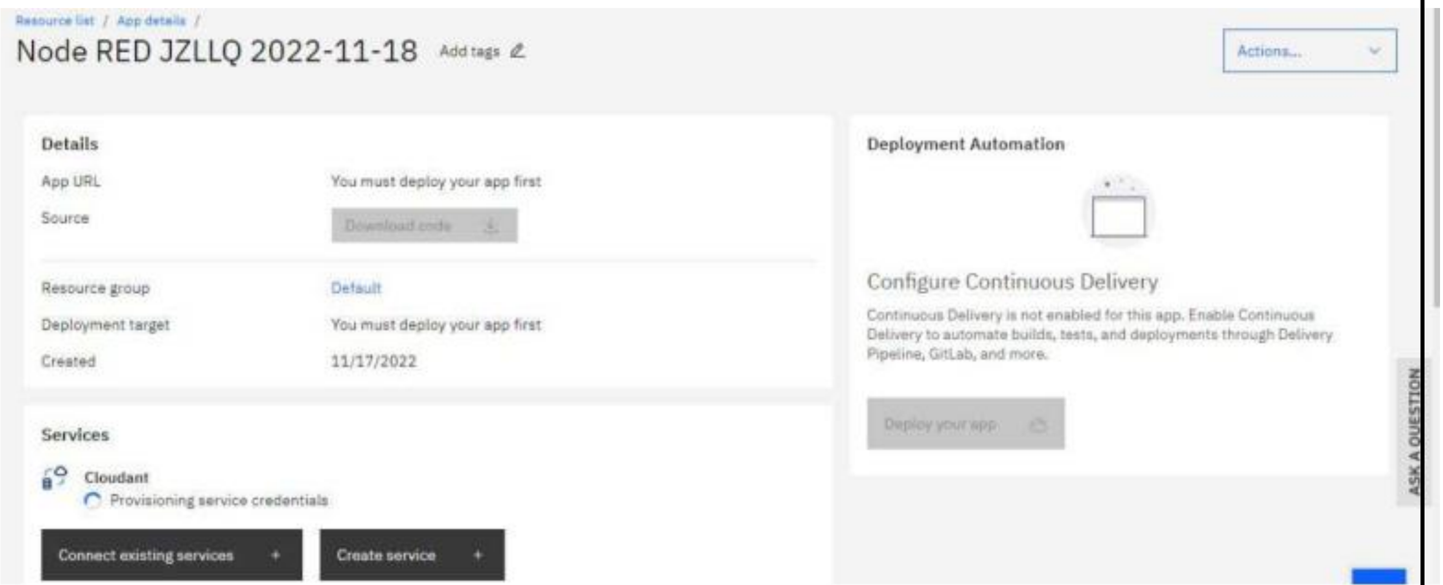
IBM WATSON IOT PLATFORM:



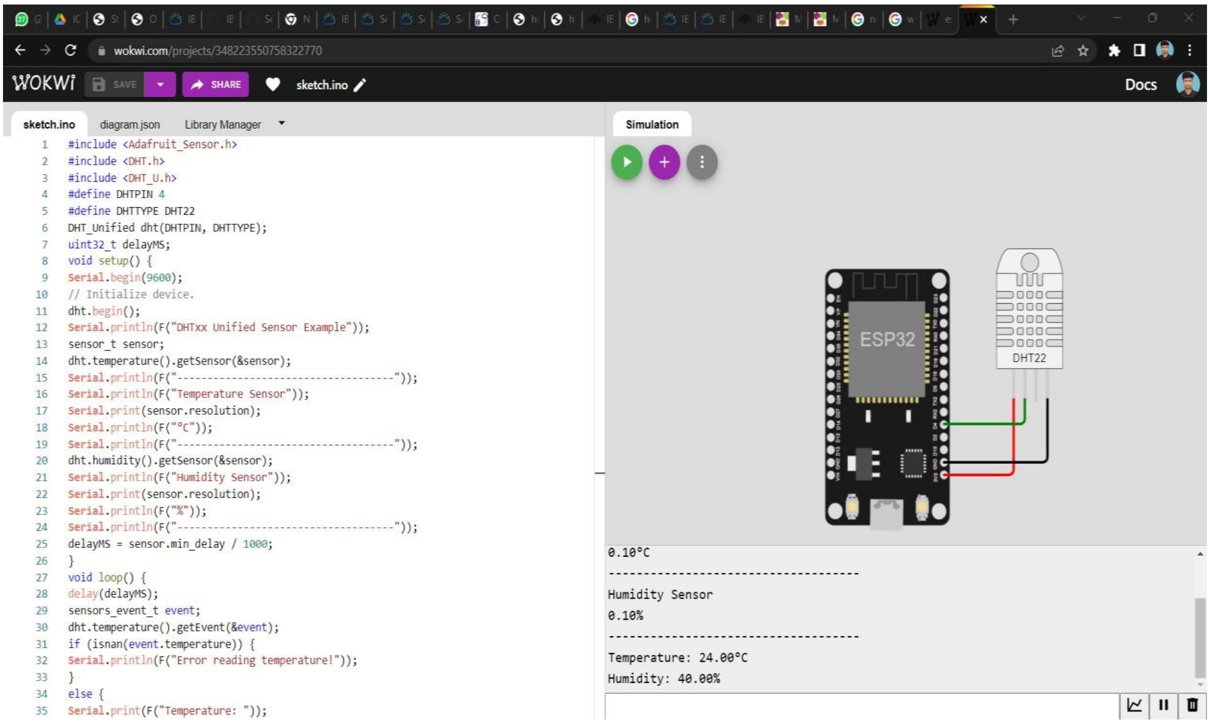
MIT APP INVENTOR:



NODE-RED:



MEASURING TEMPERATURE AND HUMIDITY VALUES WITH ESP 32:



Program:

```
#include <Adafruit_Sensor.h>#include
<DHT.h>
#include <DHT_U.h> #define
DHTPIN 4 #define DHTTYPE
DHT22
DHT_Unified dht(DHTPIN, DHTTYPE);
uint32_t delayMS;
void setup() {
Serial.begin(9600);
// Initialize device.dht.begin();
Serial.println(F("DHTxx Unified Sensor Example"));sensor_t sensor;
dht.temperature().getSensor(&sensor);
Serial.println(F(" -------------------------------------------------------------------------- "));
Serial.println(F("Temperature Sensor"));
Serial.print(sensor.resolution); Serial.println(F("°C"));
Serial.println(F("          ---------------------------------------------------------"));
dht.humidity().getSensor(&sensor); Serial.println(F("Humidity Sensor"));
Serial.print(sensor.resolution);
Serial.println(F("%"));
Serial.println(F("          ---------------------------------------------------------"));
delayMS = sensor.min_delay / 1000;
```

```
}
void loop() { delay(delayMS);
sensors_event_t event;
dht.temperature().getEvent(&event);if
(isnan(event.temperature)) {
Serial.println(F("Error reading temperature!"));
}
else { Serial.print(F("Temperature: "));
Serial.print(event.temperature);
Serial.println(F("°C"));
}
dht.humidity().getEvent(&event);
if (isnan(event.relative_humidity)) {
Serial.println(F("Error reading humidity!"));
}
else {
Serial.print(F("Humidity: ")); Serial.print(event.relative_humidity);
Serial.println(F("%"));
}
}
```

Sensor Interfacing:

Program:

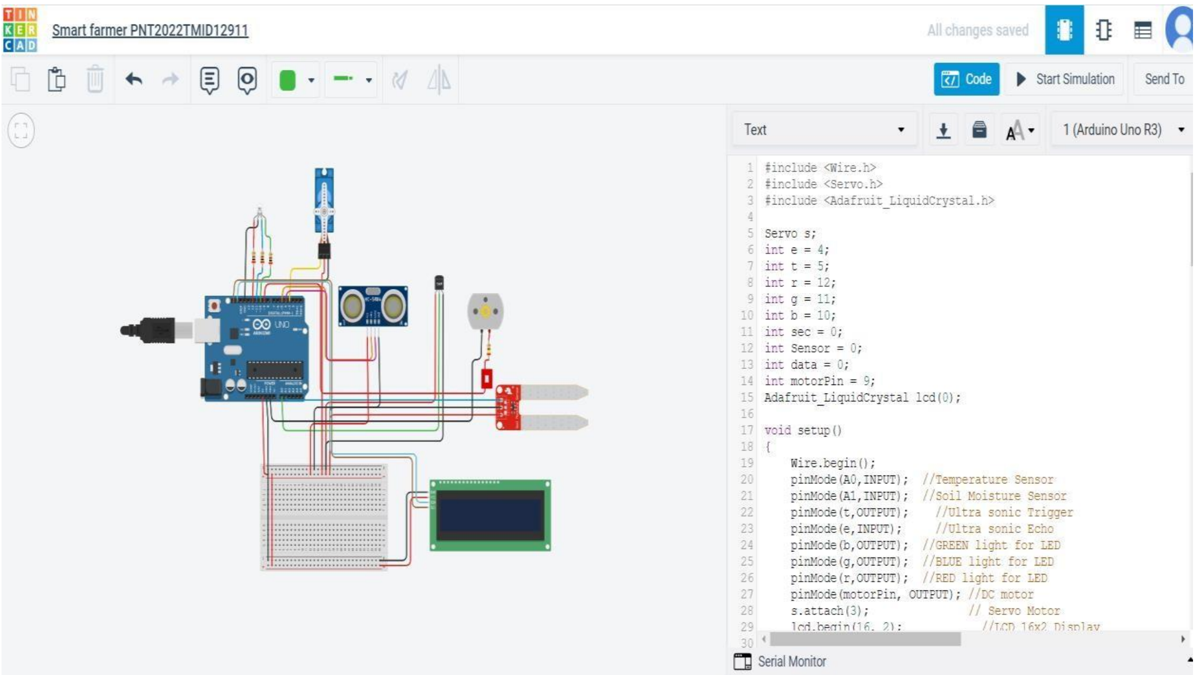
```
#include <Wire.h> #include
<Servo.h>
#include <Adafruit_LiquidCrystal.h>Servo s;
int e = 4; int t =
5; int r = 12; int g
= 11; int b = 10;
int sec = 0;
int Sensor = 0; int data
=   0; int motorPin = 9;
Adafruit_LiquidCrystal lcd(0);
```

```
void setup()
{
        Wire.begin();
        pinMode(A0,INPUT);              //Temperature Sensor
        pinMode(A1,INPUT);              //Soil Moisture Sensor
        pinMode(t,OUTPUT);                //Ultra sonic Trigger
        pinMode(e,INPUT);                 //Ultra sonic Echo
        pinMode(b,OUTPUT);              //GREEN light for LED
        pinMode(g,OUTPUT);              //BLUE light for LED
        pinMode(r,OUTPUT);              //RED light for LED
        pinMode(motorPin, OUTPUT); //DC motor s.attach(3);
                                          // Servo Motor
        lcd.begin(16, 2);                                  //LCD 16x2 Display
        lcd.setBacklight(0);
        Serial.begin(9600);
}

float readDistanceCM(){
        digitalWrite(t, LOW);
        delayMicroseconds(2);
        digitalWrite(t, HIGH);
        delayMicroseconds(10);
        digitalWrite(t, LOW);
        int duration = pulseIn(e, HIGH);return duration
        * 0.034 / 2;
}

void loop(){
        //Soil Moisture:
        Sensor = analogRead(A1);                              //Reads data from SoilMoisture
sensor
        data = map(Sensor,0, 1023, 0, 100); //Low analog value indicates HIGH moisture
level and High analog value indicatesLOW moisture level
        //data = map(analogValue,fromLOW,fromHIGH,toLOW,toHIGH)
        Serial.print("Soil Moisture value:"); Serial.println(data);
        //'data = 0' indicates wet and 'data = 100' indicates dry

        //Temperature:
        double a = analogRead (A0);                        //Reads data from
Temperature sensor
        double t = (((a/1024)*5)-0.5)*100;
        Serial.print("Temperature value:");Serial.println(t);

        //Ultrasonic sensor:
        float distance = readDistanceCM();
        Serial.print("Measured distance: ");
        Serial.println(readDistanceCM());
```

```
  //LCD Display:
  lcd.setBacklight(1);lcd.clear();

//Conditions:
if (t>40 & t<50){ digitalWrite(b,0);
        digitalWrite(g,1); digitalWrite(r,0);
        s.write(90); digitalWrite(motorPin, HIGH);
        Serial.println("Water Partially Flows");
}

else if (t>50){ digitalWrite(b,1);
        digitalWrite(g,1); digitalWrite(r,0);
        s.write(180); digitalWrite(motorPin,
        HIGH);
        Serial.println("Water Fully Flows");
}

else if (t>30 & data<30){
        digitalWrite(b,1);
        digitalWrite(g,1);
        digitalWrite(r,0);
          s.write(90); digitalWrite(motorPin, HIGH);
        Serial.println("Water Partially Flows");
}

else if (data<50){ digitalWrite(b,0);
        digitalWrite(g,0); digitalWrite(r,1);
        s.write(90); digitalWrite(motorPin, HIGH);
        Serial.println("Water Partially Flows");
}

else if (distance < 10){
        digitalWrite(b, 0);
        digitalWrite(g, 0);
        digitalWrite(r, 1); s.write(0);
        digitalWrite(motorPin, LOW);
        Serial.println("Water Does Not Flow");lcd.clear();
            lcd.println("Drain the water");
}
```

```
else{
        digitalWrite(b,1); digitalWrite(g,0);
        digitalWrite(r,0); s.write(0);
        digitalWrite(motorPin, LOW);
        Serial.println("Water Does Not Flow");
  }
   lcd.setCursor(0,0);
   lcd.print("Temp:");
  lcd.println(t); lcd.println("degree");
   lcd.setCursor(0,1); lcd.print("Soil
   Moisture:");
  lcd.println(data);lcd.println("%");
  Serial.println("            ------------------------------------------------------------
----------------");
  delay(1000);
}
```

TinkerCad Circuit:



7:2:Feature 2

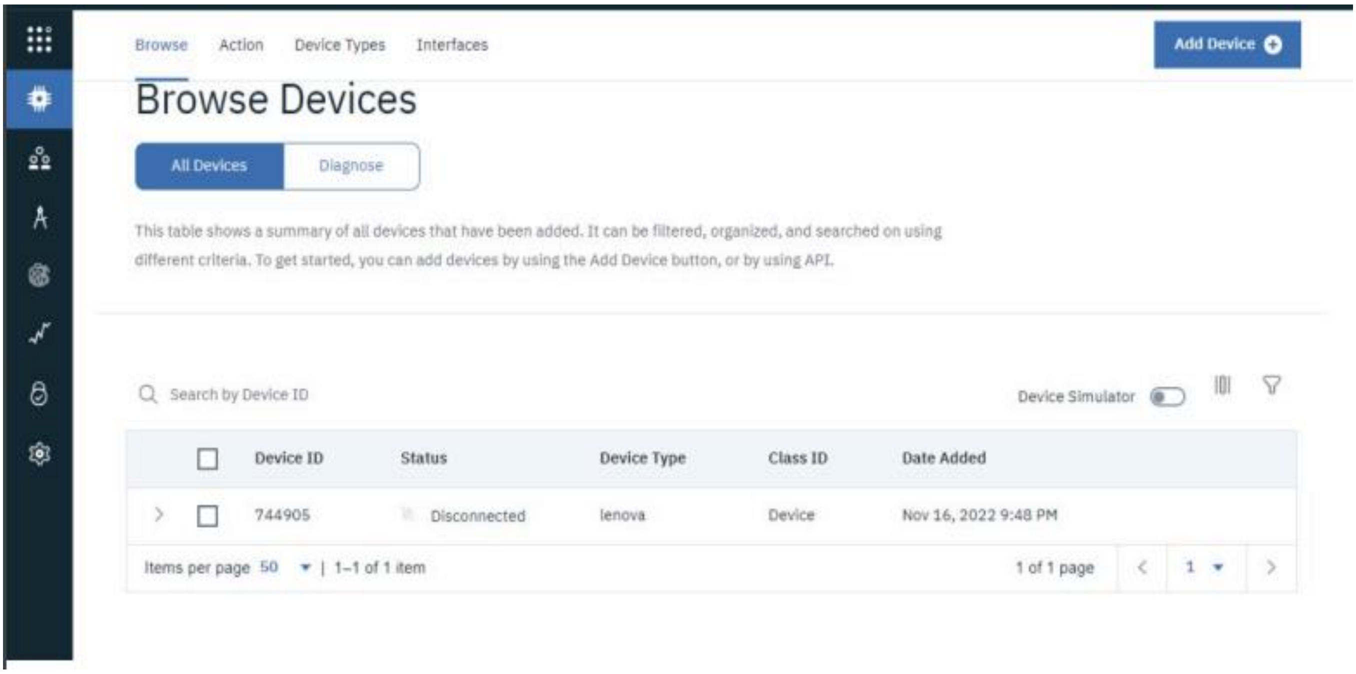
PROGRAM FOR SENDING TEMPERATURE AND HUMIDITY VALUES USINGMQTT PROTOCOL:

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#define DHTPIN 15
#define DHTTYPE DHT22
#define LED 2

DHT dht (DHTPIN, DHTTYPE);
void callback(char* subscribetopic, byte* payload, unsignedint payloadLength);
#define ORG "tu4jce"//IBM ORGANITION ID
#define DEVICE_TYPE "NodeMCU"//Device type
#define DEVICE_ID "12345"//Device ID
#define TOKEN "2W?*d5U83t+ICiNhyJ"                    //Token
String data3;
float h, t;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
 char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//  -------------------------------------------------------
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
void setup()
{
    Serial.begin(115200);dht.begin();
    pinMode(LED,OUTPUT); delay(10);
    Serial.println(); wificonnect();
    mqttconnect();
}


void loop()
{
```

```
    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();
    }
}
void PublishData(float temp, float humid) {mqttconnect();
    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");
    } 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()
{
    Serial.println();
```

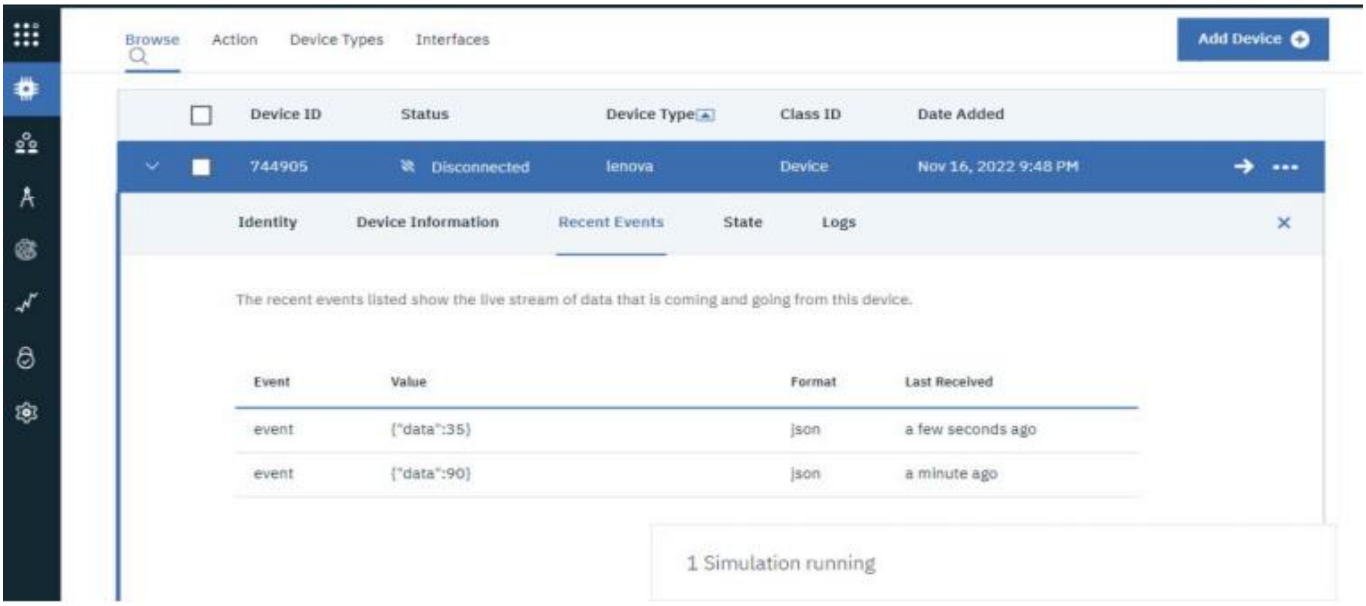
```
    Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST",   "",   6);
    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, unsignedint 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=="lighton")
    {
Serial.println(data3);
digitalWrite(LED,HIGH);
    }
    else
    {
Serial.println(data3);
digitalWrite(LED,LOW);
    }
data3="";
}
```

Connecting to IBM Watson IoT platform

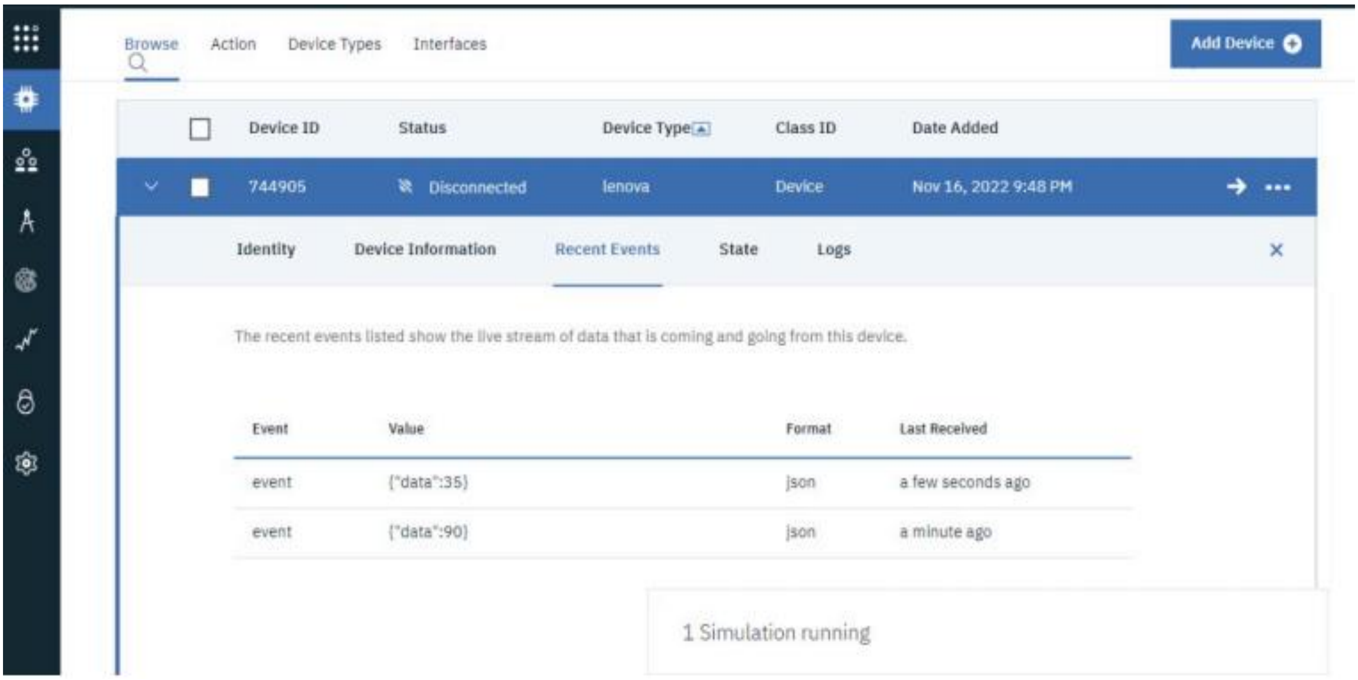


Publishing temperature and humidity values to the IBM Watson IoT platform

IBM Watson IoT platform:



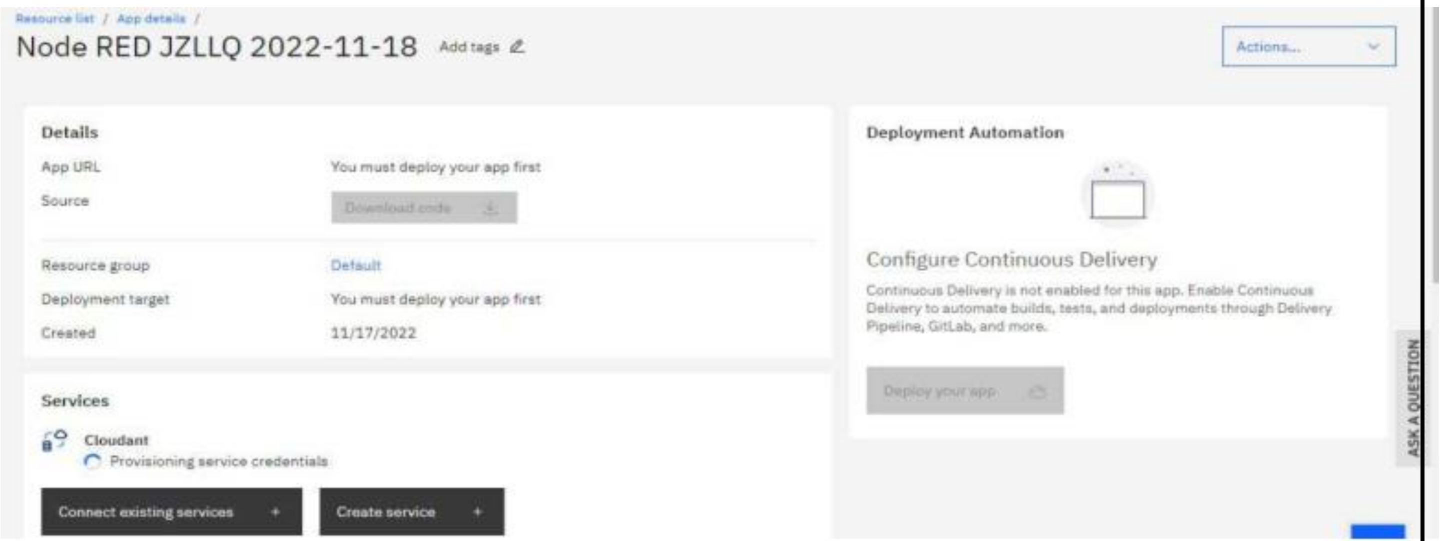
Connected Status in IBM Watson IoT platform



Recent Events in IBM Watson IoT platform

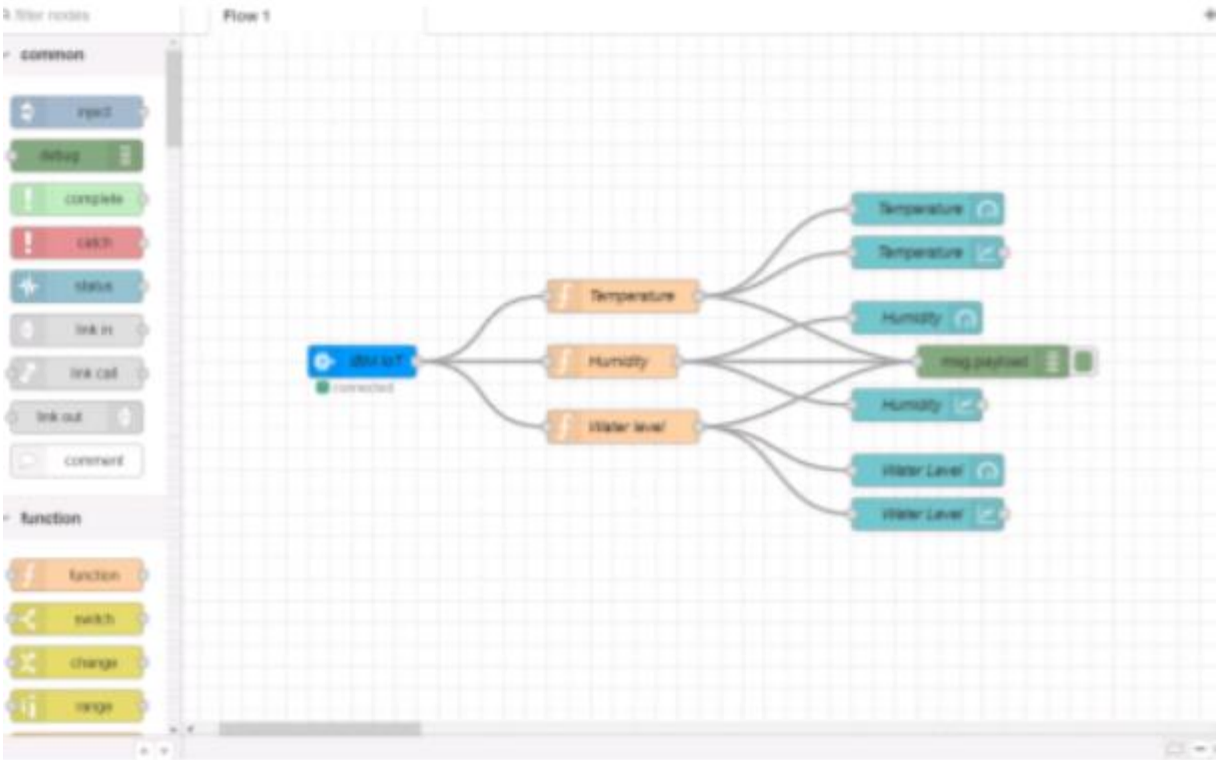
Once the sensor data like temperature and humidity gets updated in the IBM Watson IoT platform, those sensor data's will be available under recent events.

Node-RED Service Creation in IBM Cloud:



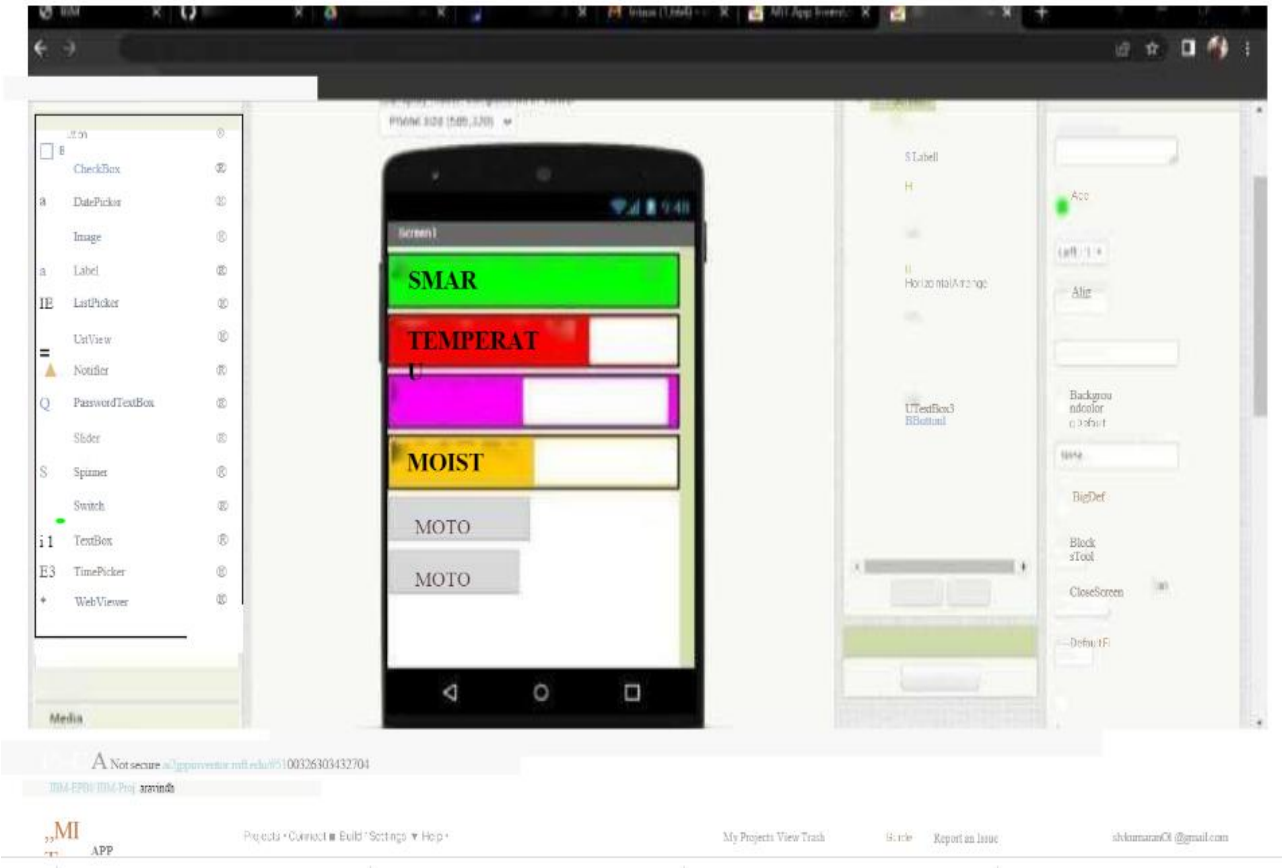
Node-RED service was created in the IBM cloud. After establishing Node-RED service, IBMIoT was installed in the Node-RED platform. Then, IBM Watson IoT platform was connected with Node-RED and the values in the IBM Watson IoT platform gets updated to the Node-RED in json file format.

Node-RED flow for getting sensor values from IBM Watson IoT Platform:

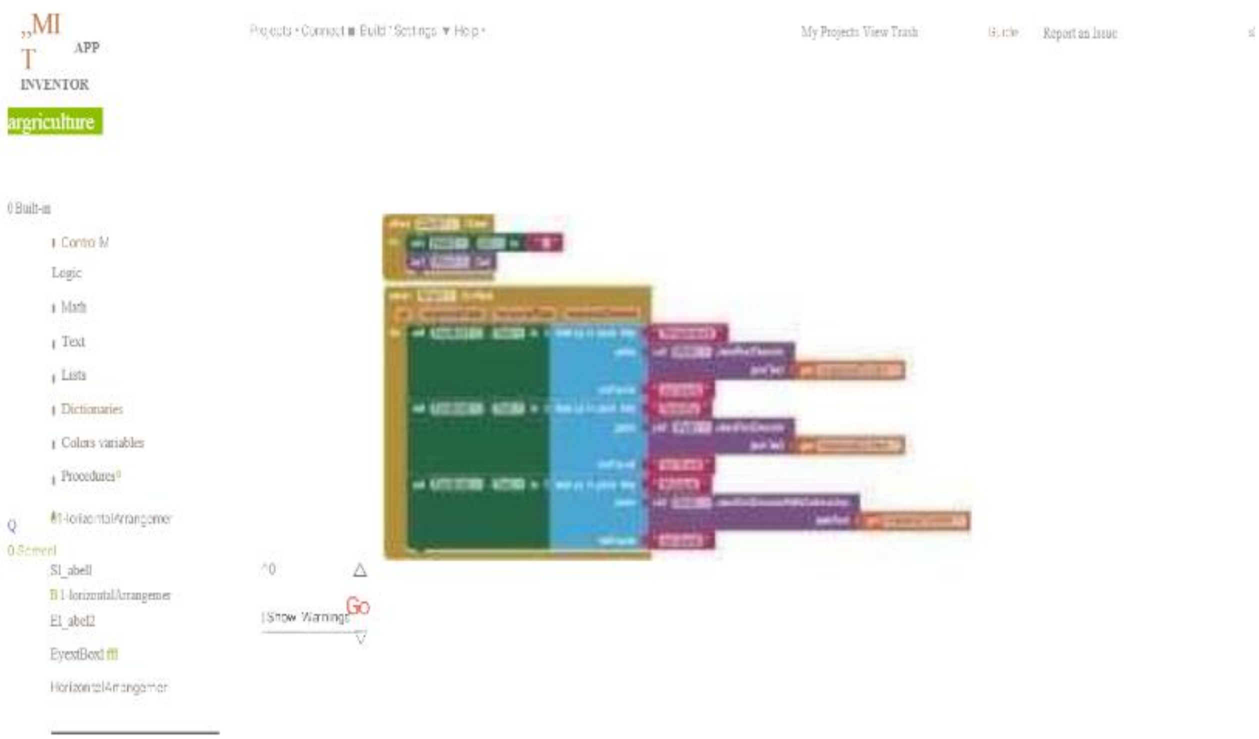


Displaying Temperature and Humidity values over the URL using http response:

MIT App inventor Front End:



MIT App Inventor Back end:



8: CONCLUSION

IoT based Smart farming system has been designed and mobile application was developed to control and monitor the field. The System has high efficiency and accuracy in fetching the live data of temperature and soil moisture. This IoT based smart farming System will assist farmers in increasing the agriculture yield and take efficient care of food production as the System will always provide helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with more than accurate results.

9: FUTURE WORKS

Crop predication plays a key role, it helps the farmer to decide future plan regarding the production of the crop, its storage, marketing techniques and risk management. To predict production rate of the crop artificial network use information collected by sensors from the farm. This information includes parameters such as soil, temperature, pressure, rainfall, and humidity. The farmers can get an accurate soil data either by the dashboard or a customized mobile application. Future work would be focused more on increasing sensors on this system to fetch more data especially with regard to Pest Control and by also integrating GPS module in this system to enhance this Agriculture IoT Technology to full-fledged Agriculture Precision ready product.

10 : APPENDIX

Source Code

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#define DHTPIN 15
#define DHTTYPE DHT22
#define LED 2

DHT dht (DHTPIN, DHTTYPE);
void callback(char* subscribetopic, byte* payload, unsignedint payloadLength);
#define ORG "tu4jce"//IBM ORGANITION ID
#define DEVICE_TYPE "NodeMCU"//Device type
#define DEVICE_ID "12345"//Device ID
#define TOKEN "2W?*d5U83t+ICiNhyJ"                    //Token
String data3;
float h, t;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
 char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/command/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
// -------------------------------------------------------
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
void setup()
{
    Serial.begin(115200);dht.begin();
    pinMode(LED,OUTPUT); delay(10);
    Serial.println(); wificonnect();
    mqttconnect();
```

```
}void loop()
{

    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();
    }
}
void PublishData(float temp, float humid) {mqttconnect();
    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");
    } 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()
{
Serial.println();
```

```
      Serial.print("Connecting to ");

    WiFi.begin("Wokwi-GUEST",   "",   6);
    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, unsignedint 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=="lighton")
    {
Serial.println(data3);
digitalWrite(LED,HIGH);
    }
    else
    {
Serial.println(data3);
digitalWrite(LED,LOW);
    }
data3="";
}
```

Reference video

https://youtu.be/795b3I2yqRI