SmartFarmer - IoT Enabled Smart Farming Application

Team id: PNT2022TMID08745

SUBMITTED BY

Santhoshkumar S	727619BEC041
Sethuram R	727619BEC081
Nesamani S	727620BEC309
Hari Krishna R	727620BEC323

In partial fulfilment for the award of the degree of
BACHELOR OF ENGINEERING
in
ELECTRONICS AND COMMUNICATION ENGINEERING
Dr . MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY An Autonomous Institution Affiliated to ANNAUNIVERSITY CHENNAI – 600 025

1.INTRODUCTION :

1.1 Project Overview

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself. Agriculture implements IoT through the use of robots, drones, sensors, and computer imaging integrated with analytical tools for getting insights and monitoring the farms. Placement of physical equipment on farms monitors and records data, which is then used to get valuable insights.

1.2 Purpose

Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology – including big data, the cloud and the internet of things (IoT) – for tracking, monitoring, automating and analyzing operations.

2. LITERATURE SURVEY

Zuraida Muhammad, Muhammad Azri Asyraf Mohd Hafez, Nor Adni MatLeh, Zakiah Mohd Yusoff , Shabinar Abd Hamid [1] The term "Internet of Things" refers to the connection of objects, equipment, vehicles, and other electronic devices to a network for the purpose of data exchange (IoT). The Internet of Things (IoT) is increasingly being utilised to connect objects and collect data. As a result, the Internet of Things' use in agriculture is crucial. The idea behind the project is to create a smart agriculture system that is connected to the internet of things. The technology is combined with an irrigation system to deal with Malaysia's variable weather. This system's microcontroller is a Raspberry Pi 4 Model B. The temperature and humidity in the surrounding region, as well as the moisture level of the soil, are monitored using the DHT22 and soil moisture sensor. The data will be available on both a smartphone and a computer. As a result, Internet of Things (IoT) and Raspberry Pi-based Smart Agriculture Systems have a significant impact on how farmers work. It will have a good impact on agricultural productivity as well. In Malaysia, employing IoT-based irrigation systems saves roughly 24.44 percent per year when compared to traditional irrigation systems. This would save money on labour expenditures while also preventing water waste in daily needs.

Divya J., Divya M.,Janani V. [2] Agriculture is essential to India's economy and people's survival. The purpose of this project is to create an embedded-based soil monitoring and irrigation system that will reduce manual field monitoring and provide information via a mobile app. The method is intended to help farmers increase their agricultural output. A pH sensor, a temperature sensor, and a humidity sensor are among the tools used to examine the soil. Based on the findings, farmers may plant the best crop for the land. The sensor data is sent to the field manager through Wi-Fi, and the crop advice is created with the help of the mobile app. When the soil temperature is high, an automatic watering system is used. The crop image is gathered and forwarded to the field manager for pesticide advice.

2.1 Existing Problem

As we can see, the use cases for IoT in agriculture are endless. There are many ways smart devices can help you increase your farm's performance and revenue. However, agriculture IoT apps development is no easy task.

2.2 References

[1] https://www.ws.com/analytics/tfarming powered by analytics/.

[2] https://www.ibm.com/case-studies/smartrural-cloud-analystics-farming

[3] https://www.ibm.com/case-studies/farming-watson-cloud

[4] http://www.ewaterautosys.com/water-automation-system.html

[5]http://www.wplawinc.com/agricultural-irrigation-blog/the-most-common-problems-with-farm-irrigationsystems

[6]https://www.pwc.com/us/en/increasing-it-effectiveness/assets/future-of-the-internet-of-things.pdf

[7] IOT based Smart Irrigation System SrishtiRawalDepartment of Computer Science, VIT University rawal-2017-ijca-913001.pdf

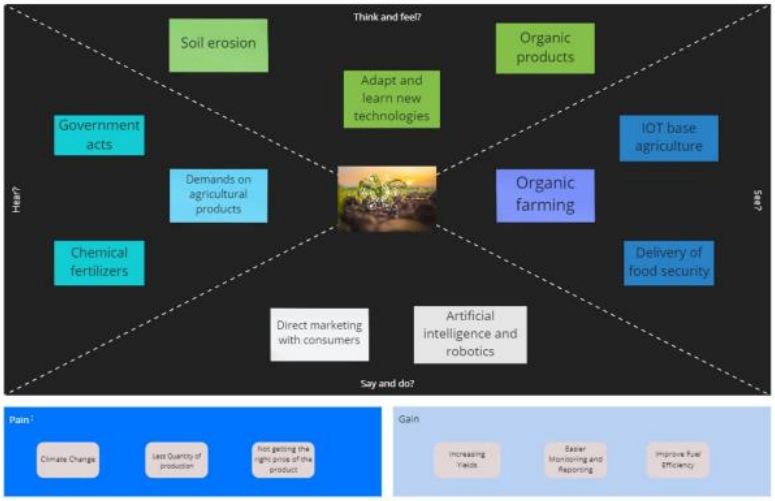
[8] Thingspeak : https:// thingspeak.com

2.3 Problem Statement Definition

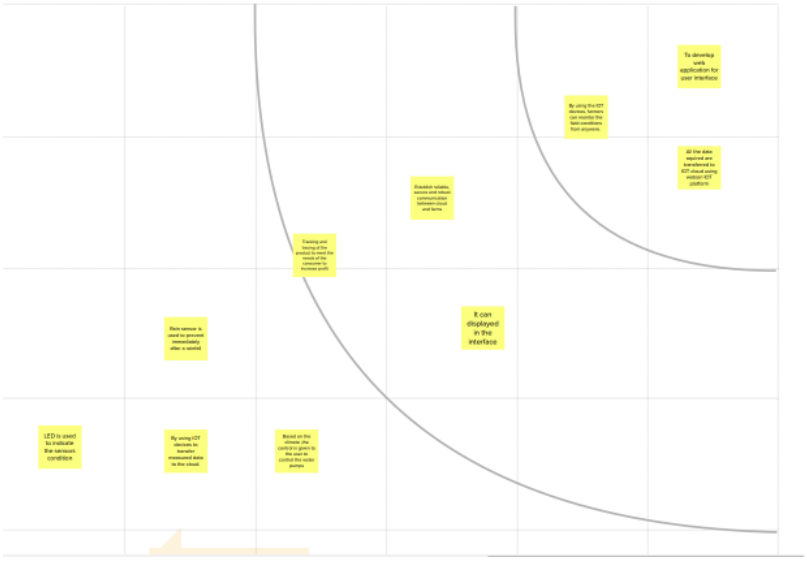
The sensor near a crop will help us to identify and alert the user whether it needs any more water or fertilizer at that moment monitored through application. Thus increasing difficulties in farming leads to the way of inventing many new technologies that will help the farmers to take some precautions to avoid losses in cultivation.

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



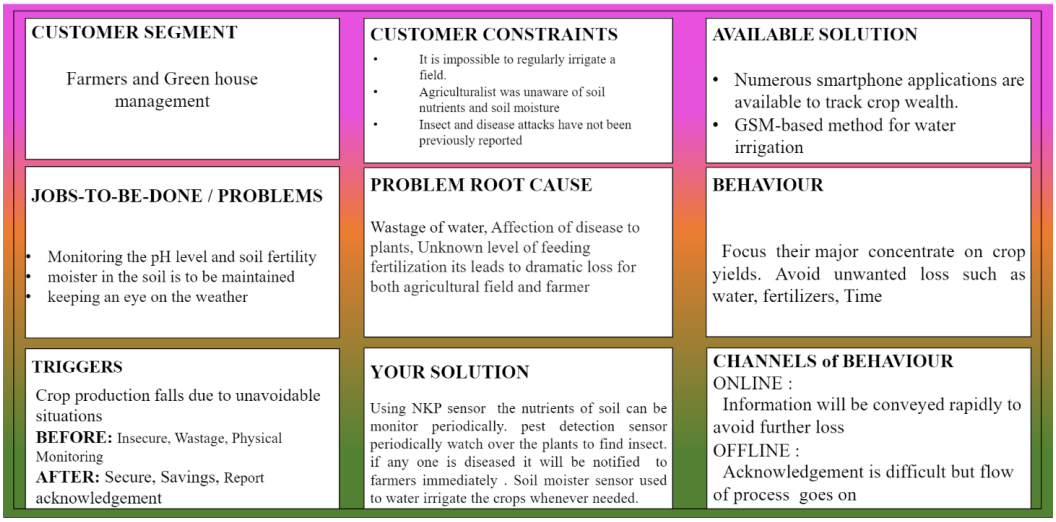
3.3 Proposed Solution

Proposed Solution

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The biggest challenge faced by Iot in agriculture are lack of information, cost analysis and quantity of fertilizers/seeds needed at specifc time. Monitoring climate change and take any precautions to avoid the spoilage of yields. Analyze the growth and intimate any need of fertilizer or water for each area of farming. Indicate of irrigation periodically. Reduction of greenhouse gas is biggest challenges in IOT.
2.	Idea / Solution description	The basic idea is to monitor the crop, weather, soil quality and irrigation control by using sensors. And through cloud publish the sensed data in web application to notify the farmer about their field.
3.	Novelty / Uniqueness	It gives great support for farmer and farming related activities. It reduced man force and enhancement of automated irrigation provides healthy plant growth.
4.	Social Impact / Customer Satisfaction	• Increases crop productivity • It makes crop pest free and control the irrigation for crops. • Constantly monitoring land and condition of crops. • It modernize farm management system
5.	Business Model (Revenue Model)	Revenue is generated for smart farming and IOT supported irrigation system by farmers, land lords, green house, roof top farming, private agro, organization and government.
6.	Scalability of the Solution	Our system with modernized technology allows the farmers to upgrade from traditional agriculture to modern methods and also enhances the productivity of cultivation and efficiency in yield.

3.4 Problem Solution Fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

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 Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	IoT Kit	Wi-Fi module (ESP 232)
FR-4	Sensors	Soil moisture sensor, temperature and humidity sensor, Light sensor.
FR-5	Power supply	Switched Mode Power supply (SMPS)

4.2 Non-Functional Requirements

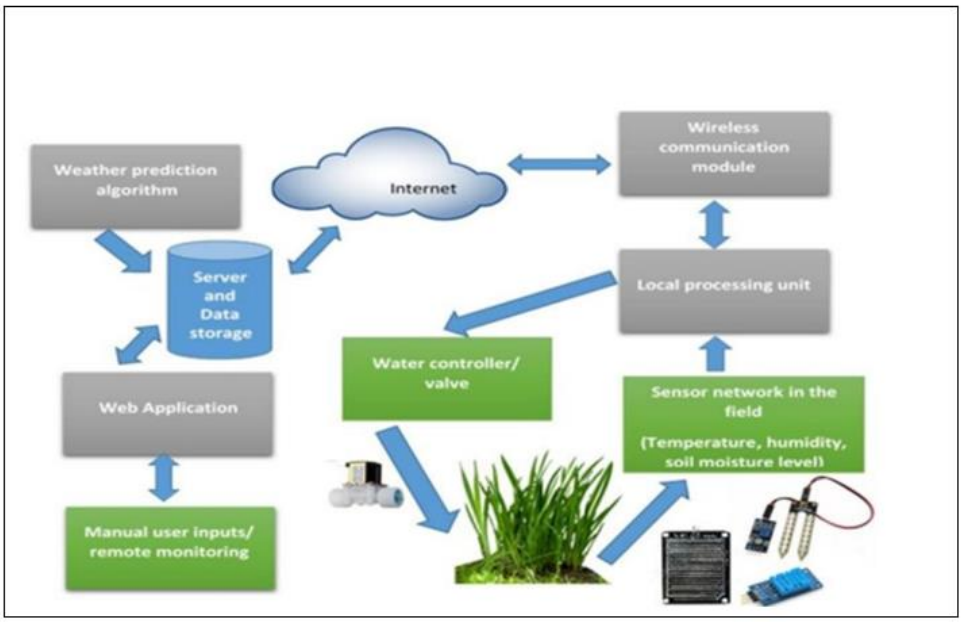
Non-functional Requirements:

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

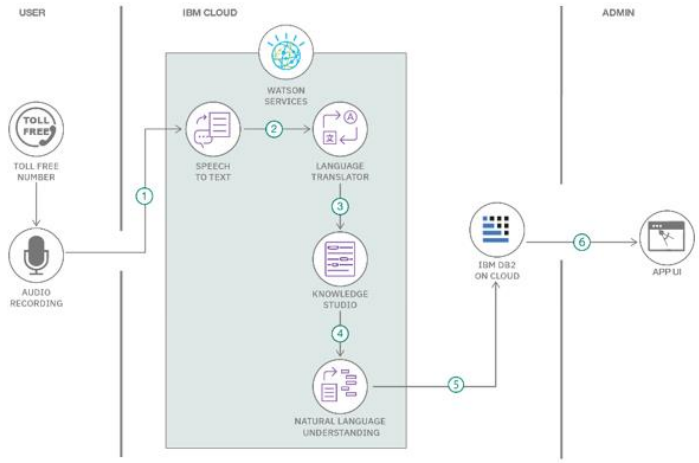
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The project is portable.
NFR-2	**Security**	It has user authentication.
NFR-3	**Reliability**	Made of reliable components and IoT material.
NFR-4	**Performance**	The project has very high performance.
NFR-5	**Availability**	Most of the sensors are available in the market.

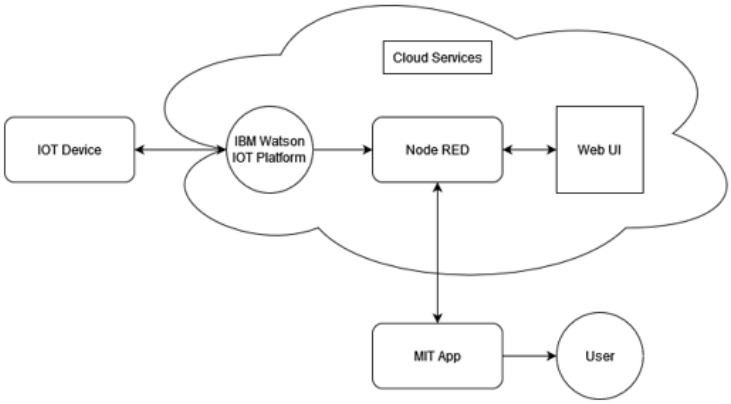
5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Solution Architecture





5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application	I can receive a confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)	Login page	USN-6	The user can log in to the web page using their password and user name.		High	Sprint-2
Customer Care Executive	Contact	USN-7	If there is any issue faced by the user, they can raise a quire in the contact or support section.		High	Sprint-1
Administrator	Admin Login	USN-8	The admin can login to see the activities of the website.		Medium	Sprint-2

6.PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation

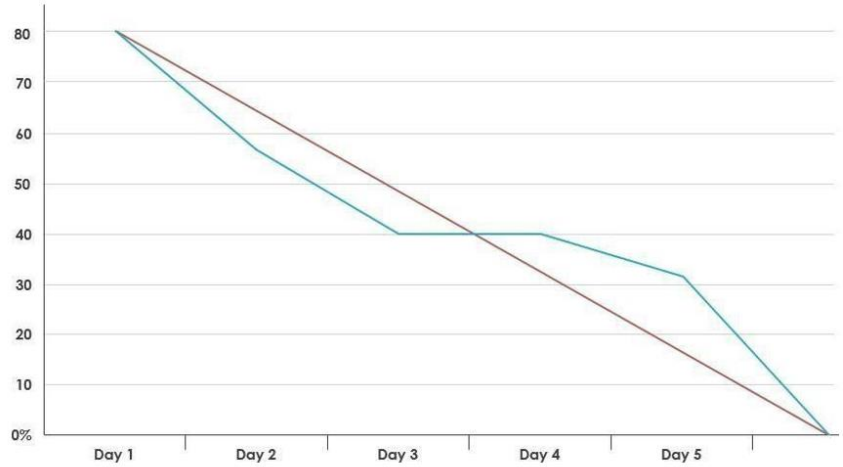
Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Member
Sprint-1	Registration (Farmer Mobile User)	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	NESAMANI S
Sprint-1	Login	USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	SETHURAM R

Sprint-2	User Interface	USN-3	As a user, I can register for the application through Facebook	3	Low	SANTHOSH KUMAR S
Sprint-1	Data Visualization	USN-4	As a user, I can register for the application through GMAIL	2	Medium	HARI KRISHNA R
Sprint-3	Registration (Farmer -Web User)	USN - 1	As a user, I can log into the application by entering email and password	3	High	NESAMANI S
Sprint - 2	Login	USN - 2	As a registered user, I need to easily login log into my registered account via the web page in minimum time	3	High	SETHURAM R
Sprint - 4	Web UI	USN - 3	As a user, I need to have a friendly user interface to easily view and access the resources	3	Medium	SANTHOSH KUMAR S
Sprint - 1	Registration (FERTILIZER Manufacturer - Web user)	USN - 1	As a new user, I want to first register using my organization email and create a password for the account.	2	High	HARI KRISHNA R

6.2 Sprint Delivery Plan

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	12	6 Days	24 Oct 2022	29 Oct 2022	20	02 NOV 2022
Sprint-2	6	6 Days	31 Oct 2022	05 Nov 2022	20	07 NOV 2022
Sprint-3	6	6 Days	07 Nov 2022	12 Nov 2022	20	16 NOV 2022
Sprint-4	6	6 Days	14 Nov 2022	19 Nov 2022	20	17 NOV 2022

6.2 Reports From JIRA



7.CODING AND SOLUTIONING

7.1 Feature 1

- IOT Device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- MIT App
- Python Code

7.2 Feature 2

- Registration Farmer User
- Registration Fertilizer User
- Farmer User Login
- Fertilizer User Login
- Verification

```
#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, unsigned int 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);
```

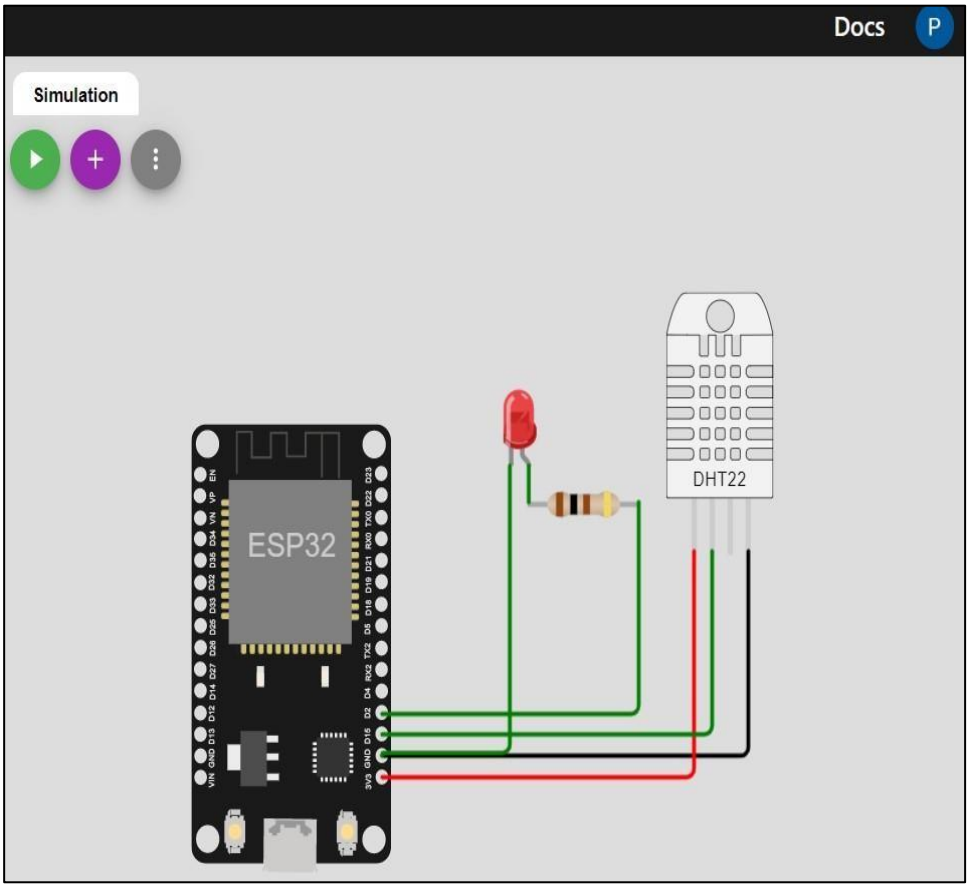
```
PublishData(
t,              h);
delay(1000);
if
(!client.loop(
))              {
mqttconnect(
);
  }

}

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, 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=="lighton")

  {

Serial.println(data3);

digitalWrite(LED,HIGH);

  }

else

  {

Serial.println(data3);

digitalWrite(LED,LOW);

  }

data3="";

}
```

8.Test Case

8.1 Test Case 1



SERIALMONITOR

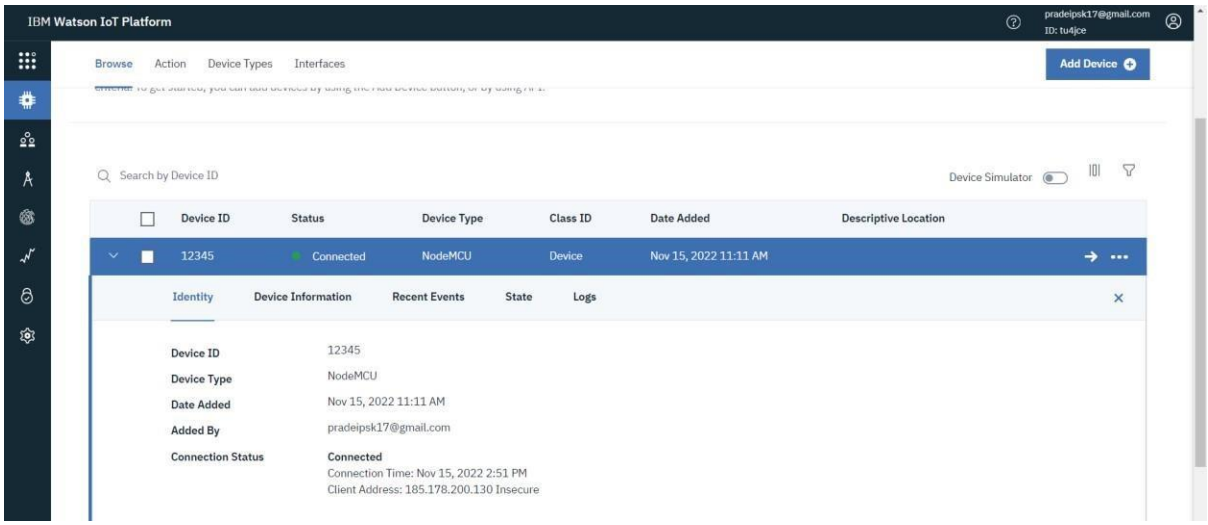
```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to tu4jce.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK
```

Connecting to IBM Watson IoT platform

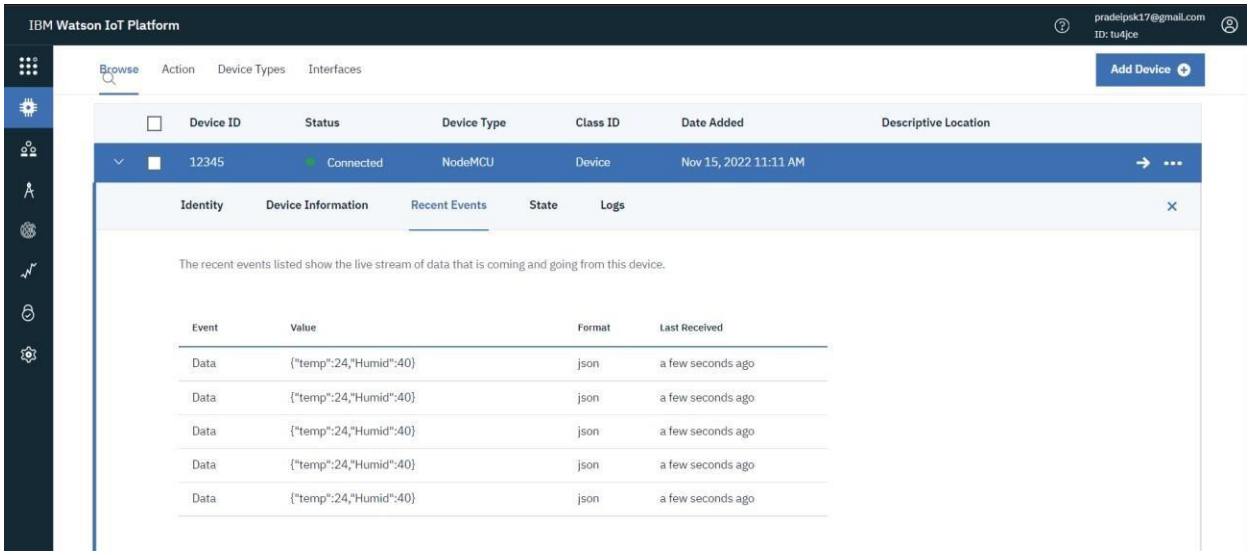
```
temp:24.00
Humid:40.00
Sending payload: {"temp":24.00,"Humid":40.00}
Publish ok
temp:24.00
Humid:40.00
Sending payload: {"temp":24.00,"Humid":40.00}
Publish ok
```

Publishing temperature and humidity values to the IBM Watson IoT platform

IBM Watson IoT platform:



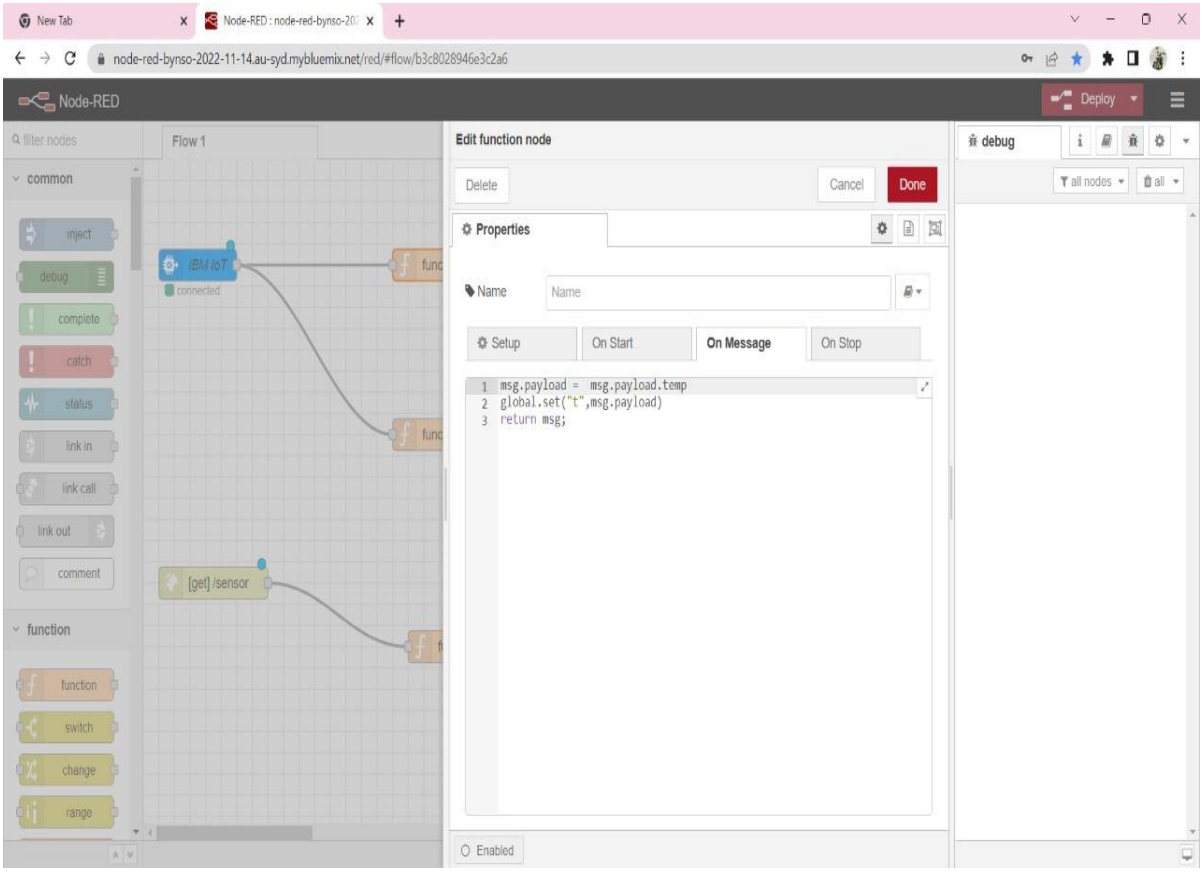
Connected Status in IBM Watson IoT platform

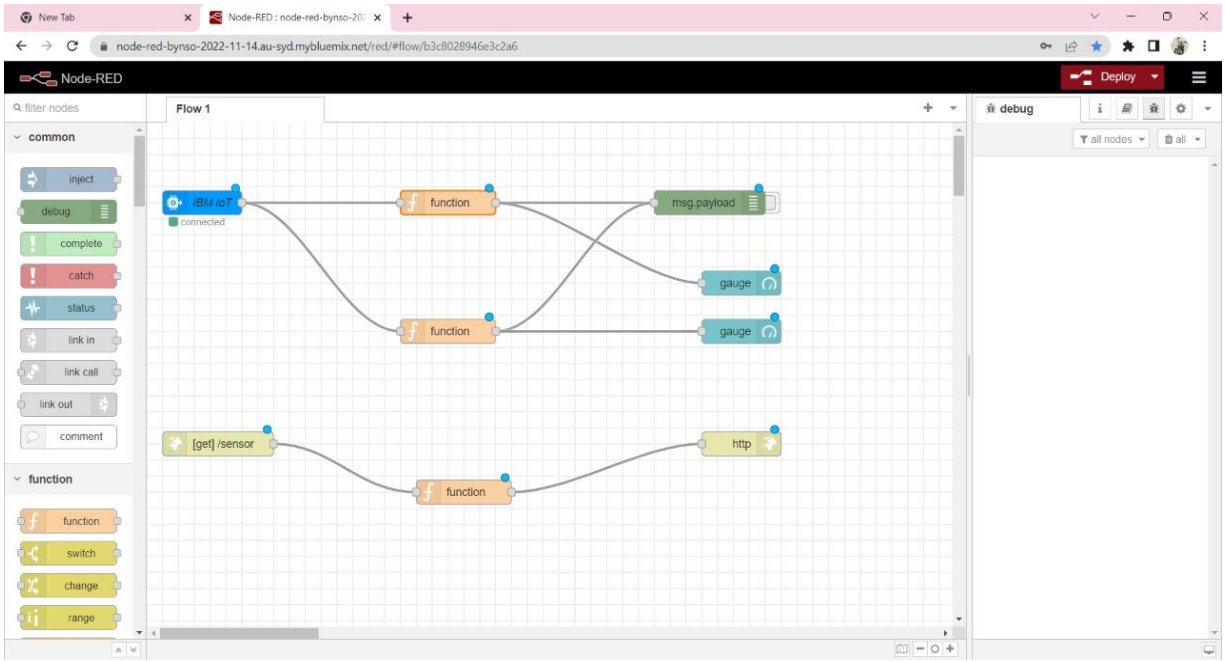


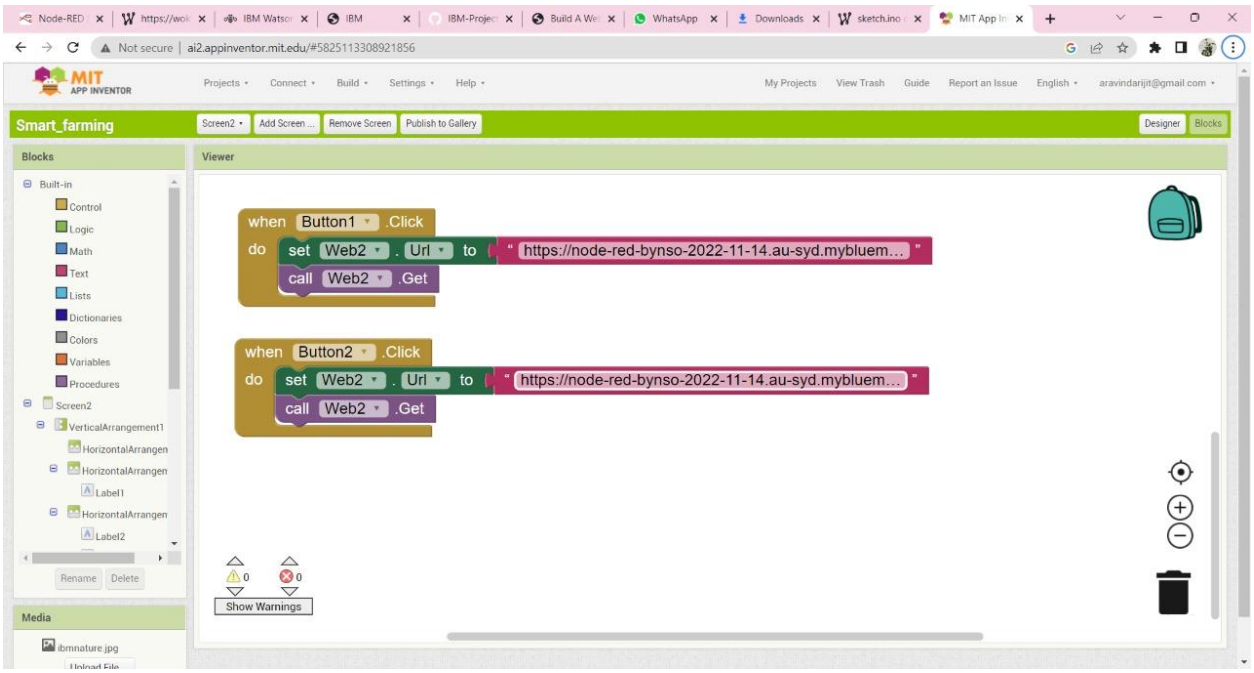
Recent Events in IBM Watson IoT platform

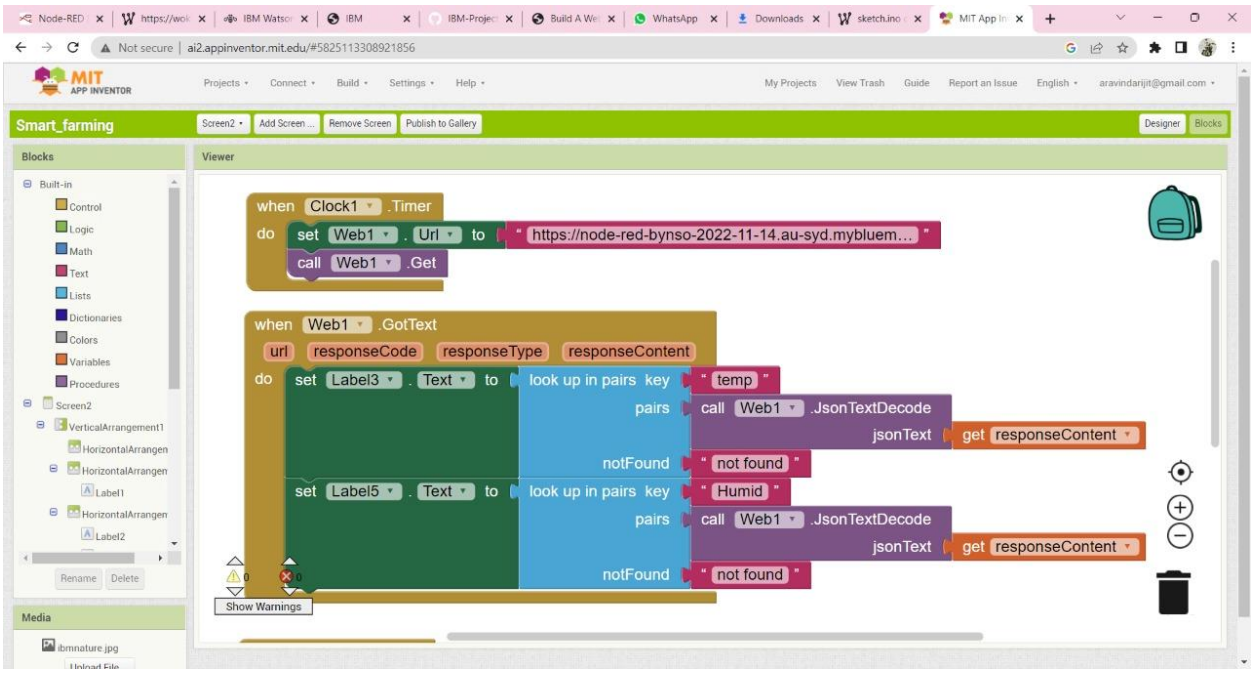
Once the sensor data like temperature and humidity gets updated in the IBM Watson IoT platform, those sensor data's will be available under recent events.

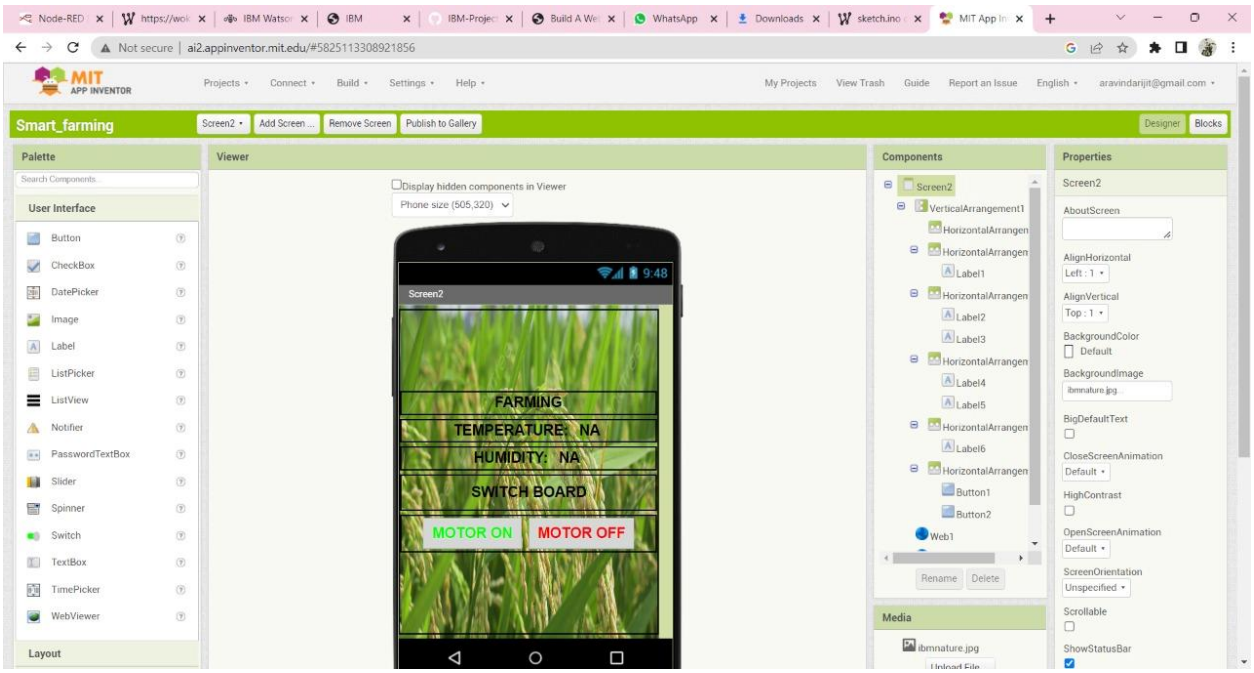
9. Results

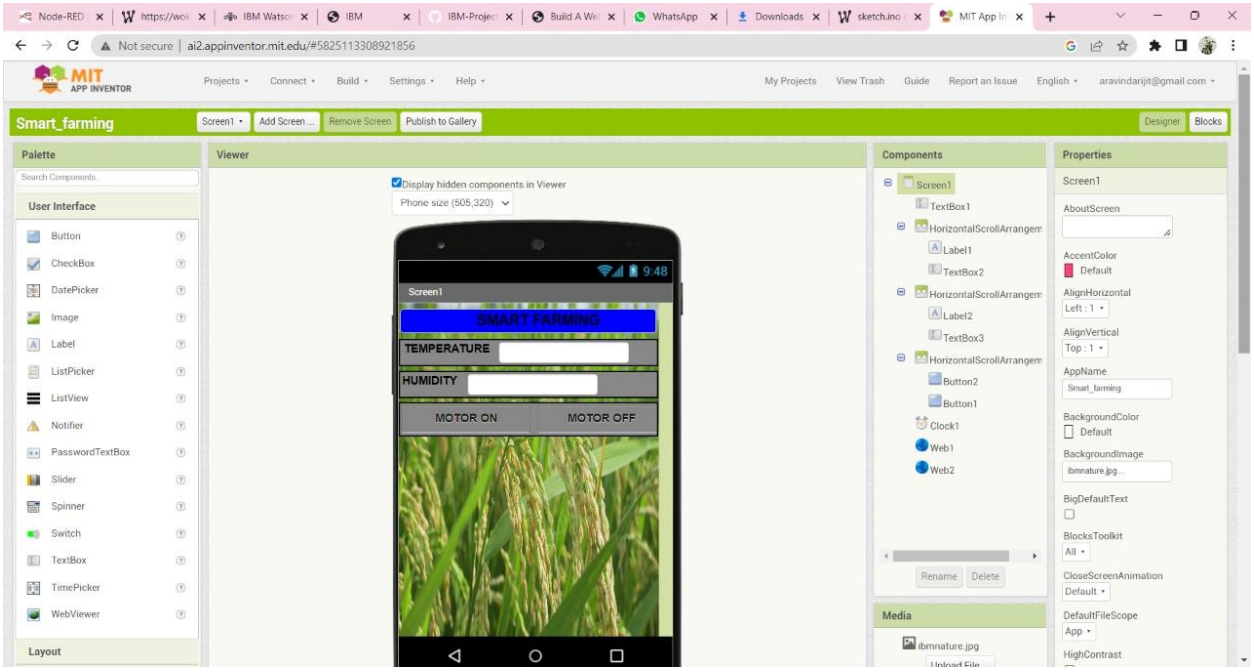

New Tab
Node-RED : node-red-bynso-20
node-red-bynso-2022-11-14.au-syd.mybluemix.net/red/#flow/b3c8028946e3c2a6
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
IBM IoT
connected
[get] /sensor
Edit function node
Delete
Cancel
Done
Properties
Name
Setup
On Start
On Message
On Stop
1 msg.payload = msg.payload.temp
2 global.set("t",msg.payload)
3 return msg;
Enabled
debug
all nodes
all

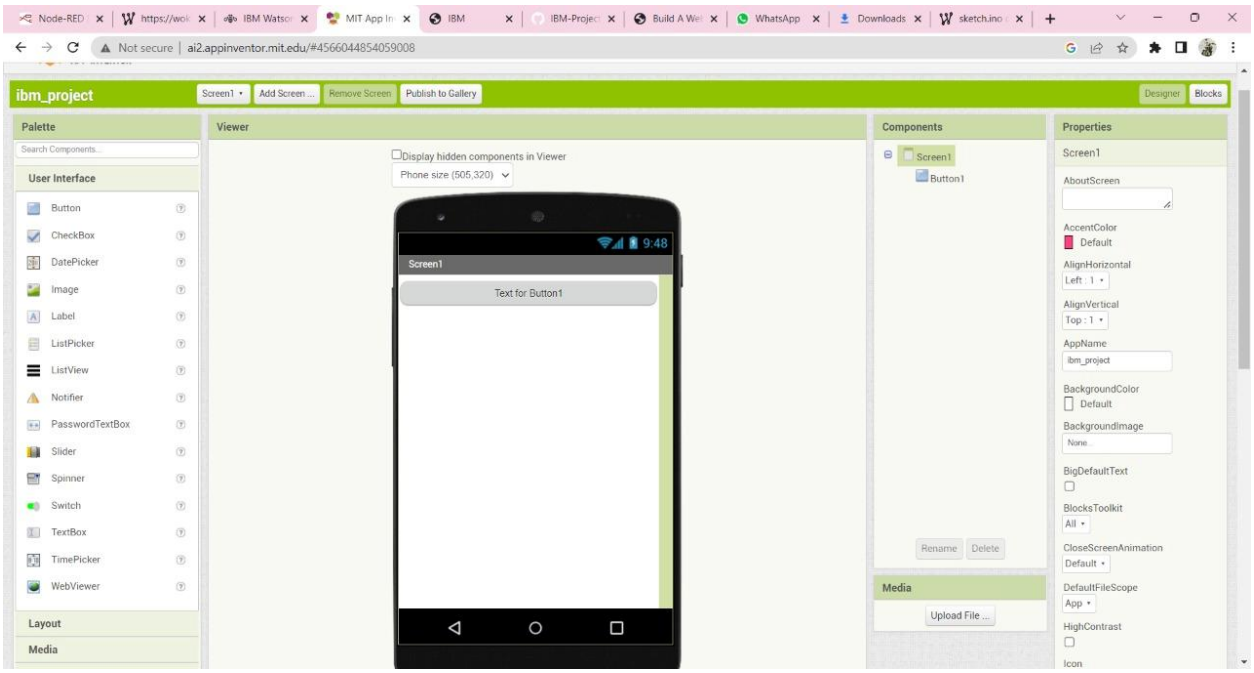

New Tab
Node-RED : node-red-bynso-20
node-red-bynso-2022-11-14.au-syd.mybluemix.net/red/#flow/b3c8028946e3c2a6
Node-RED
Deploy
filter nodes
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
IBM IoT
connected
function
msg.payload
gauge
function
gauge
[get] /sensor
http
function
debug
all nodes
all

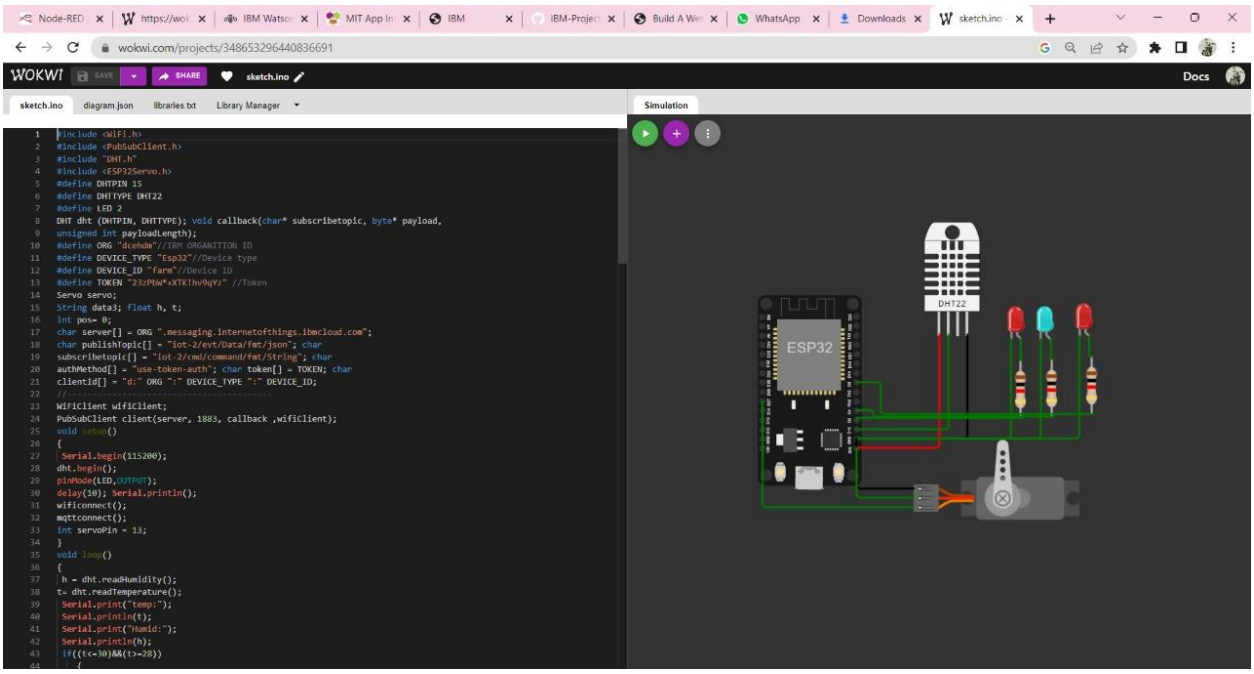

Node-RED
Not secure | ai2.appinventor.mit.edu/#5825113308921856
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
aravindarijit@gmail.com
Smart_farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
when Button1 .Click
do set Web2 . Url to " https://node-red-bynso-2022-11-14.au-syd.mybluem... "
call Web2 .Get
when Button2 .Click
do set Web2 . Url to " https://node-red-bynso-2022-11-14.au-syd.mybluem... "
call Web2 .Get
Show Warnings
Media
ibmnature.jpg


Node-RED
Not secure | ai2.appinventor.mit.edu/#5825113308921856
MIT APP INVENTOR
Projects
Connect
Build
Settings
Help
My Projects
View Trash
Guide
Report an Issue
English
aravindarijit@gmail.com
Smart_farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
when Clock1 .Timer
do set Web1 . Url to " https://node-red-bynso-2022-11-14.au-syd.mybluem... "
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Label3 . Text to look up in pairs key " temp "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
set Label5 . Text to look up in pairs key " Humid "
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound " not found "
Show Warnings
Media
ibmnature.jpg

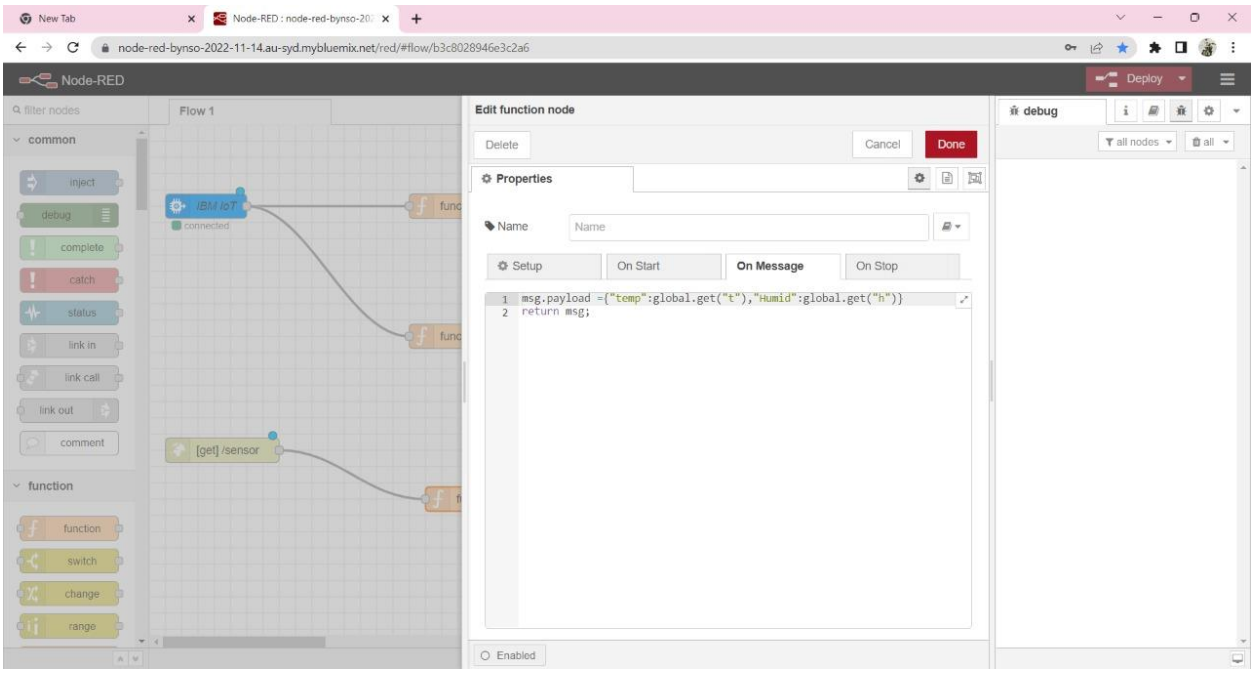

Smart_farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Viewer
Components
Properties
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen2
FARMING
TEMPERATURE: NA
HUMIDITY: NA
SWITCH BOARD
MOTOR ON
MOTOR OFF

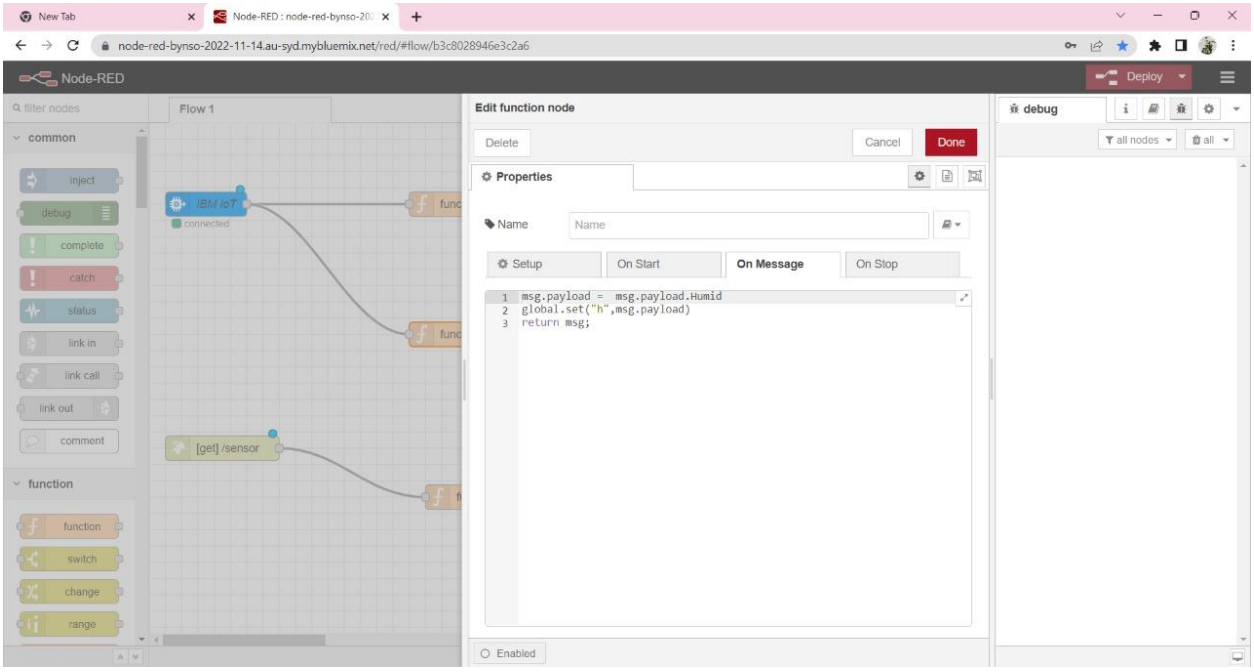

Smart_farming
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Viewer
Components
Properties
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
SMART FARMING
TEMPERATURE
HUMIDITY
MOTOR ON
MOTOR OFF






```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHT.h"
#include <ESP32Servo.h>
#define DHTPIN 15
#define DHTTYPE DHT22
#define LED 2
DHT dht (DHTPIN, DHTTYPE); void callback(char* subscribetopic, byte* payload,
unsigned int payloadLength);
#define ORG "dcehdn"//IBM ORGANITION ID
#define DEVICE_TYPE "Esp32"//Device type
#define DEVICE_ID "farm"//Device ID
#define TOKEN "23zPtW*xTKthv9qYz" //Token
Servo servo;
String data3; float h, t;
int pos= 0;
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();
int servoPin = 13;
}
void loop()
{
 h = dht.readHumidity();
t= dht.readTemperature();
 Serial.print("temp:");
 Serial.println(t);
 Serial.print("Humid:");
 Serial.println(h);
 if((t<-30)&&(t>-28))
  {
```


New Tab
Node-RED : node-red-bynso-20
node-red-bynso-2022-11-14.au-syd.mybluemix.net/red/#flow/b3c8028946e3c2a6
Node-RED
Deploy
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
IBM IoT
connected
[get] /sensor
Edit function node
Delete
Cancel
Done
Properties
Name
Setup
On Start
On Message
On Stop
msg.payload ={"temp":global.get("t"),"Humid":global.get("h")}
return msg;
Enabled
debug
all nodes
all


New Tab
Node-RED : node-red-bynso-20
node-red-bynso-2022-11-14.au-syd.mybluemix.net/red/#flow/b3c8028946e3c2a6
Node-RED
Deploy
Flow 1
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
IBM IoT
connected
[get] /sensor
Edit function node
Delete
Cancel
Done
Properties
Name
Setup
On Start
On Message
On Stop
msg.payload = msg.payload.Humid
global.set("h",msg.payload)
return msg;
Enabled
debug
all nodes
all


8:27
58%
Screen2
FARMING
TEMPERATURE: 24.4
HUMIDITY: 48
SWITCH BOARD
MOTOR ON
MOTOR OFF

10.Advantages and Disadvantages

10.1 Advantages of Smart Agriculture

- It allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc.
- Solar powered and mobile operated pumps save cost of electricity.
- Smart agriculture use drones and robots which helps in many ways. These improves data collection process and helps in wireless monitoring and control.
- It is cost effective method.
- It delivers high quality crop production.

10 .2 Drawbacks or disadvantages of Smart Agriculture

- Automated irrigation system uses only two parameters of soil like soil moisture and temperature other parameters humidity, light, air moisture, soil ph value not taken for decision making.
- Excessive seepage and leakage of water forms marshes and ponds all along the channels. The marshes and the ponds in course of time become the colonies of the mosquito, which gives rise to a disease like malaria.
- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

- The smart farming based equipments require farmers to understand and learn the use of technology. This is major challange in adopting smart agriculture farming at large scale across the countries.

11. Conclusion

A system to monitor temperature, humidity, moisture levels in the soil was designed and the project provides an opportunity to study the existing systems, along with their features and drawbacks. Agriculture is one of the most water-consuming activities. The proposed system can be used to switch the motor (on/off) depending on favourable condition of plants i.e sensor values, thereby automating the process of irrigation. which is one of the most time efficient activities in farming, which helps to prevent over irrigation or underirrigation of soil thereby avoiding crop damage. The farm owner can monitor the process online through a android App. Though this project can be concluded that there can be considerable development in farming with the use of IOT and automation.

12. Future Scope

- The performance of the system can be further improved in terms of the operating speed, memory capacity, and instruction cycle period of the microcontroller by using other high end controllers. The number of channels can be increased to interface more number of sensors which is possible by using advanced versions of controllers.

- The system can be modified with the use of a data logger and a graphical LCD panel showing the measured sensor data over a period of time.A speaking voice alarm could be used.The device can be made to perform better by providing the power supply with the help of renewable source. Time bound administration of fertilizers, insecticides and pesticides can be introduced.

13. Appendix

13.1 Source Code

```
#include <WiFi.h>

#include <PubSubClient.h>

#include "DHT.h"

#include <ESP32Servo.h>

#define DHTPIN 15

#define DHTTYPE DHT22

#define LED 2

DHT dht (DHTPIN, DHTTYPE); void callback(char* subscribetopic, byte* payload,

unsigned int payloadLength);

#define ORG "dcehdm"//IBM ORGANITION ID

#define DEVICE_TYPE "Esp32"//Device type

#define DEVICE_ID "farm"//Device ID

#define TOKEN "23zPbW*+XTK!hv9qYz" //Token

Servo servo;

String data3; float h, t;

int pos= 0;

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);
pinMode(2,OUTPUT);//red
pinMode(4,OUTPUT); //blue
pinMode(5,OUTPUT); //green
delay(10); Serial.println();
wificonnect();
mqttconnect();
const int servoPin = 18;
servo.attach(servoPin, 500, 2400);
}
void loop()
{
 h = dht.readHumidity();
t= dht.readTemperature();
 Serial.print("temp:");
 Serial.println(t);
```

```
Serial.print("Humid:");
Serial.println(h);
if((t<=40)&&(t>=25))
  {
    pos=90;
    servo.write(pos);
    digitalWrite(4, HIGH);
    digitalWrite(2, LOW);digitalWrite(5, LOW);

  }
 if(t>40)
  {
    pos=180;
    servo.write(pos);
    digitalWrite(2, HIGH);
     digitalWrite(4, LOW);digitalWrite(5, LOW);
  }
 if(t<25)
  {
    pos=0;
    servo.write(pos);
     digitalWrite(5, HIGH); digitalWrite(2, LOW); digitalWrite(4, LOW);
  }
```

```
 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, 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=="lighton")
 {
Serial.println(data3);
digitalWrite(LED,HIGH);
 }
 else
 {
Serial.println(data3); digitalWrite(LED,LOW);
 }
data3="";
}
```

13.2 Github link

https://github.com/IBM-EPBL/IBM-Project-6571-1658831935

13.3 Project demo link

https://github.com/IBM-EPBL/IBM-Project-6571-1658831935/blob/master/Final%20Deliverables/final%201%20video.mp4