SMARTFARMER – IoT ENABLED SMART FARMING APLLICATION

Team ID	PNT2022TMID11108
Project Name	Project – A low cost cloud-Based IoT System for Real-Time Monitoring and Controlling for Smart Farming

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

According to **Tanya Recalde** , The digital breach between agricultural producers and IoT technologies has reduced in the last years. In the future, these technologies will allow improving productivity through the sustainable cultivation of food, as well as to take care of the environment thanks to the efficient use of water and the optimization of inputs and treatments. IoT technologies allow developing systems that support different agricultural processes. Some of these systems are remote monitoring systems, decision support tools, automated irrigation systems, frost protection systems, and fertilization systems, among others. Considering the aforementioned facts, it is necessary to provide farmers and researchers with a clear perspective of IoT applications in agriculture. In this sense, this work presents a systematic literature review of IoT-based tools and applications for agriculture. The objective of this paper is to offer an overview of the IoT applications in agriculture through topics such IoT-based software applications for agriculture available in the market, IoT-based devices used in the agriculture, as well as the benefits provided by this kind of technologies.

According to **Sobitha** , Agriculture is the most important sector of the Indian economy that provides employment to almost half the population of the country. Traditional way of farming had less concentration on humidity, water level and climatic condition which affects a farmer dreadfully. This farming will lead a loss to farmer because of labour insufficiency, water scarcity, inefficient knowledge about pest, crop selection for their land. To overcome these issues smart farming comes into existence. Automation of the farming process is called as smart farming. Internet of Things help in collecting information about various conditions like weather, moisture, temperature and fertility of soil. Based on this information farmer can irrigate their crop with required amount of water, add required amount of fertilizer, and cultivate suitable crop based on the soil nature. This paper discusses about various technologies used in smart farming, various application in smart farming and issues of IoT in agriculture.

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-withfarm-irrigationsystems

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

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

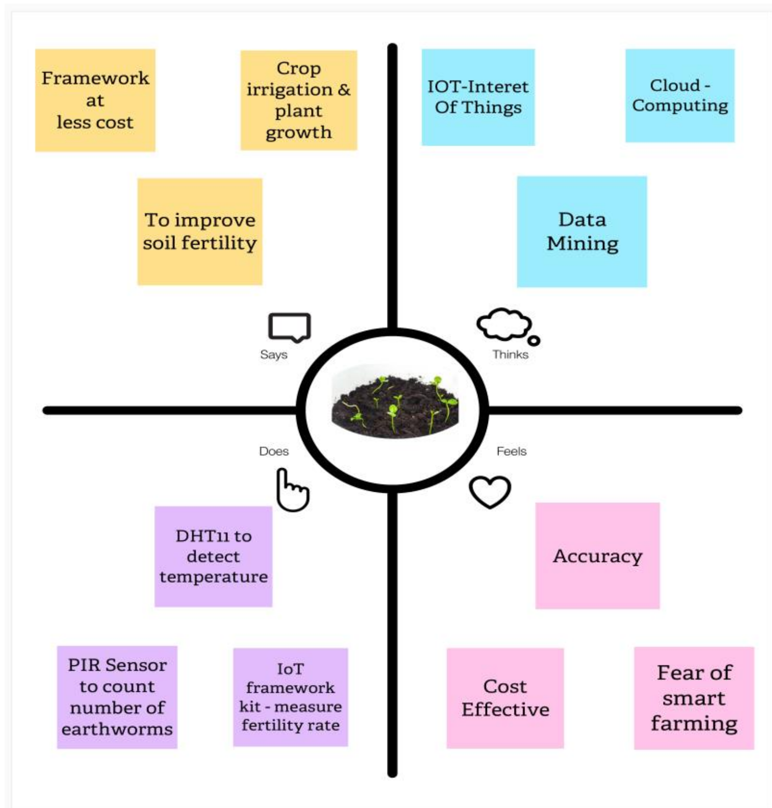
[8] Thingspeak : https:// thingspeak.com

2.3 Problem Statement Definition

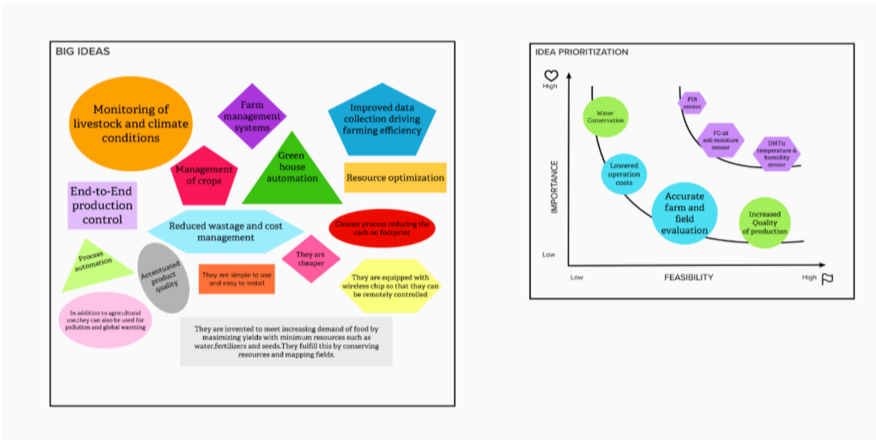
The sensor near a crop will help us to identify and alert the user whether it needs any more water or fertilizer at that moment monitored through application. Thus increasing difficulties in farming leads to the way of inventing many new technologies that will help the farmers to take some precautions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

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brainstorming



3.3 Proposed Solution

Project Design Phase-I
Proposed Solution Template

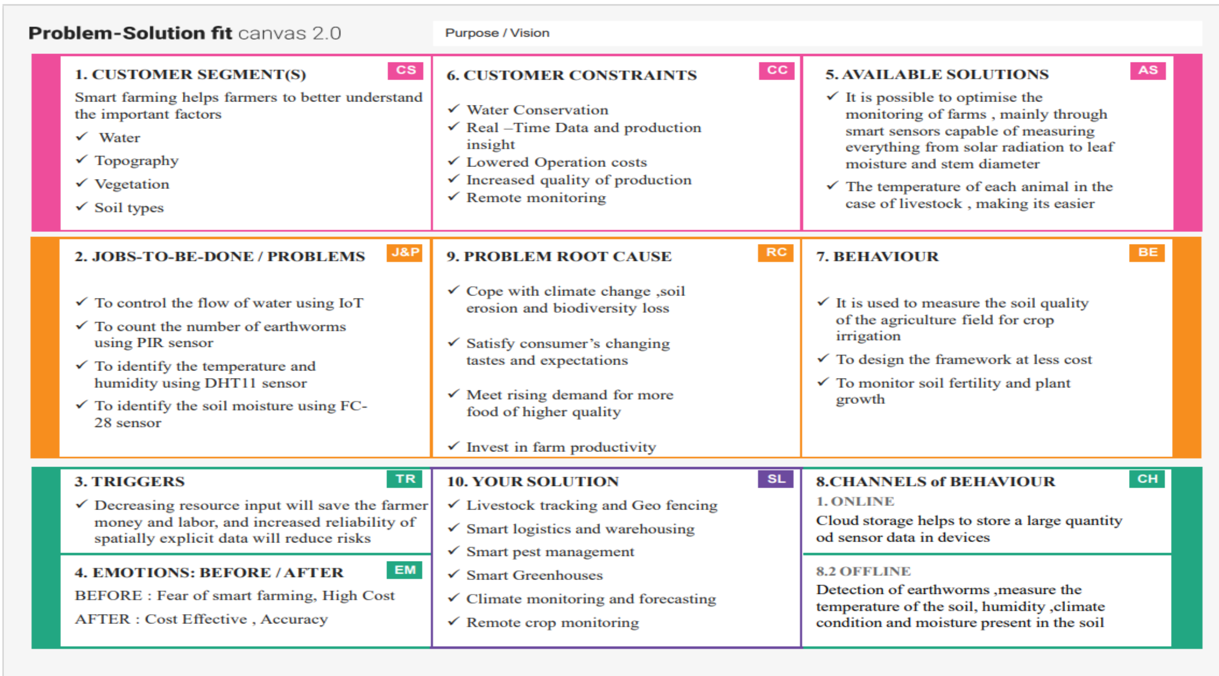
Date	1 October 2022
Team ID	PNT2022TMID11108
Project Name	Project – Smart Farming Using IoT
Maximum Marks	2 Marks

Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The whole IoT framework kit will be attached to the soil that we want to measure the fertility rate.
2.	Idea / Solution description	The framework kit consists of PIR sensor, FC28 sensor and DHT11 sensor and the data stored in cloud and shows the results in real time.
3.	Novelty / Uniqueness	It sends the notifications to the user about the soil fertility, temperature, humidity and climate conditions.
4.	Social Impact / Customer Satisfaction	It is cost effective and improve crop irrigation and also improve the growth of plant.
5.	Business Model (Revenue Model)	The IoT framework kit could be sold in the market for the purpose of testing soil fertility.
6.	Scalability of the Solution	The model could be scaled according to the fertility of the soil.

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 Requirements	A smart device will be given to the parents/guardian in order to ensure the safety of the children.
FR-2	User Registration	Manual Registration Through a Website or Gmail
FR-3	User Confirmation	Phone Confirmation Email confirmation OTP authentication
FR-4	Payments options	No payment required
FR-5	Product Delivery and installation	The installation fee will be determined with respect to the circumstances of the children and the parent.
FR-6	Product Feedback	Through a website via Gmail

4.2 Non-Functional Requirements

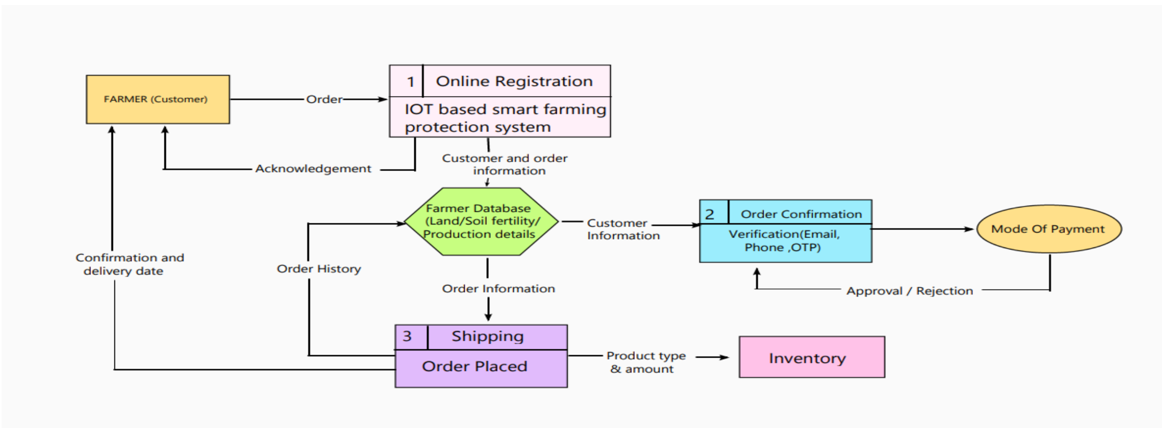
Non-functional Requirements:

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

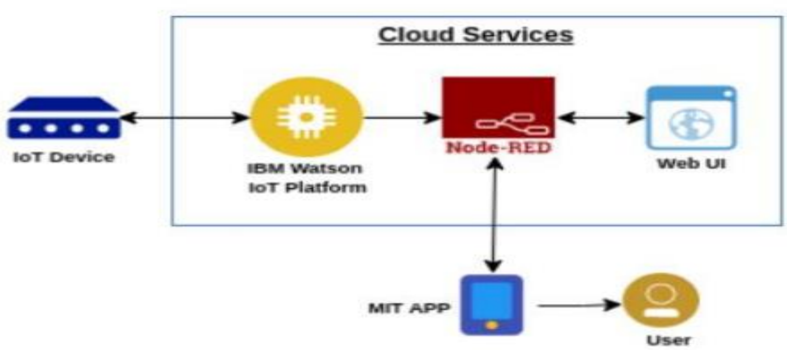
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Have clear product instructions and a self-explanatory product that is simple to use.
NFR-2	**Security**	Cloud data must be contained within the network, collapsing to be avoided, Real-time avoidance should be avoided, and the device will be constantly monitored.
NFR-3	**Reliability**	Hardware is frequently tested.
NFR-4	**Performance**	The smart device will provide a better user experience and deliver accuracy output.
NFR-5	**Availability**	All of the functions that the user demands will be provided, depending on the needs of the consumer.
NFR-6	**Scalability**	The product is based on child safety so it must ensures all types of child safety parameters are true.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Solution Architecture



5.3 User Stories

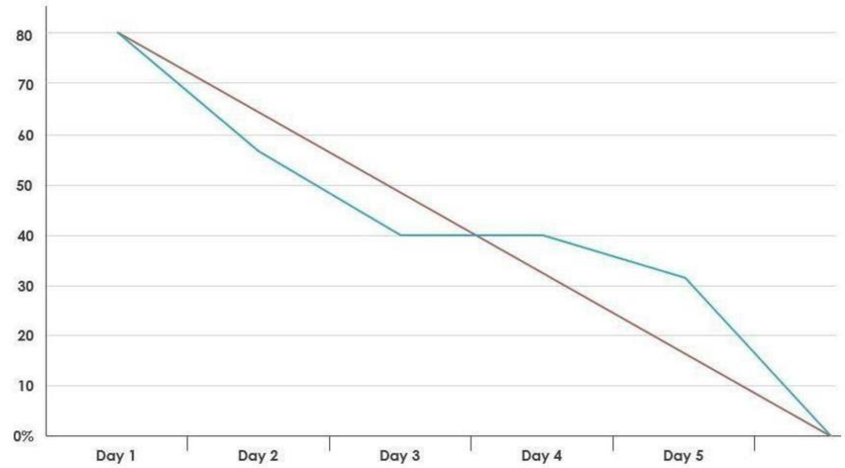
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	I can access my account/dashboard	High	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 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())) {
```

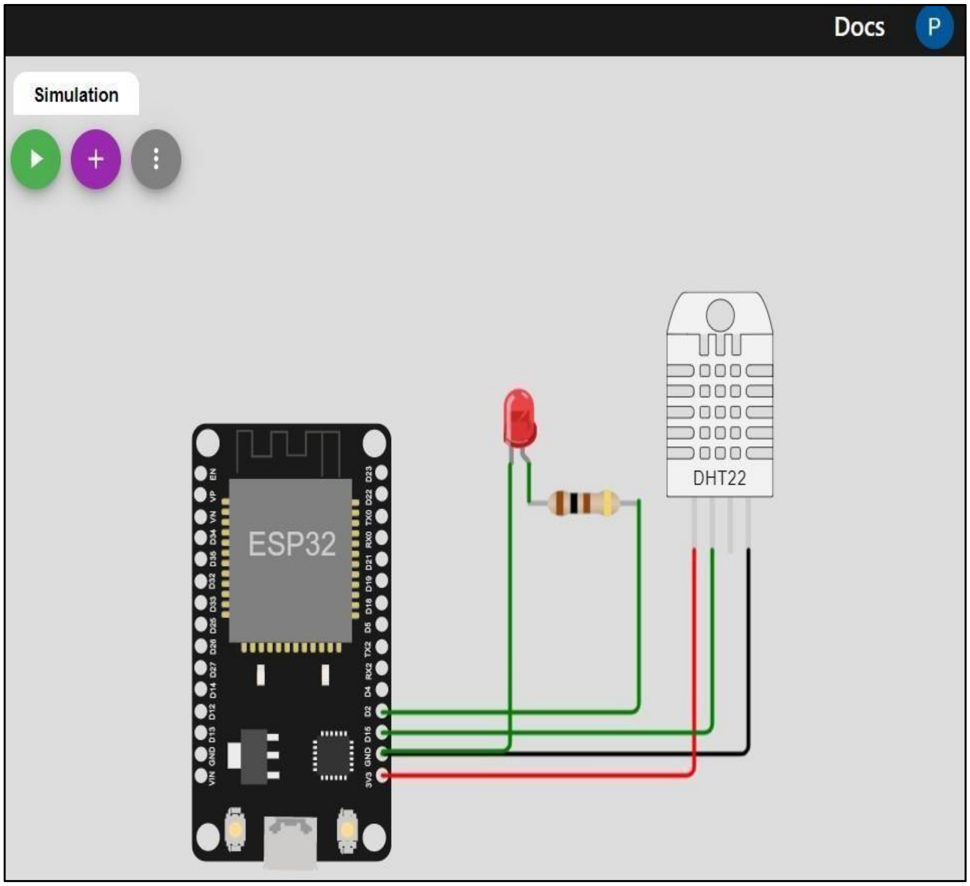
```
   Serial.println("Publish ok");
} else {

   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

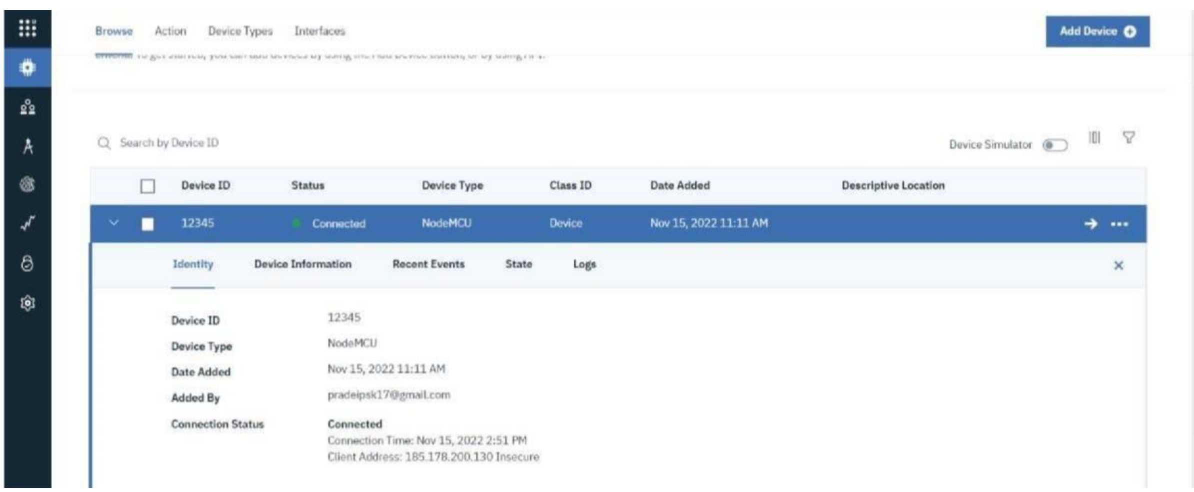
```
Connecting to ...
WiFi connected
IP address:
10.10.0.2
Reconnecting client to tu4jce.messaging.internetofthings.ibmcloud.com
iot-2/cmd/command/fmt/String
subscribe to cmd OK
```

Connecting to IBM Watson IoT platform

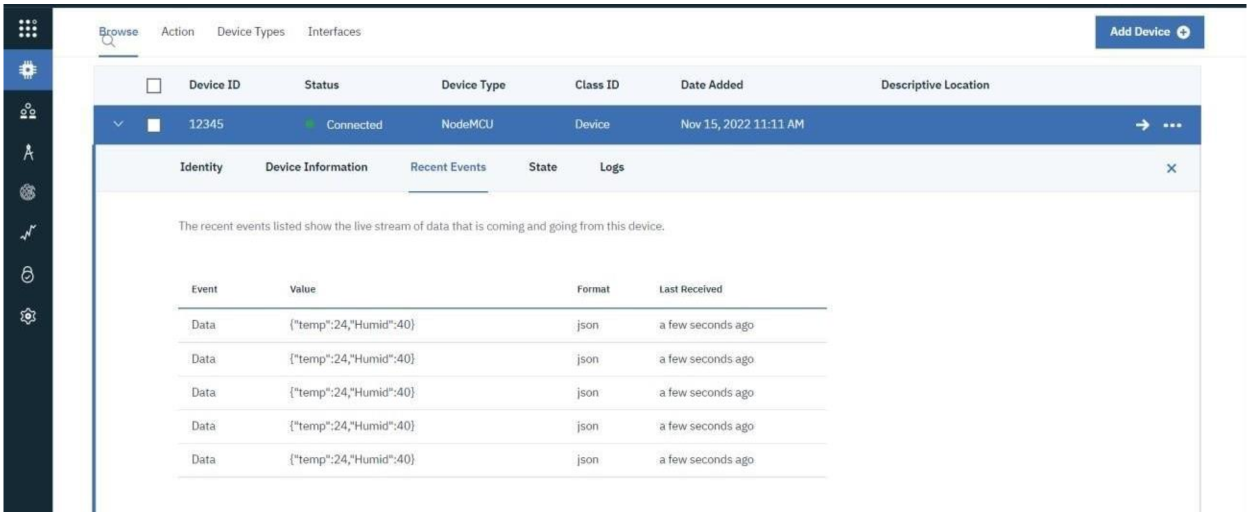
```
temp:24.00
Humid:40.00
Sending payload: {"temp":24.00,"Humid":40.00}
Publish ok
temp:24.00
Humid:40.00
Sending payload: {"temp":24.00,"Humid":40.00}
Publish ok
```

Publishing temperature and humidity values to the IBM Watson IoT platform

IBM Watson IoT platform:



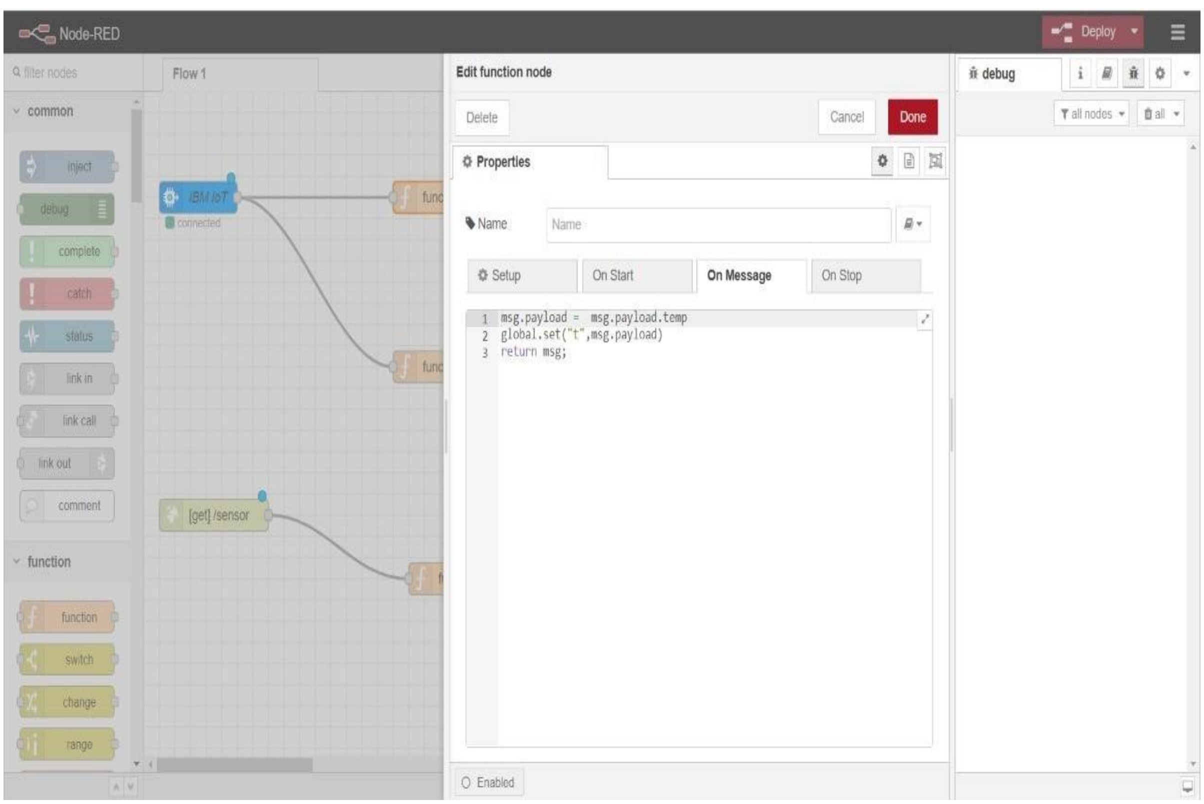
Connected Status in IBM Watson IoT platform

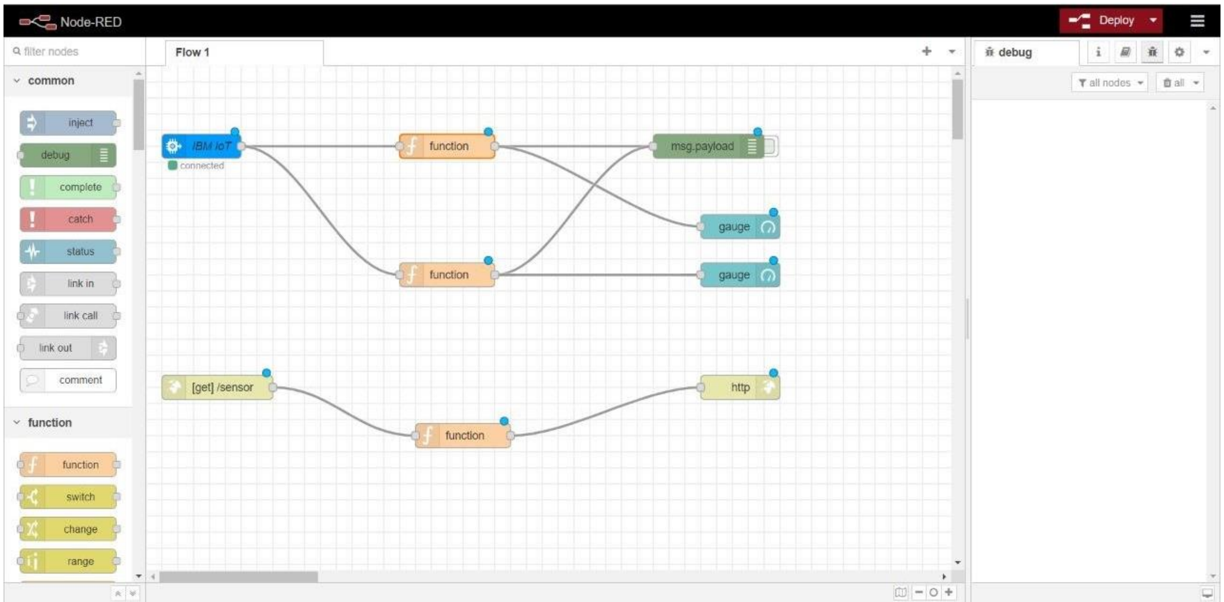


Recent Events in IBM Watson IoT platform

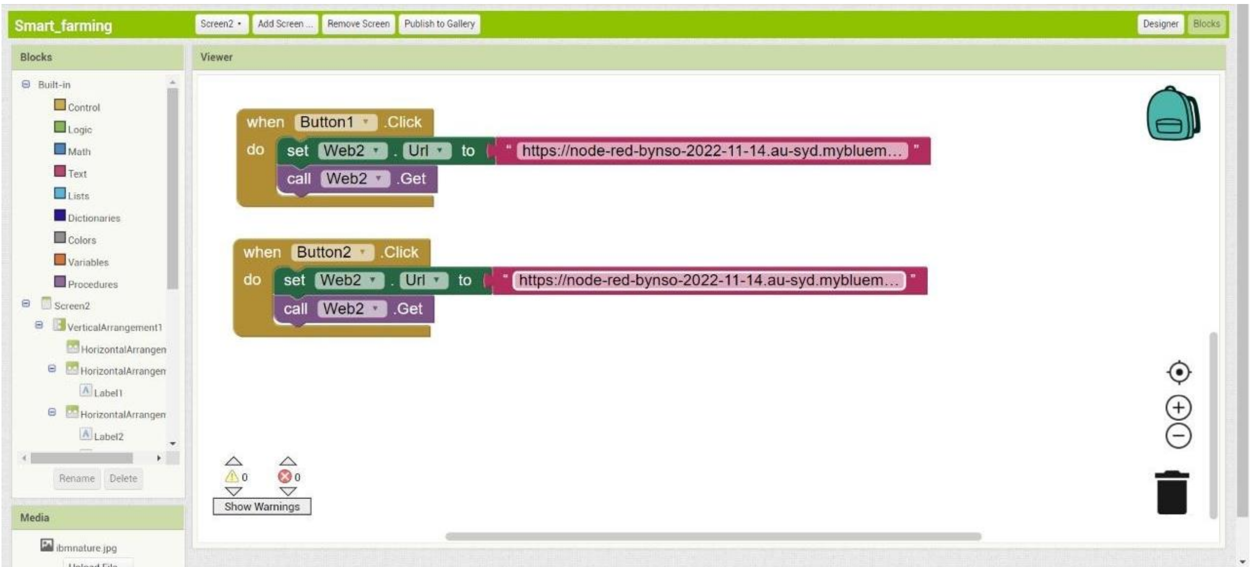
Once the sensor data like temperature and humidity gets updated in the IBM Watson IoT platform, those sensor data's will be available under recent events.

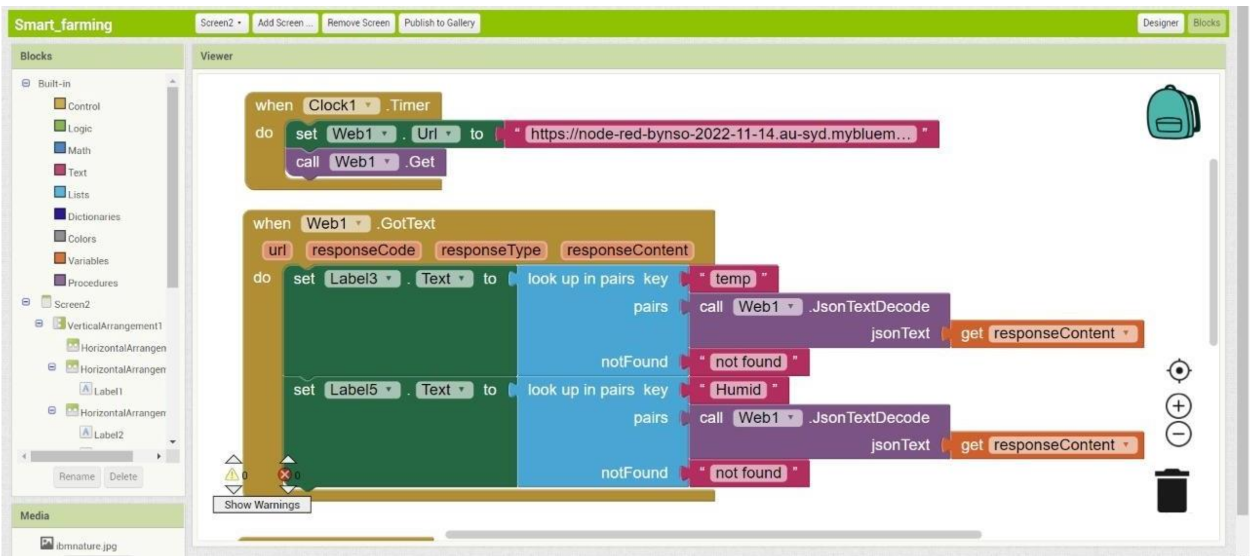
9. Results

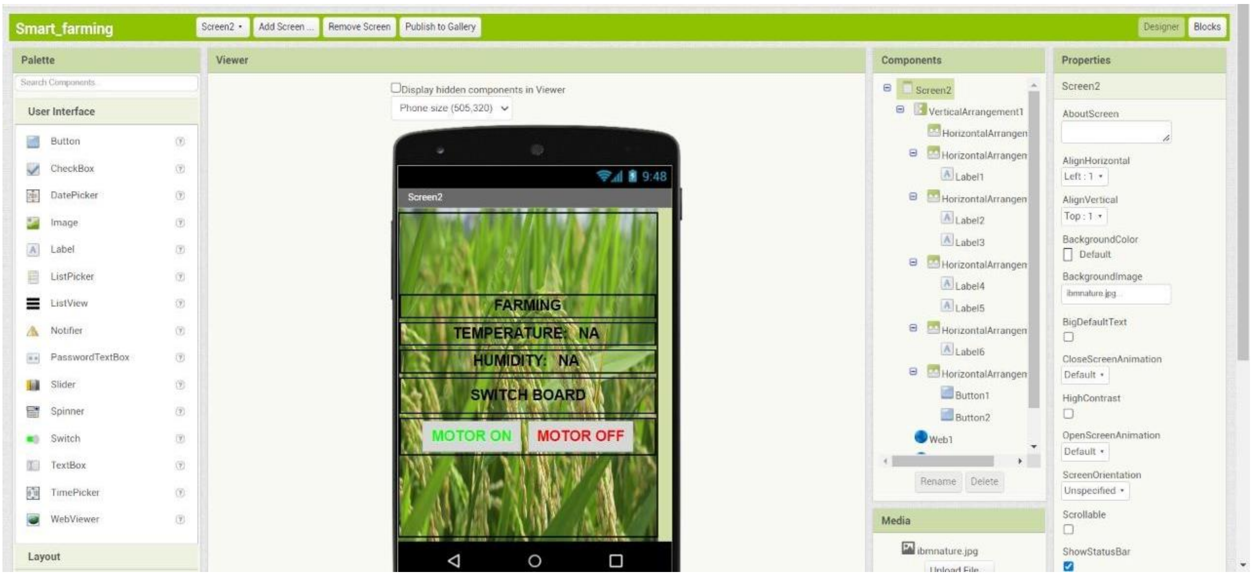


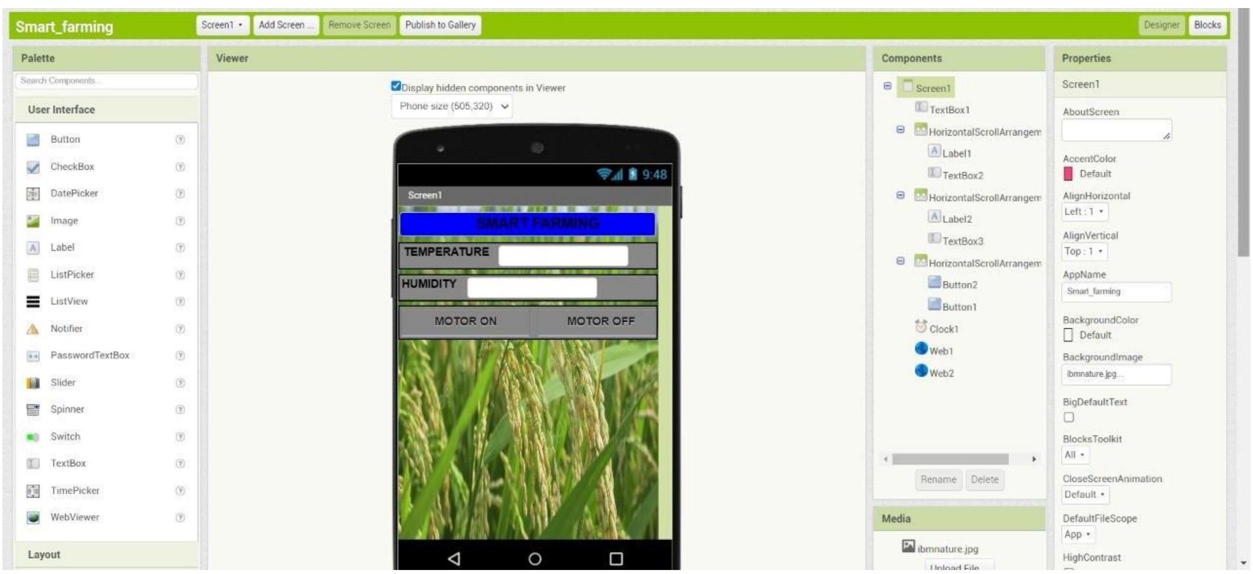


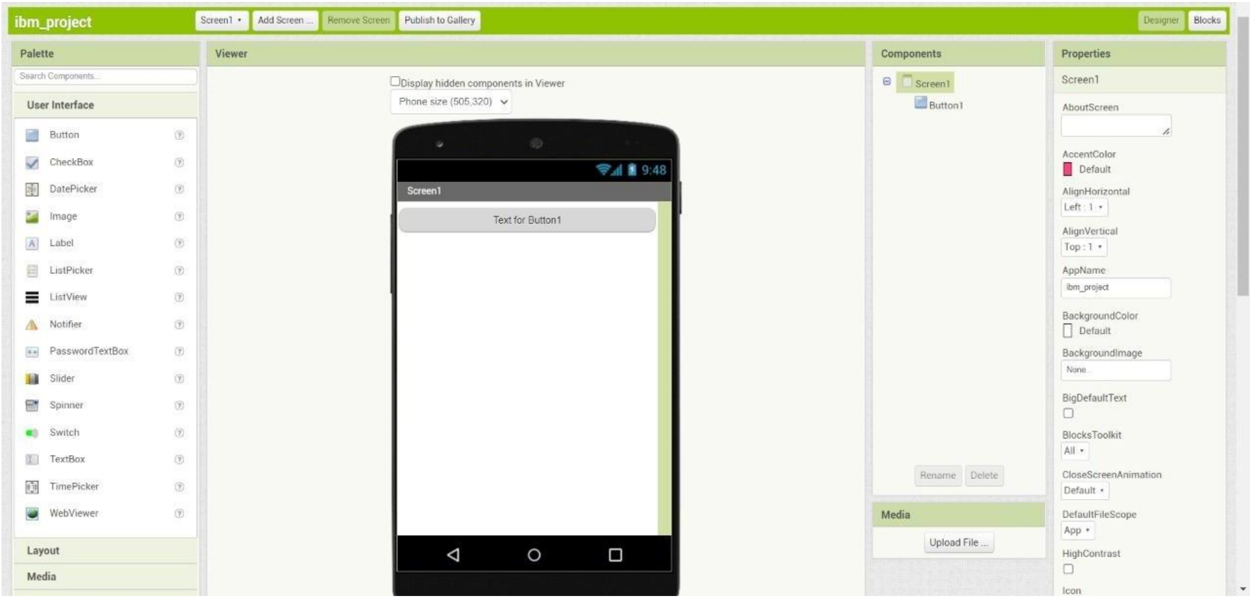
Node-RED
Deploy
filter nodes
Flow 1
debug
all nodes
all
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
IBM IoT
connected
function
function
msg.payload
gauge
gauge
[get] /sensor
function
http

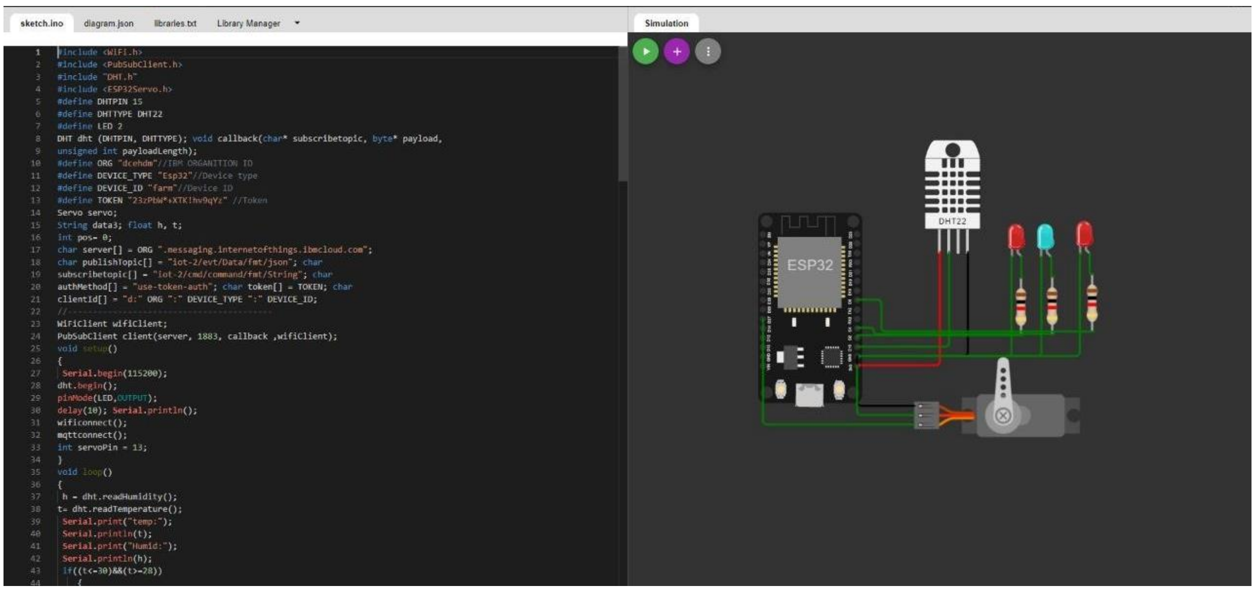

Smart_farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
VerticalArrangement1
Label1
Label2
Rename
Delete
Media
ibmnature.jpg
when Button1 .Click
do set Web2 . Url to " https://node-red-bynso-2022-11-14.au-syd.mybluem... "
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

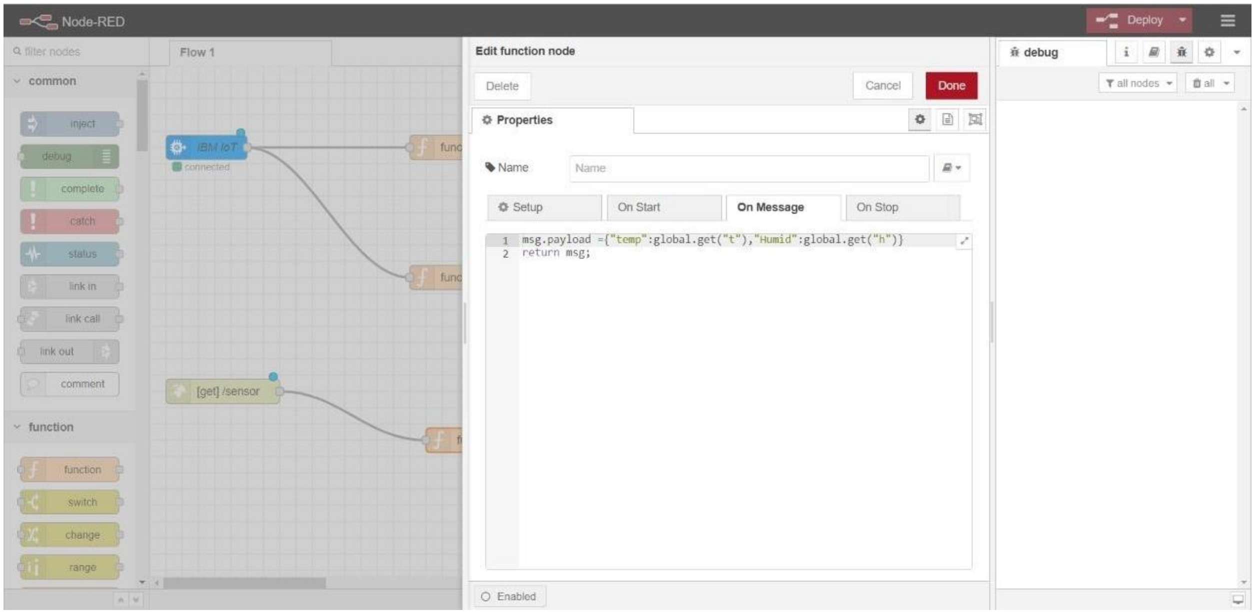

Smart_farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Built-in
Control
Logic
Math
Text
Lists
Dictionaries
Colors
Variables
Procedures
Screen2
VerticalArrangement1
Label1
Label2
Rename
Delete
Media
ibmnature.jpg
when Clock1 .Timer
do set Web1 . Url to " https://node-red-bynso-2022-11-14.au-syd.mybluem... "
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

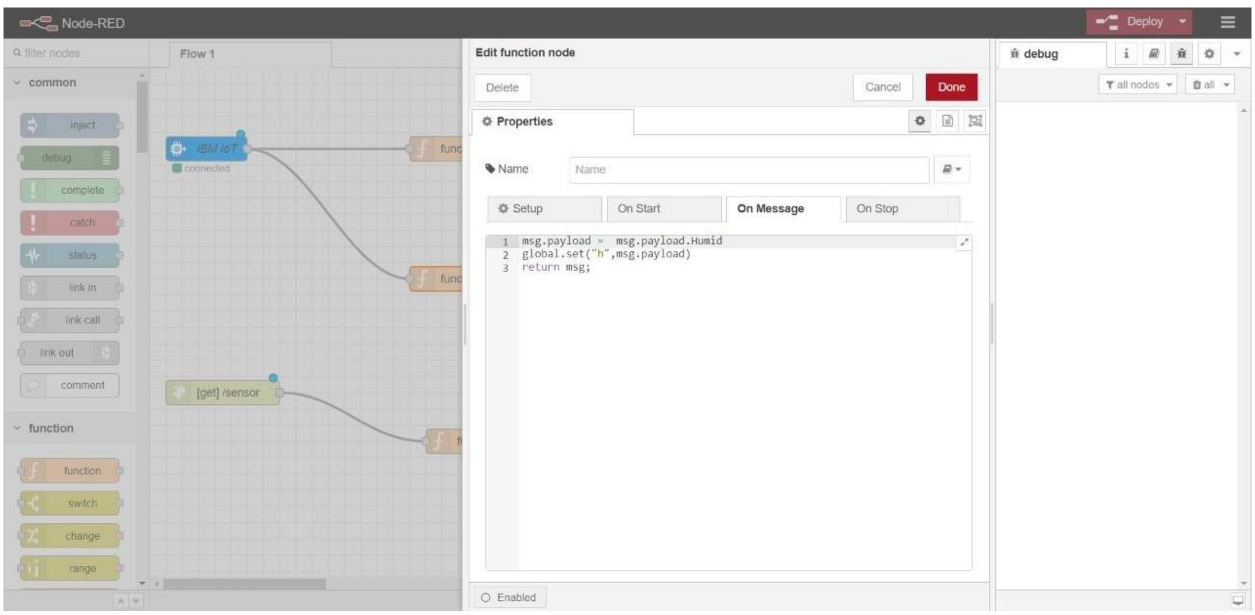

Smart_farming
Screen2
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen2
FARMING
TEMPERATURE: NA
HUMIDITY: NA
SWITCH BOARD
MOTOR ON
MOTOR OFF
Components
Screen2
VerticalArrangement1
HorizontalArrangen
HorizontalArrangen
Label1
HorizontalArrangen
Label2
Label3
HorizontalArrangen
Label4
Label5
HorizontalArrangen
Label6
HorizontalArrangen
Button1
Button2
Web1
Rename
Delete
Media
ibmnature.jpg
Properties
Screen2
AboutScreen
AlignHorizontal
Left : 1
AlignVertical
Top : 1
BackgroundColor
Default
BackgroundImage
ibmnature.jpg
BigDefaultText
CloseScreenAnimation
Default
HighContrast
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowStatusBar

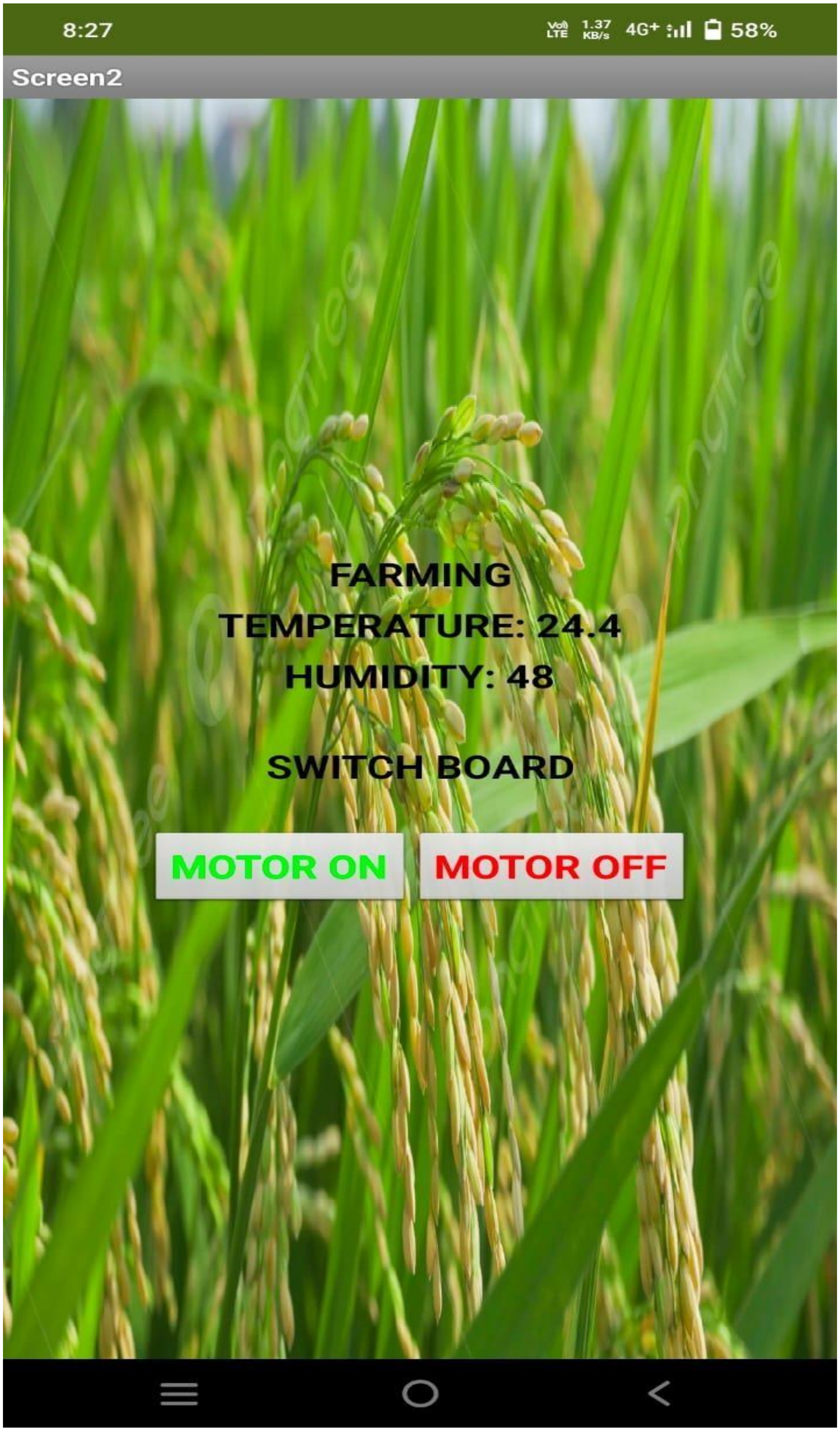

Smart_farming
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
SMART FARMING
TEMPERATURE
HUMIDITY
MOTOR ON
MOTOR OFF
Components
Screen1
TextBox1
HorizontalScrollArrangem
Label1
TextBox2
HorizontalScrollArrangem
Label2
TextBox3
HorizontalScrollArrangem
Button2
Button1
Clock1
Web1
Web2
Rename
Delete
Media
ibmnature.jpg
Properties
Screen1
AboutScreen
AccentColor
Default
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
Smart_farming
BackgroundColor
Default
BackgroundImage
ibmnature.jpg...
BigDefaultText
BlocksToolkit
All
CloseScreenAnimation
Default
DefaultFileScope
App
HighContrast


ibm_project
Screen1
Add Screen ...
Remove Screen
Publish to Gallery
Designer
Blocks
Palette
Search Components...
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Viewer
Display hidden components in Viewer
Phone size (505,320)
9:48
Screen1
Text for Button1
Components
Screen1
Button1
Rename
Delete
Media
Upload File ...
Properties
Screen1
AboutScreen
AccentColor
Default
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
ibm_project
BackgroundColor
Default
BackgroundImage
None...
BigDefaultText
BlocksToolkit
All
CloseScreenAnimation
Default
DefaultFileScope
App
HighContrast
Icon


sketch.ino
diagram.json
libraries.txt
Library Manager
#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);
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))
{
Simulation
ESP32
DHT22


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
msg.payload ={"temp":global.get("t"),"Humid":global.get("h")}
return msg;
Enabled
debug
all nodes
all


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
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-6390-1658827688