-                                   -

PROJECT REPORT

PROJECT NAME: SMARTFARMER-IOT ENABLED SMART FARMING APPLICATION

TEAM ID: PNT2022TMID49427

TEAM SIZE:4

TEAM LEADER: PRIYADHARSHINI N

TEAM MEMBERS:

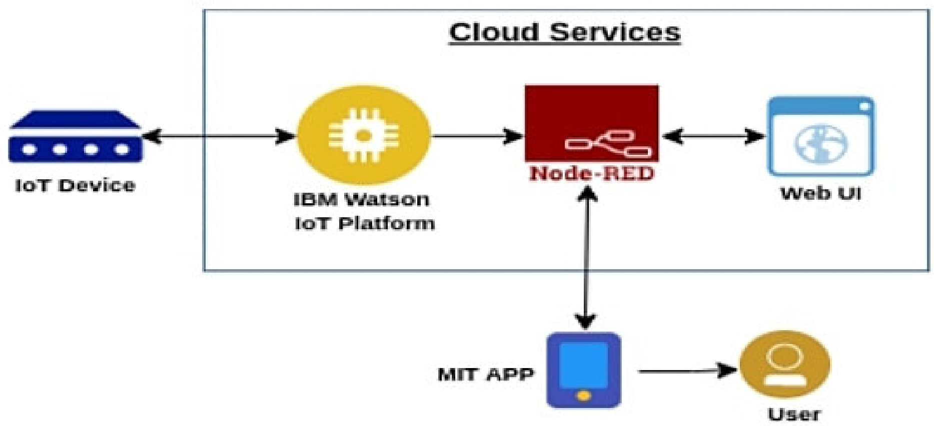
1. AMBIKA T
2. NIHARIHA P
3. SINDHUPRIYA G

1.INTRODUCTION
1.1 Project Overview
1.2 Purpose
2.LITERATURE SURVEY
2.1 Existing problem
2.2 References
2.3 Problem Statement Definition
3.IDEATION & PROPOSED SOLUTION
3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution fit
4.REQUIREMENT ANALYSIS
4.1Functional requirement
5.PROJECT DESIGN
5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
6.PROJECT PLANNING & SCHEDULING
6.1 Sprint Planning, Schedule & Estimation
7.CODING & SOLUTIONING (Explain the features added in the project along with code)
7.1 Feature
8.TESTING
8.1 Test Cases
8.2 User Acceptance Testing
9.RESULTS
9.1 Performance Metrics
10.ADVANTAGES & DISADVANTAGES
11.CONCLUSION
12.FUTURE SCOPE
13.APPENDIX
Source Code
GitHub & Project Demo Link

1.INTRODUCTION

1.1 PROJECT OVERVIEW

IOT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, 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.



1.2 PURPOSE

The smart agriculture model aim to **monitoring the fields and control the irrigation process**. It is low cost and efficient system. It includes ESP32,cloud, nodered, MIT app inverter, sensors like DHT22,soil moisture sensor.

2.LITERATURE SURVEY

2.1 Existing problem

The challenges of a smart agriculture system include the integration of these sensors and tying the sensor data to the analytics driving automation and response activities. When integrated, the use of data analytics can reduce the overall cost of agriculture and contribute to higher production from the same amount of area through precise control of water, fertilizer and light. Smart methods allow for farming on smaller and more distributed lands through remote monitoring, whether indoor or outdoor.

To successfully deploy a smart agriculture system, consider setting up a communications network that can integrate a limited number of sensors across a large area of farmland. This will require third-party network provisioning or setting up a private network consisting of access points and uplinks to a private backhaul network, which channels all the data traffic to centralized monitoring software or an analytics head-end system

a. It is not a secure system.
b. There is no motion detection for protection of agriculture field.
c. Automation is not available.

2.2 REFERENCES:

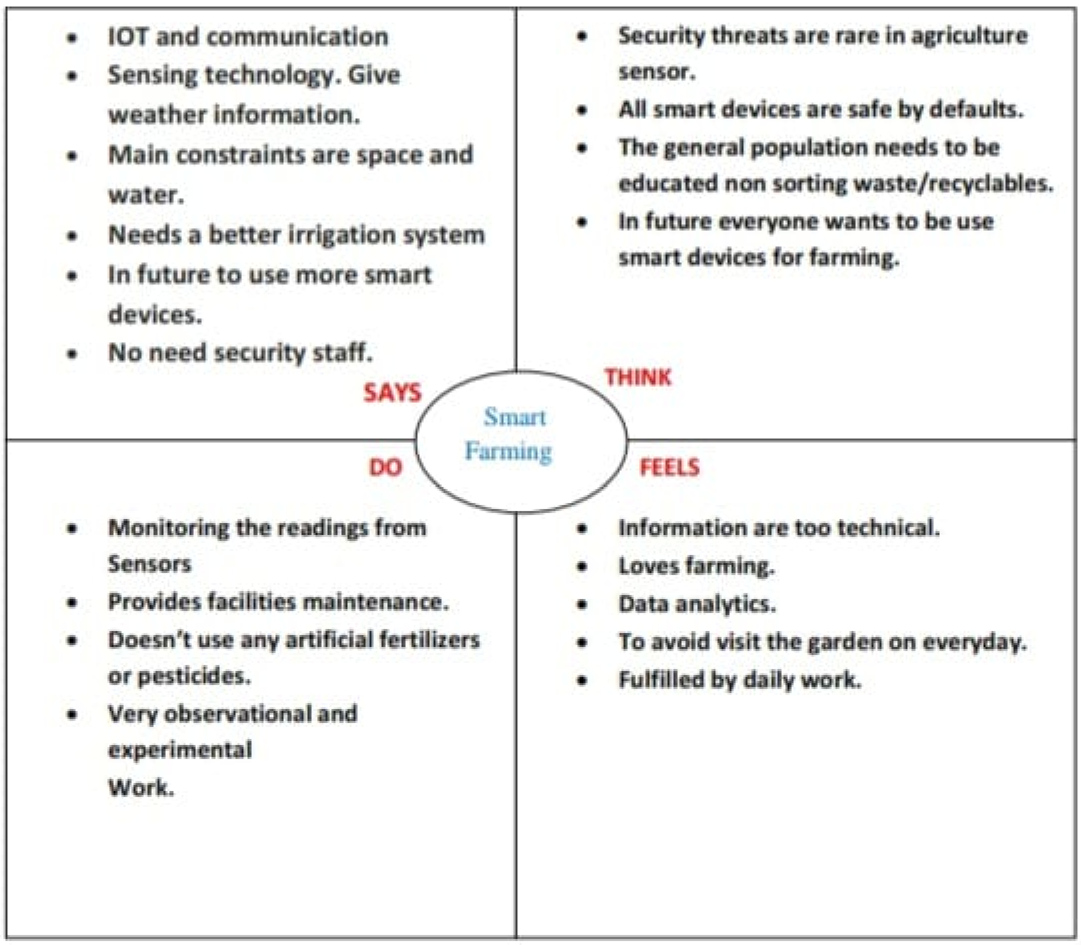
- IoT Enabled Smart Farming and Irrigation System: Authors: M. Rohith, R Sainivedhana, Dr. N. Sabiyath Fatima Published: IEEE 2021
- A Systematic Review of IoT Solutions for Smart Farming: Authors: Emerson Navarro, Nuno Costa, and António Pereira Published: MDPI 2020
- A Multi-collective, IoT-enabled, Adaptive Smart Farming Architecture: Authors: G. Kakamoukas, P. Sariciannidis, G. Livanos, M. Zervakis, D. Ramnalis, V. Polychrnos, T. Karamitsou, A. Folinas, N. Tsitsiokas Published: IEEE 2019
- Internet of Things and LoRaWAN – Enabled Future Smart Farming Authors: Bruno Citoni, Francesco Fioranelli, Muhammad A. Imran, Qammer H. Abbasi Published: IEEE 2019
- A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming Authors: Muhammad Shoaib Farooq, Shamyla Riaz, Adnan Abid, Kamran Abid, Muhammad Azhar Naeem Published: IEEE 2019

2.3 PROBLEM STATEMENT DEFINITION:

The soil moisture sensor measures wetness content in the soil. The ESP32 microcontroller used to receive input from a various sensors and it can be controlled automatically. When soil moisture sensor goes low the water pump will be on and it exceeds defined levels of the water motor will turn off automatically. DHT22 sensor detects the temperature and humidity level in the agricultural land. This device his very helpful to the farmer to monitor and control environmental parameters at their field. The farmers did not go to their field, they can remotely monitor and control using cloud.

3.IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



3.2 IDEATION AND BRAINSTORMING:

Introduction on Internet of Things (IoT), application of IoT in agricultural field to improve the yield and quality by reducing the cost is provided. The sensors which are used in the architecture are discussed briefly and the process of transmission of data from the agriculture field to the central system is explained. The proposed system advantages are included. In addition, open research issues, challenges, and future of IoT in agricultural field are highlighted. The concept is basically developed on an idea, where there are numerous things or objects - such as ESP32, sensors, cloud, nodered etc., that are connected with the Internet. Each of the objects has a different address and is able to interact with other items. The things or objects co-operate with each other to reach a common goal.

We are going to construct a smart agricultural monitoring system which can collect crucial agricultural data and send it to an IoT platform called Thingspeak in real time where the data can be logged and analyzed. The logged data on Thingspeak is in graphical format, a botanist or a reasonably knowledged farmer can analyze the data (from anywhere in the world) to make sensible changes in the supplied resources (to crops) to obtain high quality yield.

Smart agriculture monitoring system or simply smart farming is an emerging technology concept where data from several agricultural fields ranging from small to large scale and its surrounding are collected using smart electronic sensors. The collected data are analyzed by experts and local farmers to draw short term and long-term conclusion on weather pattern, soil fertility, current quality of crops, amount of water that will be required for next week to a month etc.

We can take smart farming a step further by automating several parts of farming, for example smart irrigation and water management. We can apply predictive algorithms on microcontrollers to calculate the amount of water that will be required today for a particular agriculture field. Say, if there was rain yesterday and the quantity of water required today is going to be less. Similarly, if humidity was high the evaporation of water at upper ground level is going to be less, so water required will be less than normal, thus reducing water usage.

3.3 PROPOSED SOLUTION:

The IoT technology has realized the smart wearable's, connected devices, automated machines, and driverless cars. However, in agriculture, the IoT has brought the greatest impact.

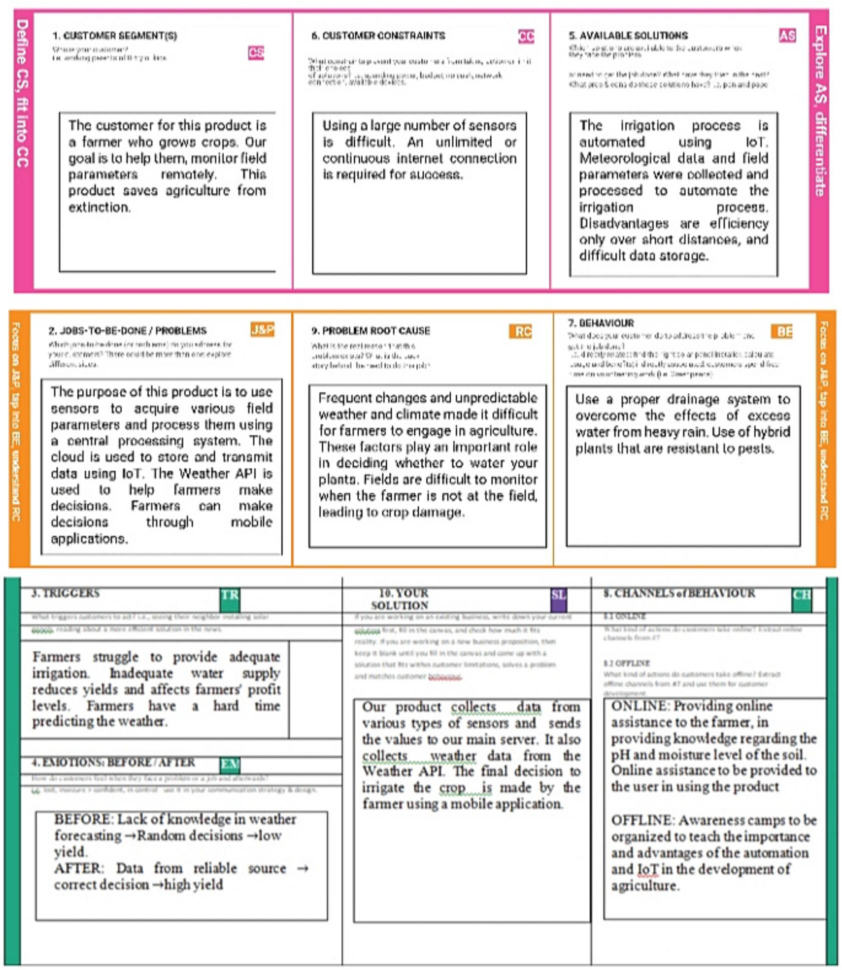
Recent statistics reveal that the global population is about to reach 9.6 billion by 2050. And to feed this massive population, the agriculture industry is bounded to adopt the Internet of Things. Amongst the challenges like extreme weather conditions, climatic changes, environmental impact, IoT is eradicating these challenges and helping us to meet the demand for more food.

Throughout the world, mechanical innovations such as tractors and harvesters took place and brought into the agriculture operations in the late 20th century. And the agriculture Industry relies heavily on innovative ideas because of the steadily growing demand for food.

The Industrial IoT has been a driving force behind increased agricultural production at a lower cost. In the next several years, the use of smart solutions powered by IoT will increase in the agriculture operations. In fact, few of the recent report tells that the IoT device installation will see a compound annual growth rate of 20% in the agriculture industry. And the no. of connected devices (agricultural) will grow from 13 million in 2014 to 225 million by 2024.

Due to lack of constant and reliable communication network infrastructure, an IoT solutions provider as well as the business owners had faced implementation challenges in remote or less developed regions. But, many network providers are making it possible by introducing satellite connectivity and expending cellular networks.

3.4 PROBLEM SOLUTION FIT:



4.REQUIREMENT ANALYSIS

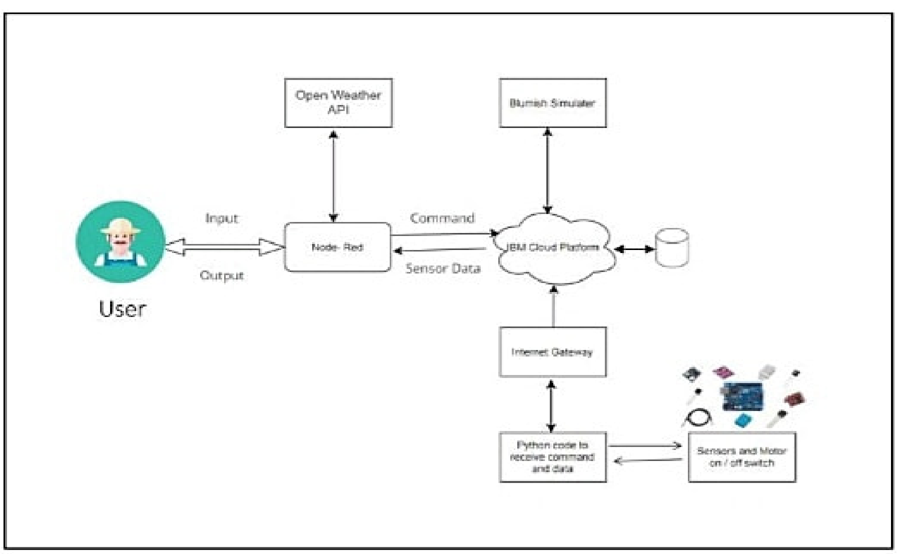
4.1 FUNCTIONAL REQUIREMENT:

Following are the functional requirements of the proposed solution.

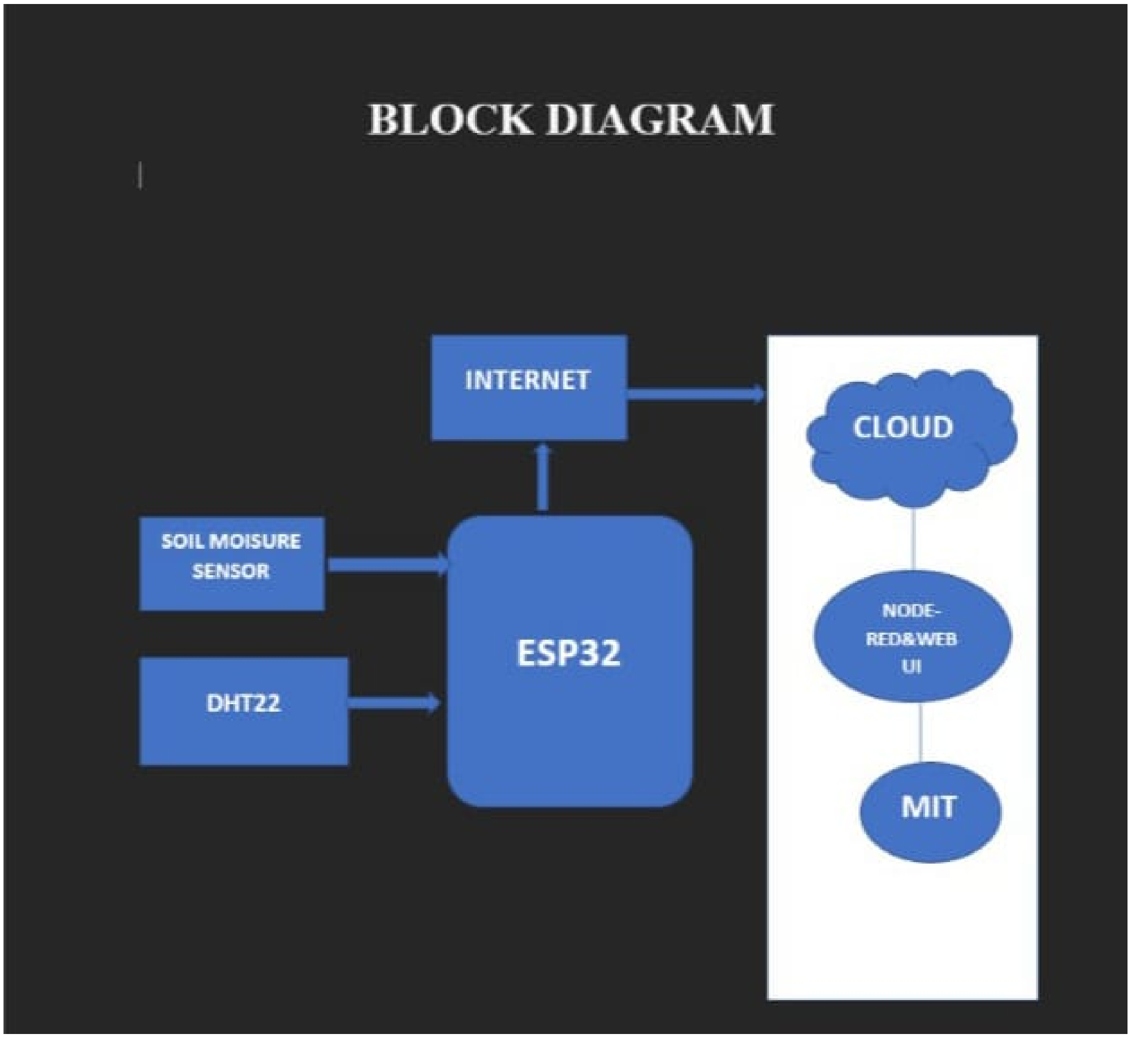
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Sensor Function for framing System	Measure the Temperature and Humidity Measure the Soil Monitoring Check the crop diseases
FR-4	Manage Modules	Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details Humidity details
FR-6	Data Management	Manage the data of weather conditions Manage the data of crop conditions Manage the data of live stock conditions

5.PRODUCT DESIGN

5.1DATA FLOW DIAGRAM:



5.2SOLUTION AND TECHNICAL ARCHITECTURE



6.PROJECT PLANNING AND SCHEDULING

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation creation	USN-1	Connect Sensors and Arduino with python code	4	High	N.Priyadharshini T.Ambika P.Nihariha G.Sindhupriya
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	4	High	N.Priyadharshini T.Ambika P.Nihariha G.Sindhupriya
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using MIT App Inventor	4	High	N.Priyadharshini T.Ambika P.Nihariha G.Sindhupriya
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	4	High	N.Priyadharshini T.Ambika P.Nihariha G.Sindhupriya
Sprint-4	Web UI	USN-4	To make the user to interact with software	4	High	N.Priyadharshini T.Ambika P.Nihariha G.Sindhupriya

7.CODING AND SOLUTIONING

7.1 FEATURE:

```
#include <WiFi.h>//library for wifi

#include <PubSubClient.h>//library for MQtt

#include "DHT.h"// Library for dht11

#define DHTPIN 15      // what pin we're connected to

#define DHTTYPE DHT22    // define type of sensor DHT 11



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

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

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

#define ORG "gtes6m"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "project"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "M3hA0E*8k)?DrC4@eY"     //Token
String data3;
float h, t;


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


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


void setup()// configureing the ESP32
{
  Serial.begin(115200);
```

```
  dht.begin();

  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  int s=random(100);
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);
  Serial.print("Moisture:");
  Serial.println(s);

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



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

void PublishData(float temp, float humid,int Moisture) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
```

```
  */
  String payload = "{\"temp\":";
  payload += temp;
  payload += "," "\"Humid\":";
  payload += humid;
   payload += "," "\"Moisture\":";
  payload += Moisture;
  payload += "}";


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


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

}


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

     initManagedDevice();
     Serial.println();
```

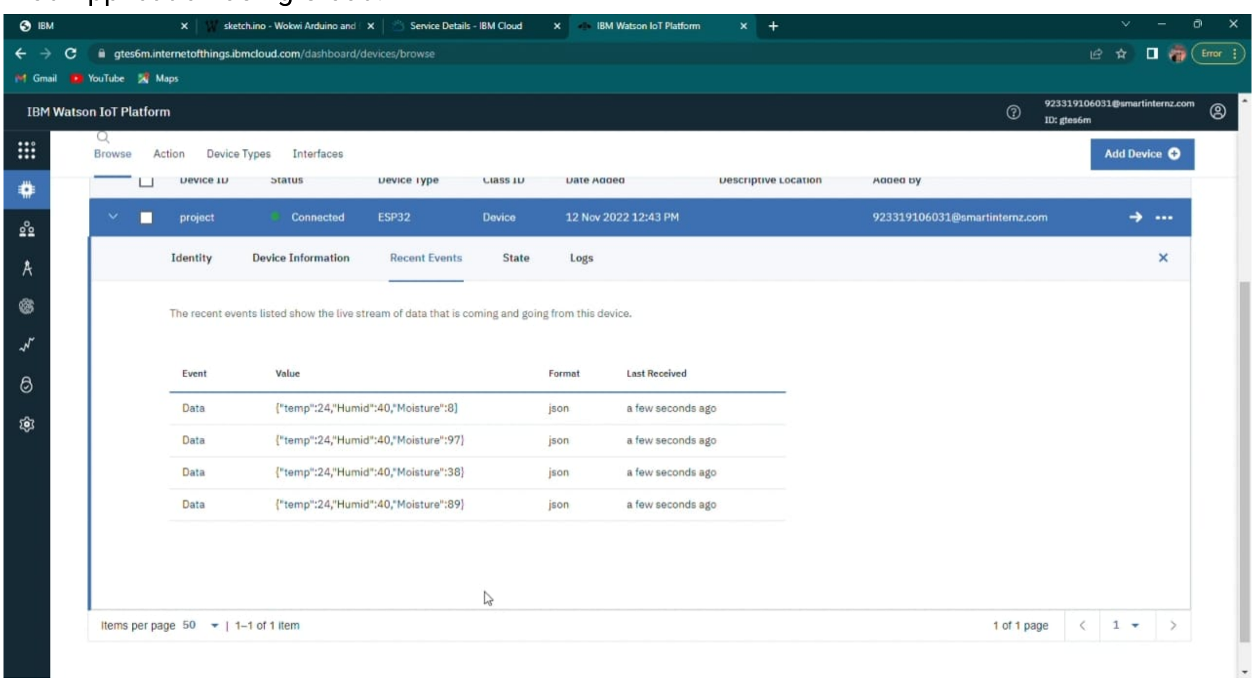
```
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");

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

  }
  else
  {
Serial.println(data3);

  }
data3="";
}
```

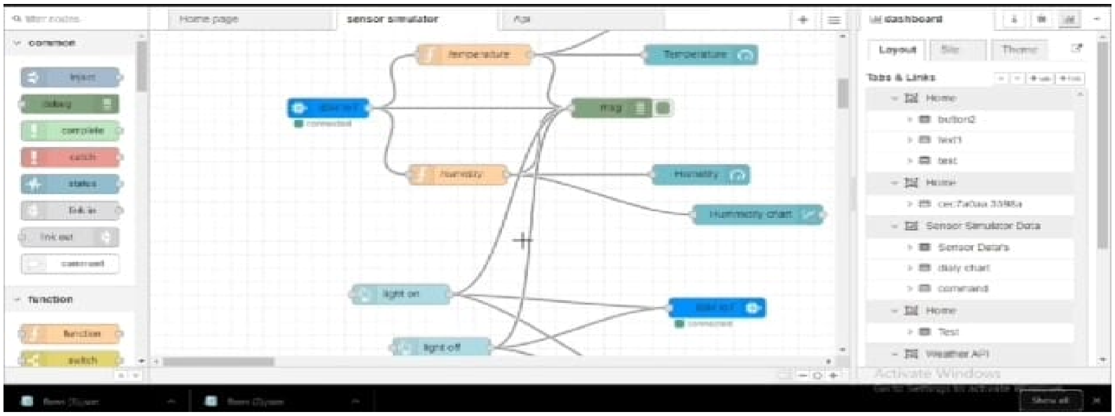
8.TESTING

8.1 TEST CASE

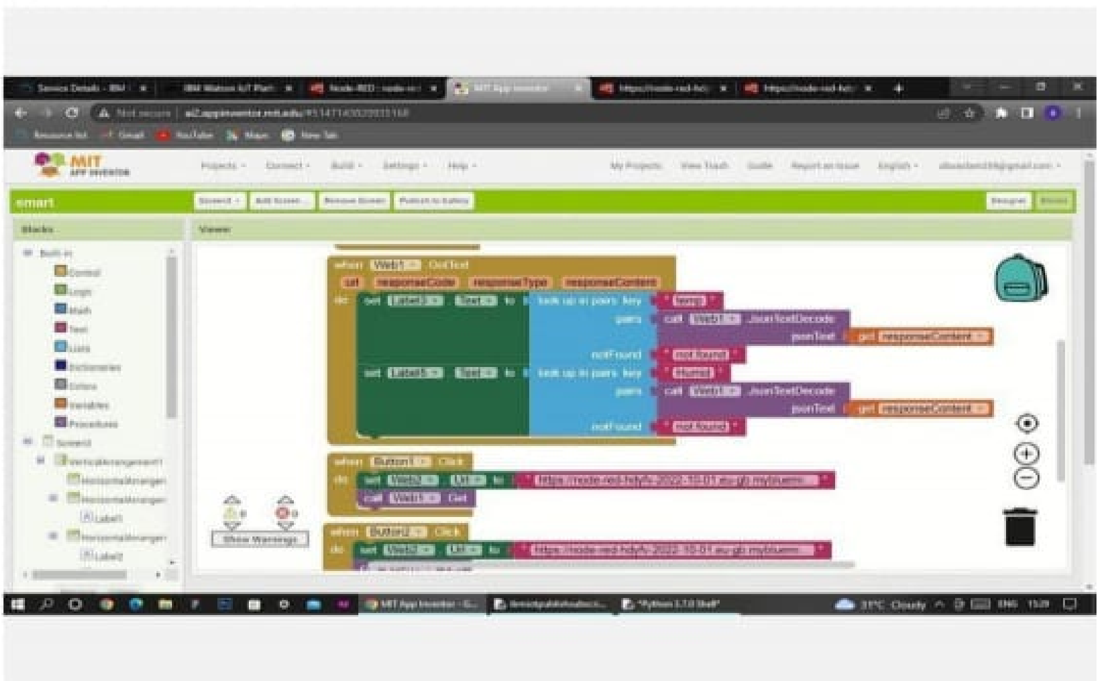
Web Application using Cloud:



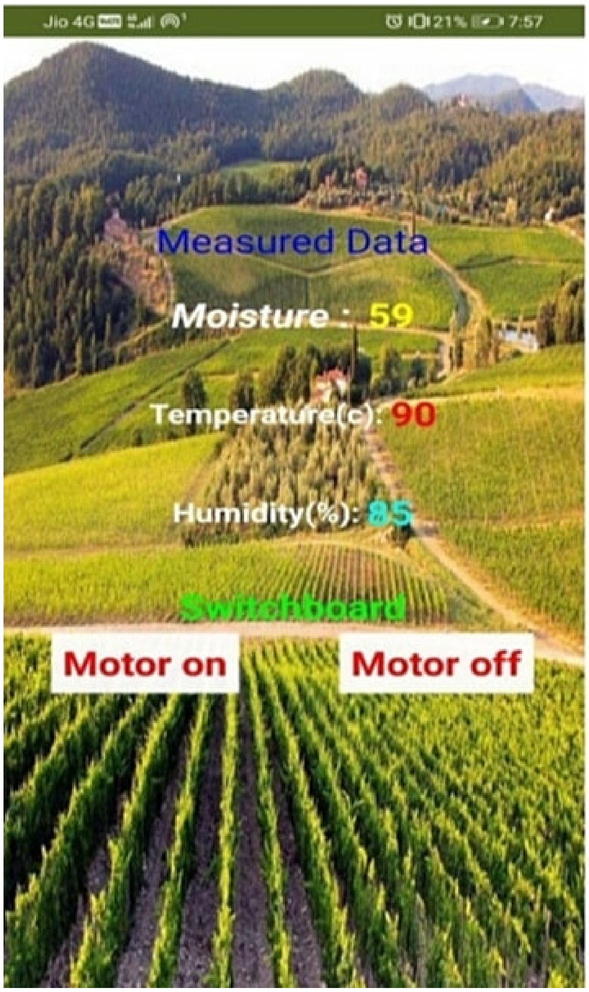
Web application using node-red:



Web application using MIT App inverter:

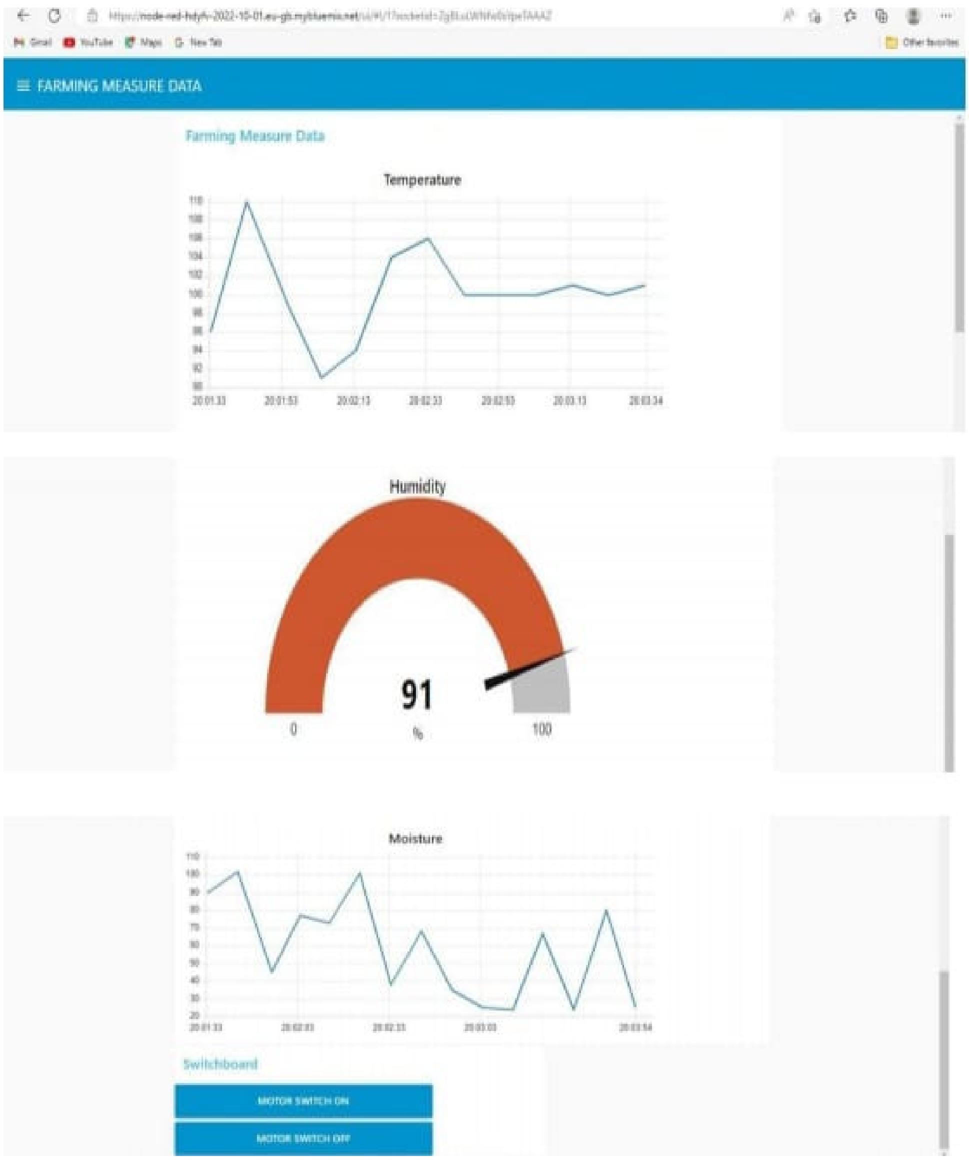


8.2 USER ACCEPTANCE TESTING:



9.RESULTS

9.1 PERFORMANCE METRICS:



10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.

- For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.

- Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.

- Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.

DISADVANTAGES:

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

11.CONCLUSION

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted dependent on the local environmental variables. This technology also detects animal invasions, which are a major cause of crop loss. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source. It helps in deciding the farmer to whether to do irrigation or not to do. Continuous internet connectivity is required for continuous monitoring of data from sensors. This also can be overcome by using GSM unit as an alternative of mobile app. By GSM, SMS can be sent to farmers phone.

12.FUTURE SCOPE

In the current project we have implemented the project that can protect and maintain the the crop. In this project the farmer monitor and control the field remotely. In future we can add or update few more things to this project

- We can create few more models of the same project ,so that the farmer can have information of a entire.
- We can update the this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. It reduces the power line cost. It will be a one time investment. We can add solar fencing technology to this project.
- We can use GSM technology to this project so that the farmers can get the information directly to his home through SMS. This helps the farmer to get information if there is a internet issues.

13.APPENDIX

SOURCE CODE:

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



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

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

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

#define ORG "gtes6m"//IBM ORGANITION ID
#define DEVICE_TYPE "ESP32"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "project"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "M3hA0E*8k)?DrC4@eY"      //Token
String data3;
float h, t;
```

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


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


void setup()// configureing the ESP32
{
  Serial.begin(115200);
  dht.begin();

  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  h = dht.readHumidity();
  t = dht.readTemperature();
  int s=random(100);
```

```
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);
  Serial.print("Moisture:");
  Serial.println(s);

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



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

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


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

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

}


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

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

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

  }
  else
  {
Serial.println(data3);

  }
```

```
data3="";
}
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-9114-1658981159

PROJECT DEMO LINK:

https://drive.google.com/file/d/1YQ4pqDs8WVeKwWUrpJDoXzkezJscKQlI/view?usp=drivesdk