PROJECT REPORT

IBM – NALAIYA THIRAN

PROJECT NAME : SMART FARMER – IOT ENABLED SMART FARMING APPLICATION

TEAM ID : PNT2022TMID38157

TEAM :

SRI RAJESWARI . E – TEAM LEAD

MATHIBALAN . M

RUHI ANGEL . A

THIRUPONPUGAZH . P

1. INTRODUCTION

Project Overview

Agriculture plays a important role in country's economy and provides a large scale employment to the people. However, agriculture is highly dependent upon weather and climate. For example, changes in temperature, soil moisture, carbon dioxide may result in low yield of crops. It is Significant to monitor environmental parameters in order to manage crop growth and increase the agricultural production yield. The sensed information is not only important for decision making but also for evaluating impacts of agricultural practices on environment. Nowadays, it is more necessary than ever to increase the crop yields food grain production. Cloud connected, wireless system aid in this crop yield maximization, which automates day- to-day agricultural tasks and real time monitoring for smart decision-making.

Purpose

- Need for technology to monitor important parameters like soil moisture,temperature, Humidity etc. to improve the cultivation process.
- Need for technology to monitor weather of particular area with reliable source to save the crops at the time of natural calamities like flood, cyclone etc.
- Development of certain techniques to reduce the workforce, energy and time forcultivation.
- Development of a feasible method to control the electrical equipment in the farm fromany part of the world.

2. LITERATURE SURVEY

EXISTING PROBLEM

1. Controlling the device from longer distance from web application.
2. Getting the weather data from weather station.
3. Transfer of node data to the gateway at faster rate.
4. Unavailability of data's such as PH level, potassium, Nitrogen etc related to thesoil.

PROPOSED SOLUTION

1. To control a device from longer distance from web application.
2. To get the weather details like wind speed, temperature, humidity from weatherstation through weather API.
3. To display the data in the web application.

References

1) IoT Enabled Smart Farming and Irrigation System:

Authors: M. Rohith, R Sainivedhana, Dr. N. Sabiyath Fatima
Published: IEEE 2021

Description: In this paper, authors have demonstrated a IoT enabled smart farming and irrigationsystem 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 plantscan be grown.

2) A Systematic Review of IoT Solutions for Smart Farming:

Authors: Emerson Navarro, Nuno Costa, and António Pereira
Published: MDPI 2020

Description: 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. Fromthe 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.

3) 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

Description: In this paper, authors have proposed a precision architecture for Smart Farming inorder 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 protectionof plant capital from the diseases

1) Internet of Things and LoRaWAN – Enabled Future Smart Farming

Authors: Bruno Citoni, Francesco Fioranelli, Muhammad A. Imran, Qammer H. Abbasi

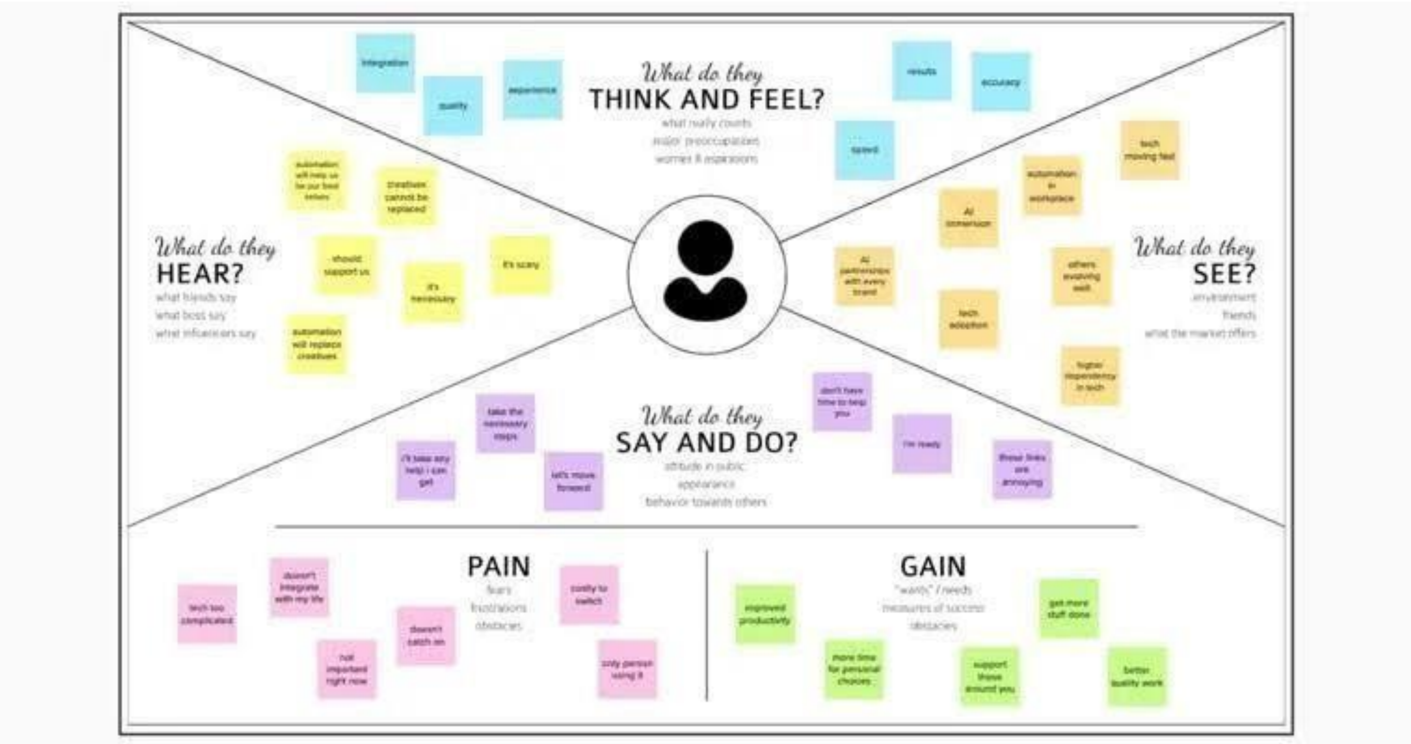
Published: IEEE 2019

Description: 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

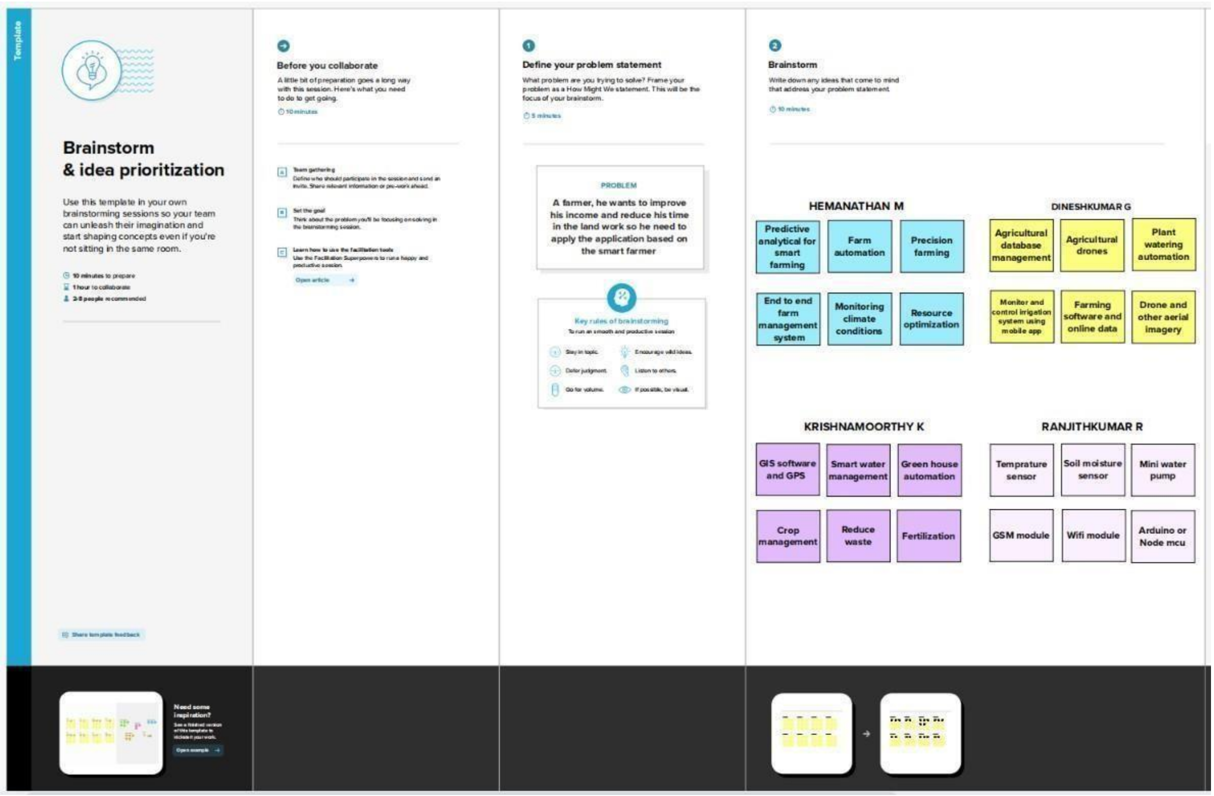
PROBLEM STATEMENT DEFINITION

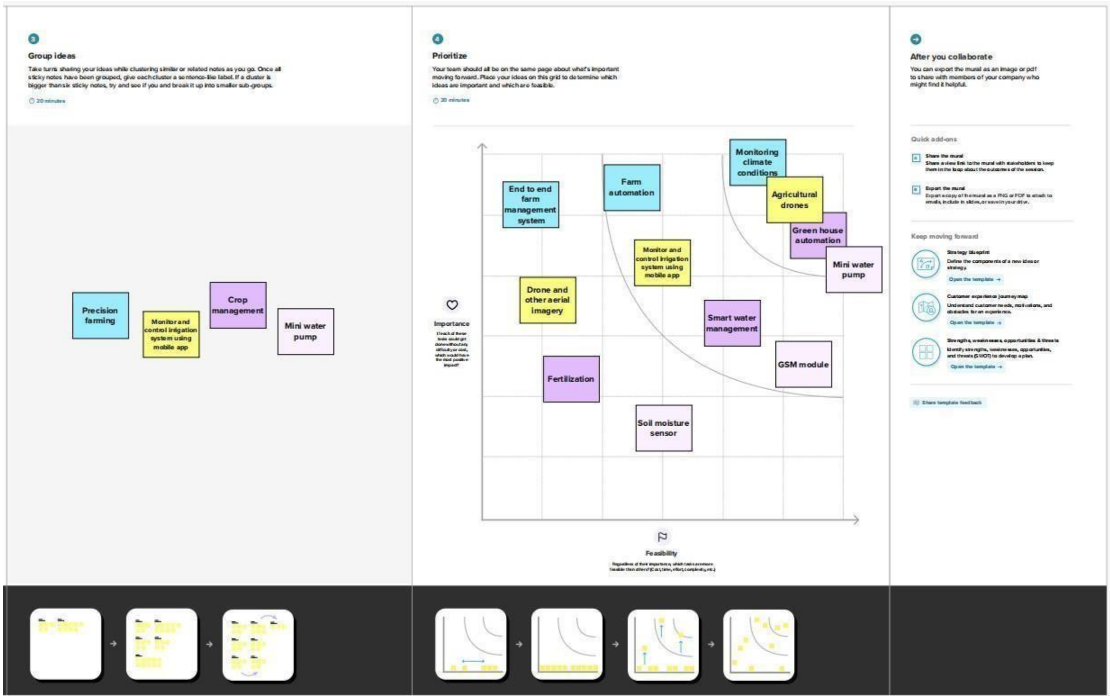
3. <u>IDEATION & PROPOSED SOLUTION</u>

EMPATHY MAP CANVAS

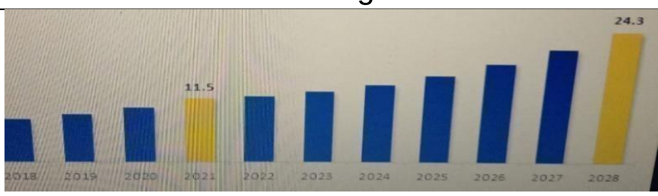


IDEATION & BRAINSTORMING





PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problemto be solved)	○ Most of the farmers use large portionsof farming land and it becomes very difficultto reach and track each cornerof large lands. Sometime there is a possibility of uneven water sprinkles. ○ Challenges faced by IOT in agriculuture are high adoption,security concerns, information lackness.
2.	Idea / Solution description	○ Smart Farming has enabled farmersto reduce waste and enhance productivitywith the help of sensors(light, humidity,temperature, soil moisture,etc..) ○ Further with the help of these sensors, farmers can monitor thefieldconditions from anywhere.
3.	Novelty / Uniqueness	○ Role of SENSORS : IOT smart agriculture products are designed tohelpmonitor crop fields using sensors and byautomating irrigationsystems. ○ As a result, farmers and associatedbrands can easily monitor the field conditions from anywhere without any hassle.
4.	Social Impact / Customer Satisfaction	○ Water conservation ○ Saves lot of time ○ Increased quality of production ○ Real time data and production insight. ○ Remote monitoring.
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	○ Scalability in smart farming refers totheadaptability of a system to increase the

FIT PROBLEM SOLUTION

1. CUSTOMER SEGMENT(S) CS Customers are the farmers and they are of types Marginal Farmers, Small Farmers, Semi-medium, Medium and Large. Farmers with large hectares of land require Smart-farming assistance to make things easy and reliable.	**6. CUSTOMER LIMITATIONS** CL Improper irrigation, Productivity issues, Difficulty in the management of inputs and outputs for farming activity, also climatic conditions affect the farmers, Reliability is less in traditional farming.	**5. AVAILABLE SOLUTIONS** AS Smart Farming has increased the productivity, and management of farming activity and timely reaction towards moisture, temperature, & climatic prediction. Automation via app/web-app has made it more sustainable than before.
2. PROBLEMS / PAINS PR Smart farming includes the Internet of things and this integrates the hardware and software part helping to make the automation easy like the irrigation facilities on timely basis and also prediction of climatic conditions before in hand gives the farmer a warning and to be prepared for the change .	**9. PROBLEM ROOT / CAUSE** RC Major problems the farmers facing is the soil erosion, climatic changes, and biodiversity loss. Expectations of the customers get ruined. Demand for the quality food. Investment in farming i.e. productivity.	**7. BEHAVIOR** BE The climatic condition and changes prediction is literally hard for the farmers and via smart farming its resolved.
3. TRIGGERS TO ACT TR Growing the awareness among people by showing up some ads or poster and also arranging campaigns to teach about Smart farming and also showing an example of it. **4. EMOTIONS** EM All the farmers want the traditional way only because they are emotionally connected to it but once they start using smart farming then the yield and productivity make them fulfilled.	**10. YOUR SOLUTION** SL To overcome all the problems and hurdles there is only one way and that is to integrate smart farming practices into the farming industry.	**8. CHANNELS of BEHAVIOR** CH Online mentoring can help farmer to use the smart farming technology. Farmers can buy the products from trusted shops and also week-based campaigns teaching them the smart farming practice can help them to convert from the traditional way to the modern way.

4. REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENT

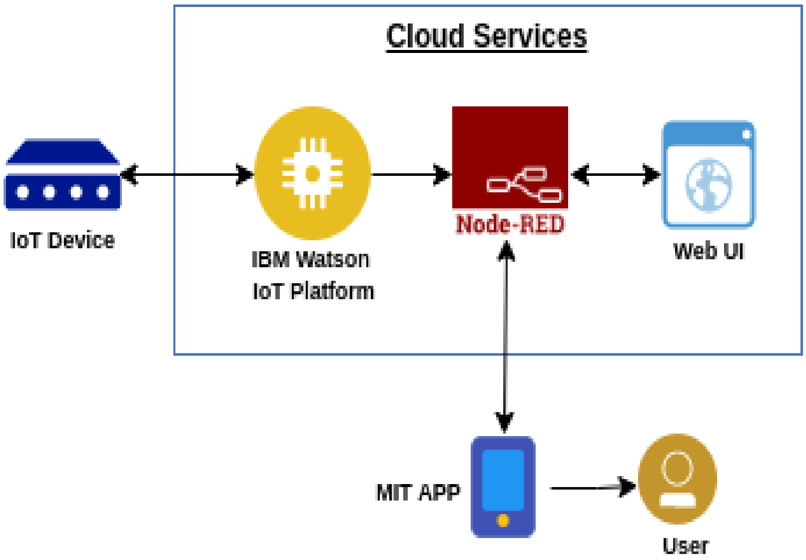
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration/Login	Phone Application And Wifi Module.
FR-2	User Permission	User permission for irrigation via Mobile Application and software Web UI Application
FR-3	Log in / App	Check Id / username And Check Roles access.
FR-4	Check whether details	Temperature and Humidity details
FR-5	Log out	Exit

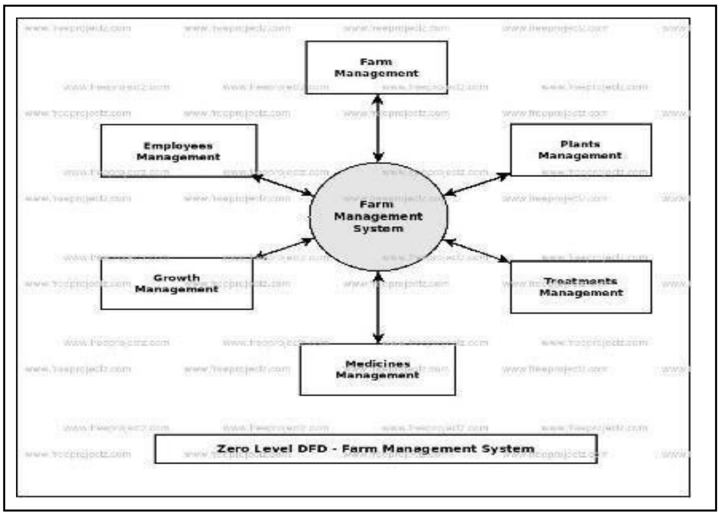
NON - FUNCTIONAL REQUIREMENT

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User-friendly Interface and learn to ability, effciency in use, remember ability, lack of errors in
NFR-2	**Security**	Sensitive and private data must be protected from their production until the decision-making and storage stages
NFR-3	**Reliability**	The shared protection achieves a better trade-off between costs and reliability. Accuracy of data
NFR-4	**Performance**	The process of the usage is easy and simple which allows to monitor and control with application's stability and accuracy
NFR-5	**Availability**	Automatic adjustment of farming equipment made possible by linking information like crops and weather
NFR-6	**Scalability**	we can upgrade the Smart farming Application automatic real time decision -making is feasible in anenvironment composed of dozens of thousand

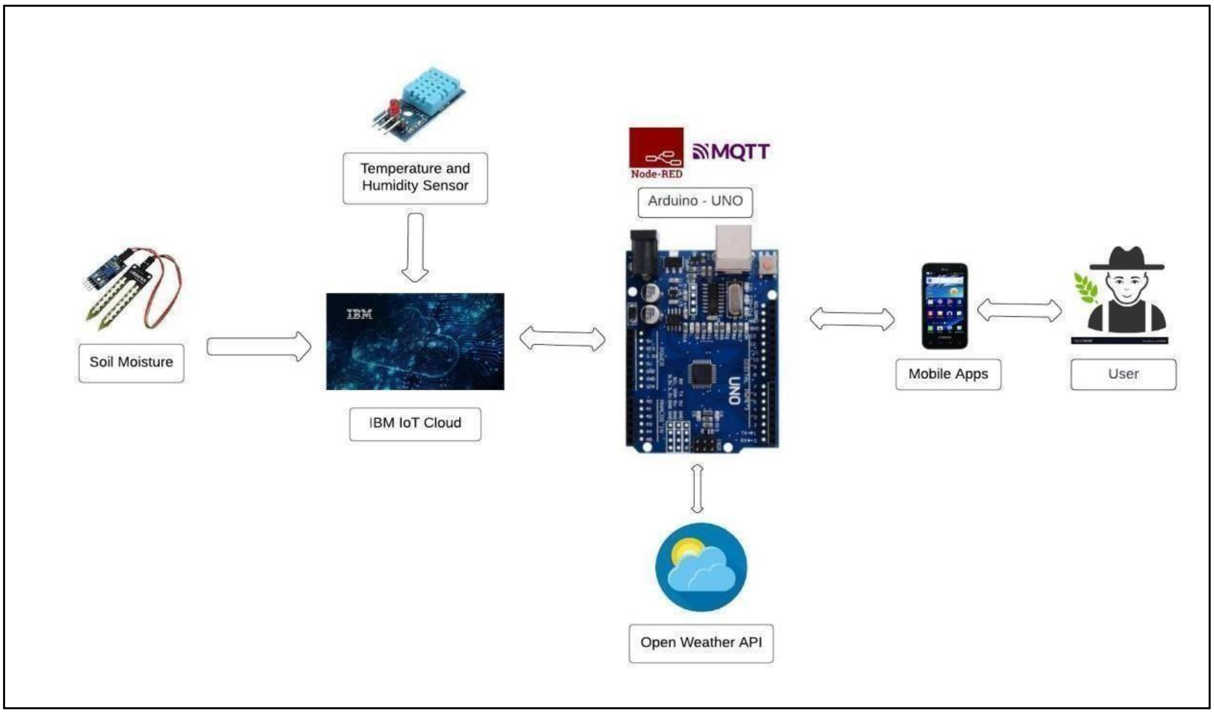
5. PROJECT DESIGN

DATA FLOW DIAGRAMS





SOLUTION & TECHNICAL ARCHITECTURE



USER STORIES

6. PROJECT PLANNING & SCHEDULING

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobileuser)	Registration	USN-1	As a user, I can register for the application byentering my email, password, and confirmingmy password.	I can access my account /dashboard	High	Sprint-1
	Permission	USN-2	As a user, I will receive confirmation emailonce Ihave registered for the application	I can receive confirmationemail & click confirm	High	Sprint-2
Customer (Webuser)	Login/App	USN-3	As a user, I can log into the application by entering email & password	I can register & access thedashboard with Login	High	Sprint-3
	Check whether rdetails	USN-4	As a user, I can register for the applicationthroughMobile application	Temperature and Humidity details	High	Sprint-4
Customer CareExecutive		USN-5	To make the user to interact withthe software	Data base to stored incloud services	High	Sprint-5
	Log out	USN-6	Exit	Sign out	High	Sprint-6

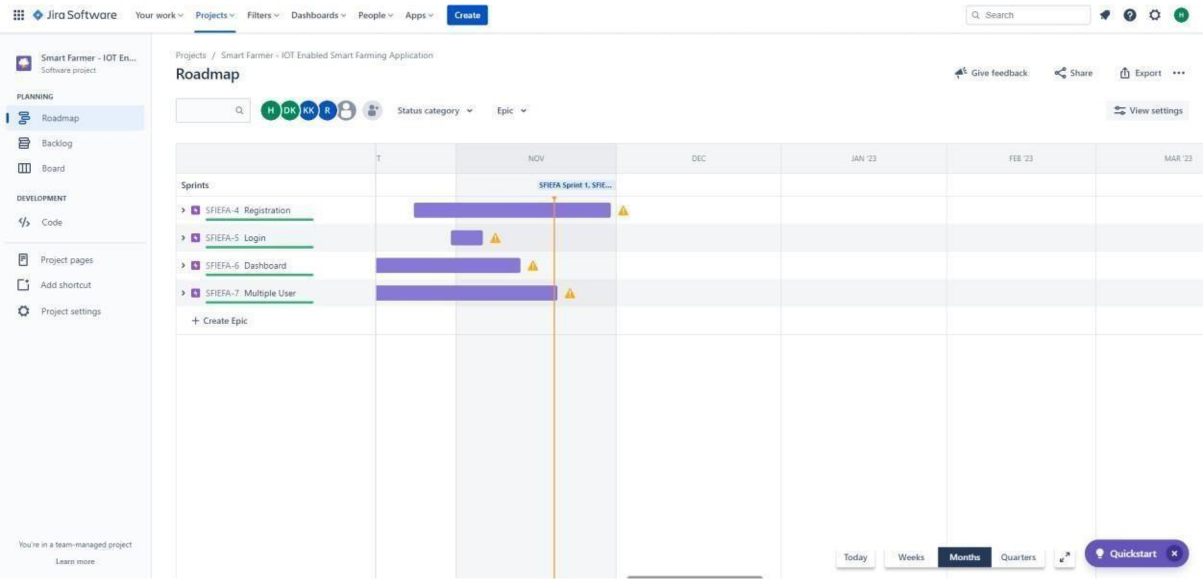
SPRINT PLANNING & ESTIMATION

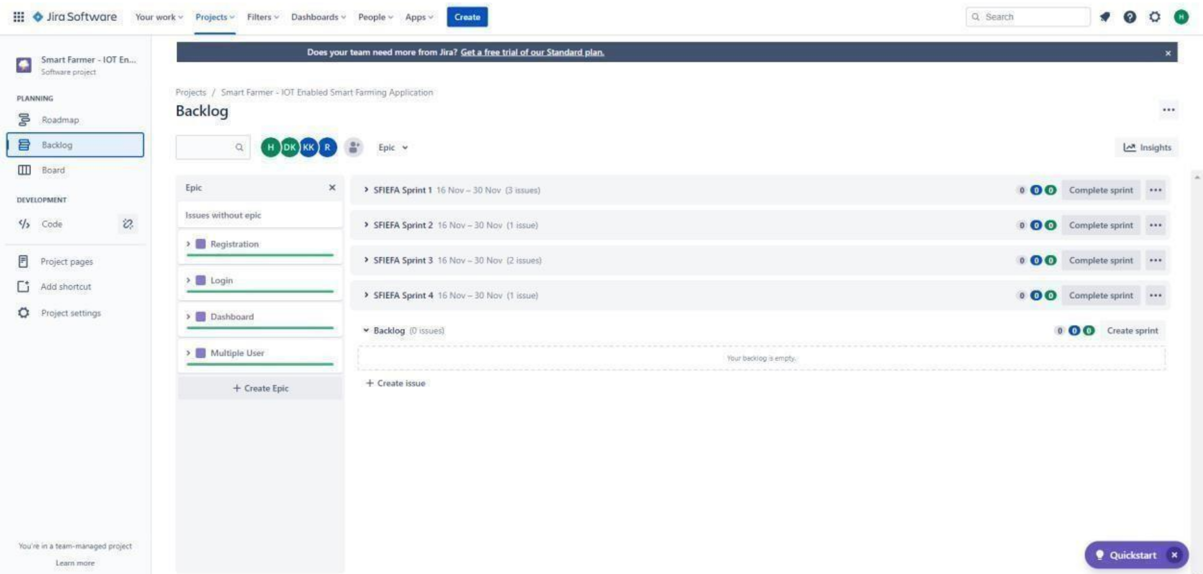
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Hardware	USN-1	Sensors and wi-fi modulewith python code .	2	High	Sri Rajeswari. E Mathibalan. M Thiruponpugazh. PRuhi Angel. A
Sprint-2	Software	USN-2	IBM Watson IoT platform, Workflows for IoTscenarios using Node-red	2	High	Sri Rajeswari. E Mathibalan. M Thiruponpugazh. p Ruhi Angel. A
Sprint-3	MIT app	USN-3	To develop an mobileapplication using MIT	2	High	Sri Rajeswari. E Mathibalan. M Thiruponpugazh .pRuhi Angel. A
Sprint-4	Web UI	USN-4	To make the user to interact with software .	2	High	Sri Rajeswari. E Mathibalan. M Thiruponpugazh. P Ruhi Angel. A

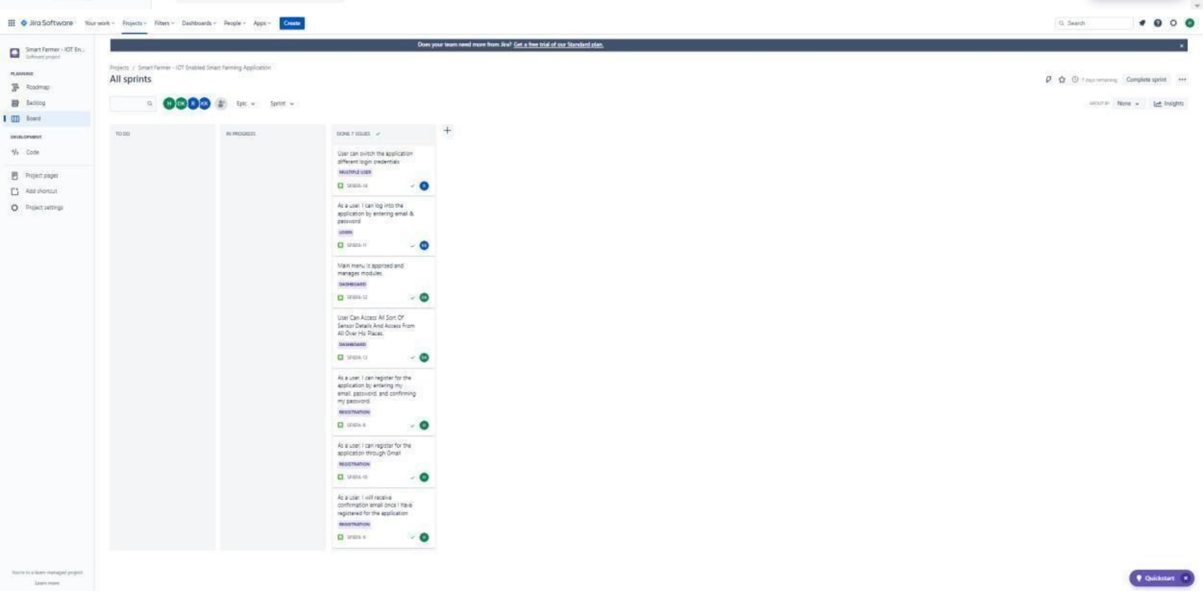
SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned EndDate)	Sprint Release Date (Actual)
Sprint -1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint -2	20	6 Days	31 Oct 2022	05 Nov 2022	20	5th NOV 2022
Sprint -3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12th NOV 2022
Sprint -4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14th NOV 2022

JIRA REPORTS FROM JIRA



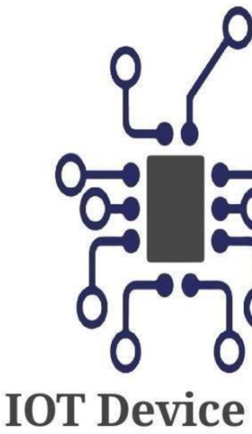


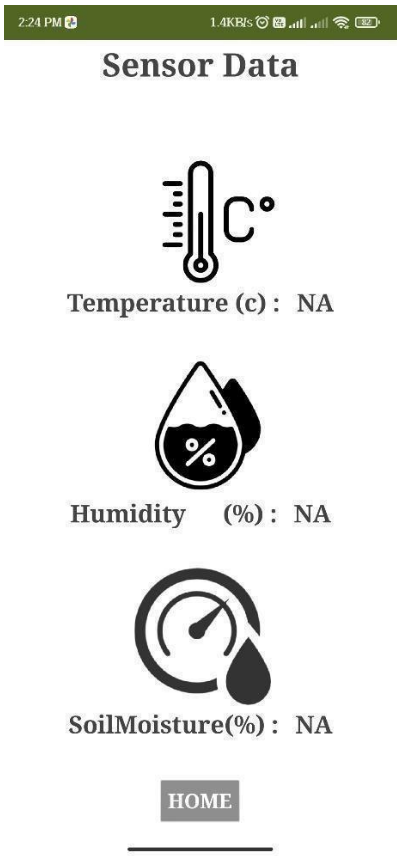


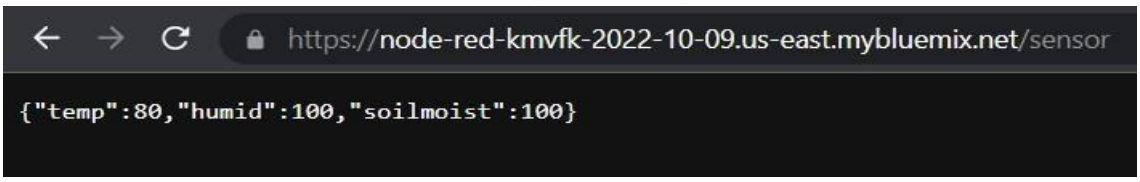
7. <u>CODING & SOLUTIONING</u>

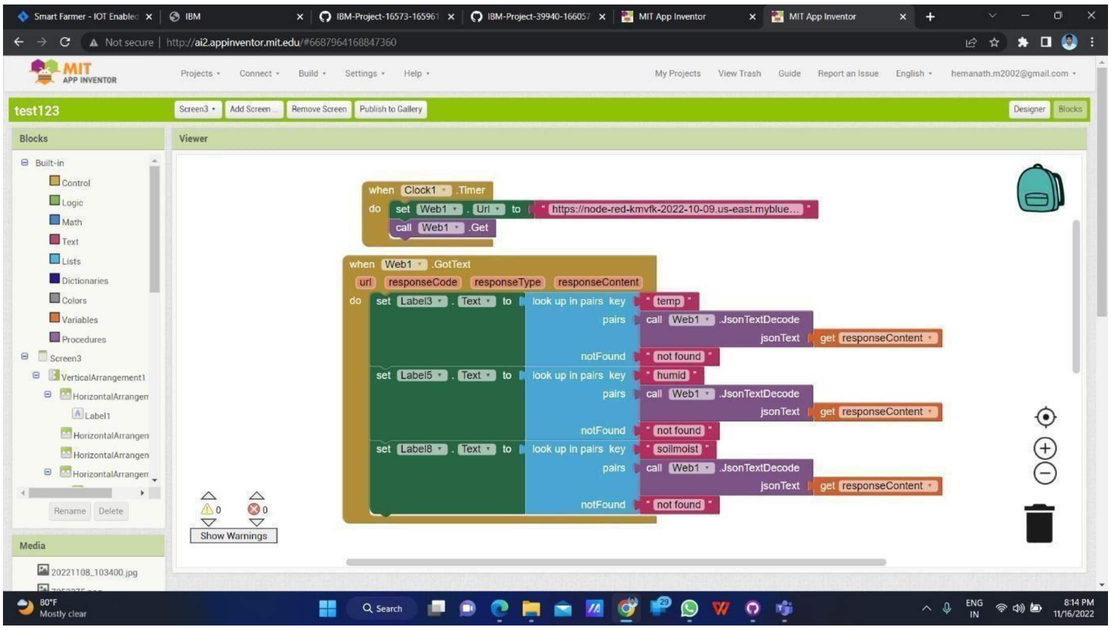
Feature 1

Monitor the Realtime sensor data





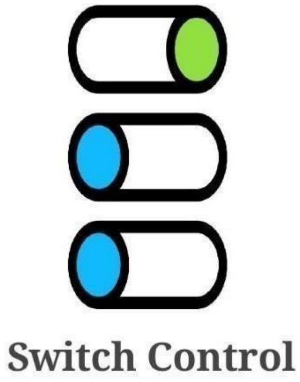


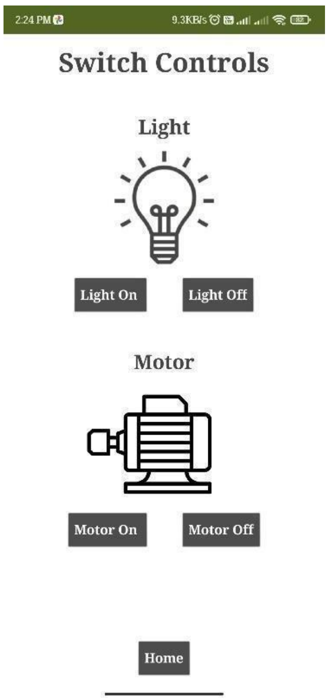


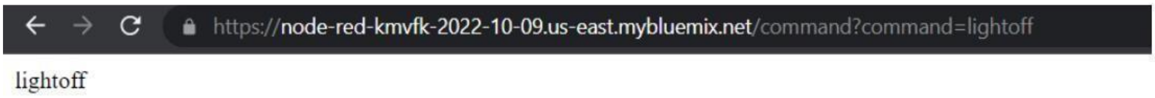
test123
Blocks
Viewer
when Clock1 .Timer
do set Web1 . Url to "https://node-red-kmvfk-2022-10-09.us-east.myblue..."
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do set Label3 . Text to look up in pairs key "temp"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
set Label5 . Text to look up in pairs key "humid"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
set Label8 . Text to look up in pairs key "soilmoist"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
Show Warnings

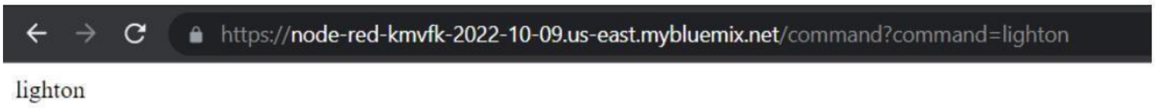
Feature 2

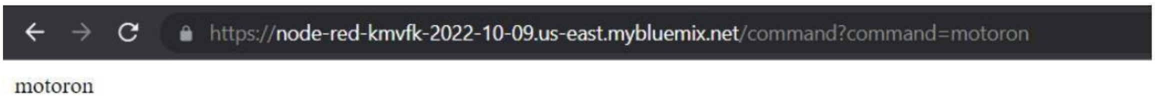
Control the switch remotely

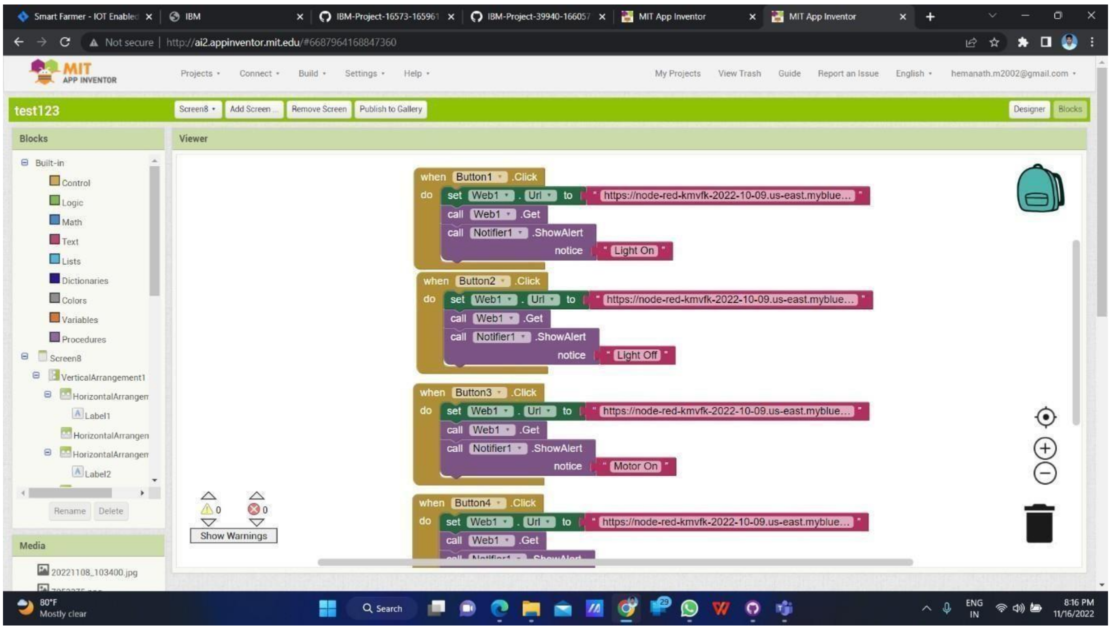








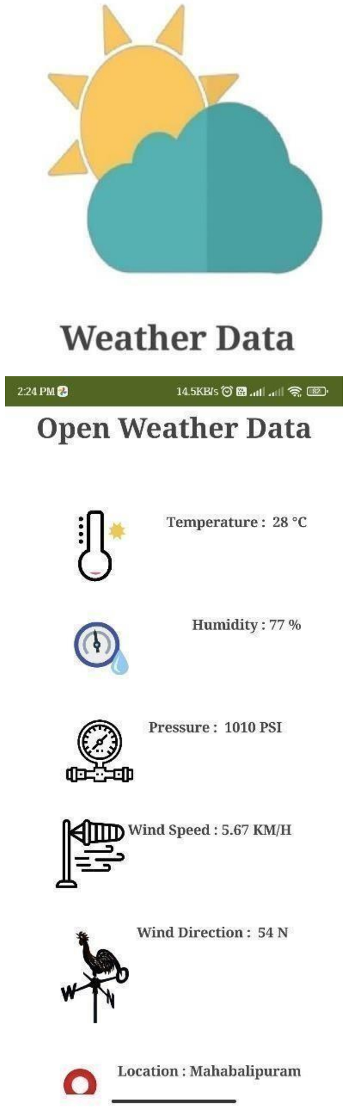


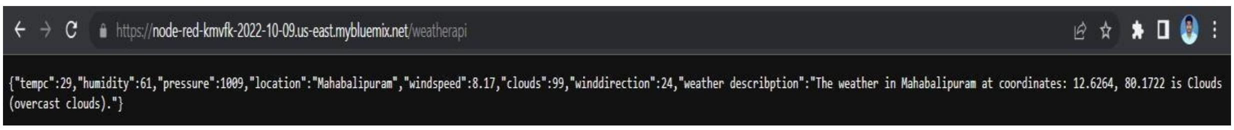


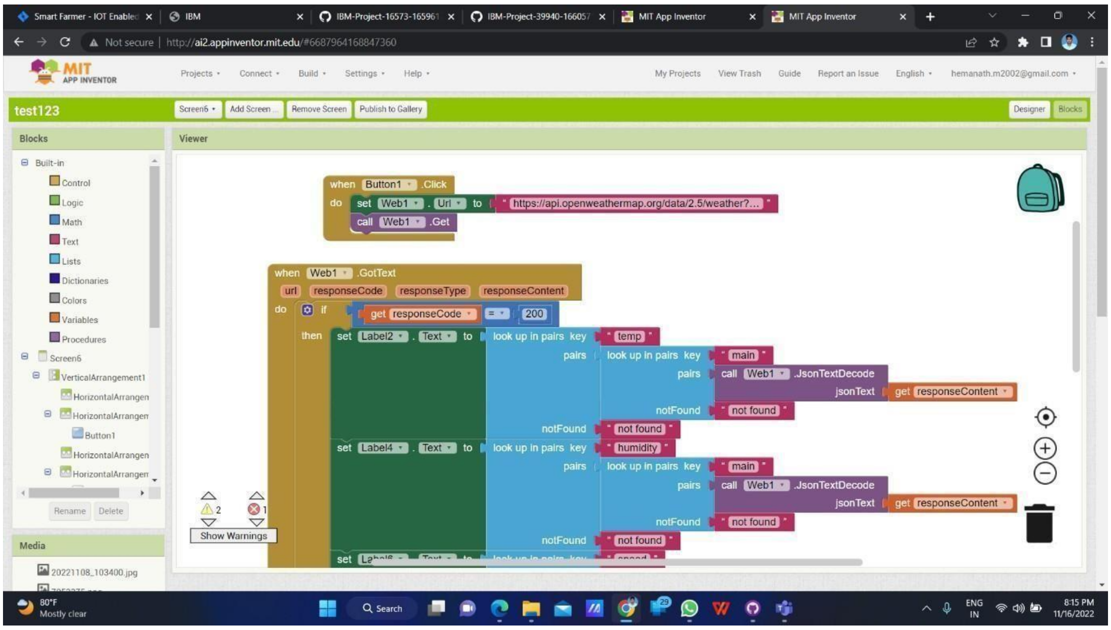
test123
Blocks
Viewer
when Button1 .Click
do set Web1 . Url to "https://node-red-kmvfk-2022-10-09.us-east.myblue..."
call Web1 .Get
call Notifier1 .ShowAlert
notice "Light On"
when Button2 .Click
do set Web1 . Url to "https://node-red-kmvfk-2022-10-09.us-east.myblue..."
call Web1 .Get
call Notifier1 .ShowAlert
notice "Light Off"
when Button3 .Click
do set Web1 . Url to "https://node-red-kmvfk-2022-10-09.us-east.myblue..."
call Web1 .Get
call Notifier1 .ShowAlert
notice "Motor On"
when Button4 .Click
do set Web1 . Url to "https://node-red-kmvfk-2022-10-09.us-east.myblue..."
call Web1 .Get
Show Warnings

Feature 3

Get the Realtime weather data






test123
Blocks
Viewer
when Button1 .Click
do set Web1 . Url to "https://api.openweathermap.org/data/2.5/weather?..."
call Web1 .Get
when Web1 .GotText
url responseCode responseType responseContent
do if get responseCode = 200
then set Label2 . Text to look up in pairs key "temp"
pairs look up in pairs key "main"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
notFound "not found"
set Label4 . Text to look up in pairs key "humidity"
pairs look up in pairs key "main"
pairs call Web1 .JsonTextDecode
jsonText get responseContent
notFound "not found"
notFound "not found"
Show Warnings

8. TESTING

Test Cases

1. Verify user is able to see the Login/Signup popup when user clicked on My account button.
2. Verify the UI elements in Login/Signup popup.
3. Verify user is able to log into application with Valid credentials.
4. Verify user is able to log into application with InValid credentials.

User Acceptance Testing

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

Hence a helpful and useful system is built for farmers to assist them in farming and also prevent them from natural calamities. It also saves farmers time to maintain all these things as this is working on cloud he can turn on/off motor from anywhere so basically it helps farmers and make them relived thus helping our economy to grow.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGE

- Communicating the device at larger distance through web application. It will playan important role in reducing the man power and travellingexpenses of a farmer.
- Monitoring the parameter like temperature, humidity etc will play an importantrole in improving the growth of the plant.
- Integrating the weather station to the web browser will provide the details of status of the cloud, wind speed etc. It will allow the farmer toprevent their plantsfrom natural calamities.

DISADVANTAGE

1. Since the real time sensor will be connected to the controller, the controller requires continuous supply of internet to transfer the data.
2. Non availability of weather prediction for long period of time. Since the long weather prediction require additional payment to open weather.

11. CONCLUSION

The various parameters like temperature, humidity etc were monitored using web application. The data from weather station like wind speed, temperature, humidity etc were displayed in the web browser. The device like motor, light etc can also controlled by the web application.

12. FUTURE SCOPE

1. The various data's of soil nutrients is not added in the web browser, that can be added to the web application.
2. Long range forecast is not available in the web application, it can also be added to provide accurate information about weather.
3. Controlling the device through mobile application and voice will play important role in enhancing this project.
4. Providing the GPS and GIS information will also improve productivity of the farmer.

13. APPENDIX

Source Code

1) Python Code

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

organization = "22r9m3"
deviceType = "123"
deviceId = "1234567"
authMethod = "token"
authToken = "12345678"


def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="lighton":
      print ("led is on")
   elif status == "lightoff":
      print("led is off")
   elif status == "motoron":
      print("motor is on")
   elif status == "motoroff":
      print("motor is off")
   else :
      print ("please send proper command")
```

```
    #print(cmd)



try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #..............................................

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

deviceCli.connect()

while True:


        temp=random.randint(0,100)
        humid=random.randint(0,100)
        soilmoist=random.randint(0,100)

        data = { 'temp' : temp, 'humid': humid, 'soilmoist': soilmoist }

        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
humid,"Soilmoisture = %s %%" % soilmoist, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()
```

2) Wokwi Simulator

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

```
#define MOTOR 4

DHT dht (DHTPIN, DHTTYPE);

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



#define ORG "22r9m3"//IBM ORGANITION ID
#define DEVICE_TYPE "123"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "1234567"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
float h, t;



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;
PubSubClient client(server, 1883, callback ,wifiClient);


void setup()
{
  Serial.begin(115200);
  dht.begin();
  pinMode(LED,OUTPUT);
  pinMode(MOTOR,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 += "," "\"soilmoist\":";
  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++) {
    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);
  if(data3=="lighton")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);
  }
  else if(data3=="motoron")
  {
```

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

GitHub & Project Demo Link

GitHub : https://github.com/IBM-EPBL/IBM-Project-53453-1661407479/tree/main