Smart Farmer – IoT Enabled Smart Farming Application

Submitted
by

Team Id	PNT2022TMID47404
Team Lead	SANTHOSH R
Team Members	ANANTHAN V GOPINATH S HARUN RAGAVAN V

CONTENTS

1. Introduction

1.1 Project Overview

IoT-based farming systems help farmers monitor various parameters of their fields, such as temperature, humidity, soil moisture, sump water, tank water, single phase, and three phase using several sensors. A farmer can monitor all sensor her parameters through his web or mobile application without being near his field. Crop irrigation is one of the most important tasks for a farmer. By monitoring sensor parameters and controlling motor pumps from a mobile application, irrigation or crop movement decisions can be made.

1.2 Purpose

Better production management leads to better cost control and less waste. For example, the ability to eliminate abnormal animal health conditions helps eliminate the risk of yield loss. In addition, automation increases efficiency. Smart Farming forms the ecological base of faming. Minimizing the site-specific application of inputs such as fertilizers and pesticides in precision farming systems reduces leaching issues and digester gas emissions.

2. Literature Survey

2.1 Existing Problem

IoT’s Smart Farming improves entire farming systems by monitoring fields in real time. With the help of sensors and internet connectivity, the Internet of Things in culture has not only saved the celebrity era, but has also encouraged the abuse of resources such as water and electricity. Climate plays a very important role in agriculture. Mis-knowledge of climate also significantly reduces the quantity and quality of crop production. Precision agriculture/precision farming is one of his best known applications of IoT in agriculture. It enables smart farming applications such as livestock monitoring, field observation, and inventory monitoring, making farming practices more precise and controllable. To make greenhouses smart, IoT has enabled weather stations to automatically adjust climate conditions according to a specific set of instructions. IoT implementation in the greenhouse eliminated human intervention, making the whole process more cost-effective and more accurate.

2.2 References

1.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.

2.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 bestcrop for the land.

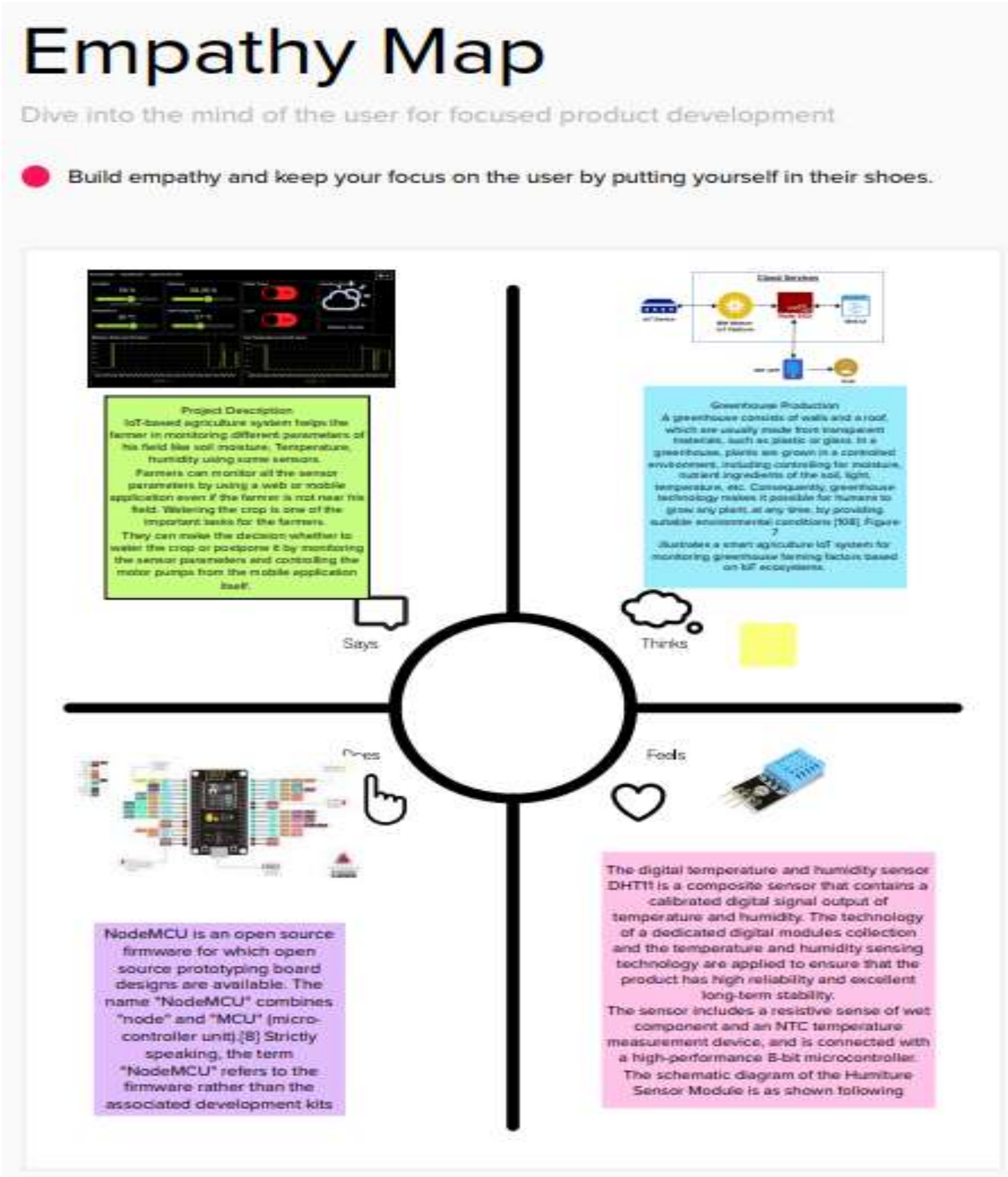
3. H.G.C.R. Laksiri, H.A.C. Dharmagunawardhana, J.V. Wijayakulasooriya [3] Development of an effective loT-based smart irrigation system is also a crucial demand for farmers in the field of agriculture. This research develops a low-cost, weather-based smart watering system. To begin, an effective drip irrigation system must be devised that can automatically regulate water flow to plants based on soil moisture levels. Then, to make this water-saving irrigation system even more efficient, an IoT-based communication feature is added, allowing a remote user to monitor soil moisture conditions and manually adjust water flow.

2.3 Problem Statement Solution

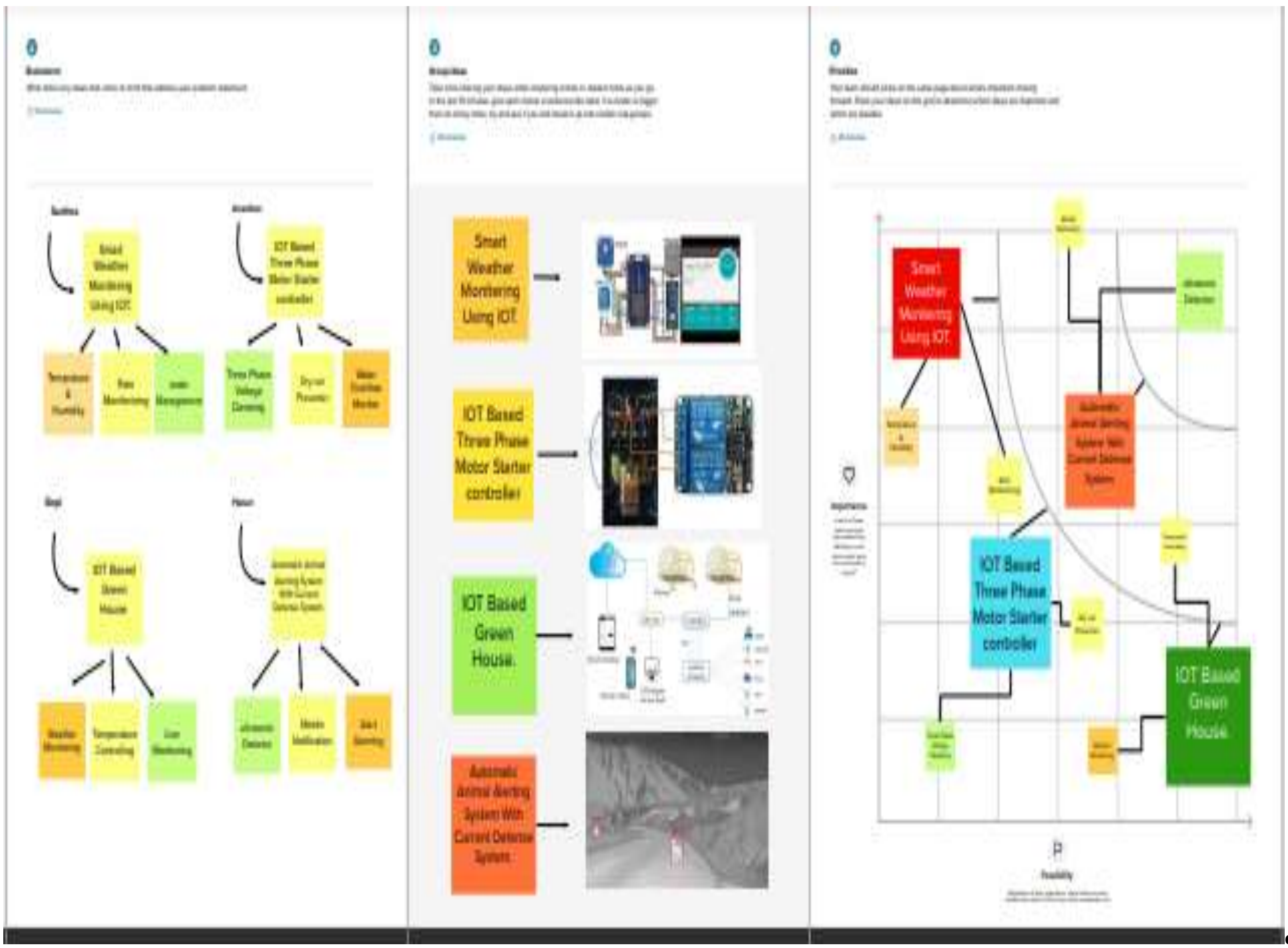
Traditional agriculture and related sectors are unable to meet the demands of modern agriculture, which requires high yield, quality and efficient production. Therefore, it is very important to look to modernize existing methods and use information technology and data over a period of time to predict the best possible productivity and country-suitable crops. The introduction of high-speed internet, mobile devices, and access to reliable and low-cost satellites is just some of the key technologies characterizing the precision farming trend in agriculture. Precision agriculture is one of his best-known applications of IoT in the agricultural sector, with many organizations around the world using the technology. Products and services used include VRI Optimization, Soil Moisture Probes and Virtual Optimizer PRO. Optimize variable rate irrigation (VRI) to maximize profitability, improve yields and increase water efficiency in irrigated fields with variable terrain and soils. IoT is making great strides in areas such as manufacturing, healthcare, and automotive. When it comes to food production, transportation and storage, it offers a range of options to improve his per capita food availability in India. Sensors that provide information on soil nutrient status, pest infestation, moisture conditions, etc. can be used to improve crop yields over time. Here are some examples of problem areas related to agriculture and related sectors where IoT applications would benefit:

3. Ideation & Proposed Solution

3.1 Prepare Empathy Map



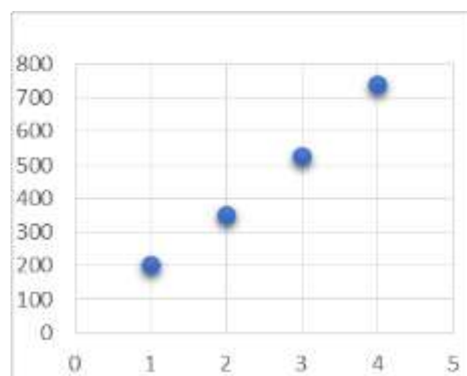
3.2 Ideation



c

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• Watering the field is a difficult process, Farmers have to wait in the field until the water covers the whole farm field. • Power Supply is also one of the problems. In Village Side, the power supply may vary. • The Biggest Challenges Faced by IoT in the Agricultural Sector are Lack of Information, High Adoption, Cost and Security Concerns, etc
2.	Idea / Solution description	• As is the case of precision Agriculture Smart Farming Technique Enables Farmers better to monitor the fields and maintain the humidity level accordingly. • The Data collected by sensors, In terms of humidity, temperature, moisture, and dew detections help in determining the weather pattern in Farms. So cultivation is done for suitable crops.
3.	Novelty / Uniqueness	**ALERT MESSAGE** – IoT sensor nodes collect information from the farming environment, such as soil moisture, air humidity, temperature, nutrient ingredients of soil, pest images, and water quality, then transmit collected data to IoT backhaul devices.

		REMOTE ACCESS – It helps the farmer to operate the motor from anywhere.
4.	Social Impact / Customer Satisfaction	• Reduces the wages for labors who work in the agricultural field. • It saves a lot of time. • IoT can help improve customer relationships by enhancing the customer's overall experience. • Easily identify maintenance needs, build better products, send personalized communications, and more. • IoT can also help e-commerce businesses thrive and increase sales. It make a wealthy society
5.	Business Model (Revenue Model)	Revenue (No. of Users vs Months) User  Months
6.	Scalability of the Solution	Scalability in smart farming refers to the adaptability of a system to increase the capacity, for example, the number of technology devices such as sensors and actuators, while enabling timely analysis.

3.4 Proposed Solution Fit

4. Requirement Analysis

4.1 Functional Requirement

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 cropdiseases
FR-4	Manage Modules	Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature detailsHumidity details
FR-6	Data Management	Manage the data of weather conditions Manage the data of crop conditions Manage the data of live stock conditions

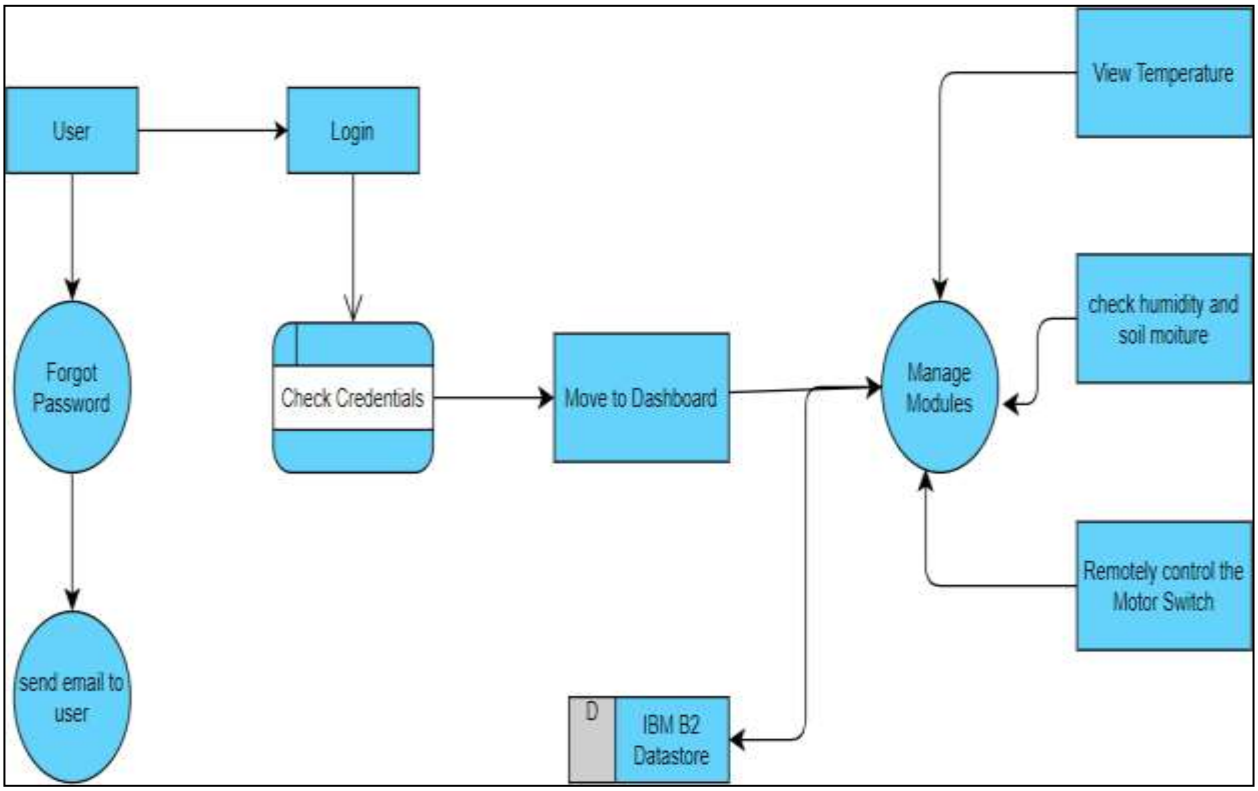
4.2 Non-Functional Requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	User friendly guidelines for users to avail the features. Most simplistic user interface for ease of use.
NFR-2	Security	All the details about the user are protected from unauthorized access. Detection and identification of any misfunctions of sensors.
NFR-3	Reliability	Implementing Mesh IoT Networks Building a Multi-layered defence for IoT Networks.
NFR-4	Performance	The use of modern technology solutions helps to achieve the maximumperformances thus resulting in better quality and quantity yields.
NFR-5	Availability	This app is available for all platforms
NFR-6	Scalability	Scalability refers to the ability to increase available resources and systemcapability without the need to go through a major system redesign or implementation.

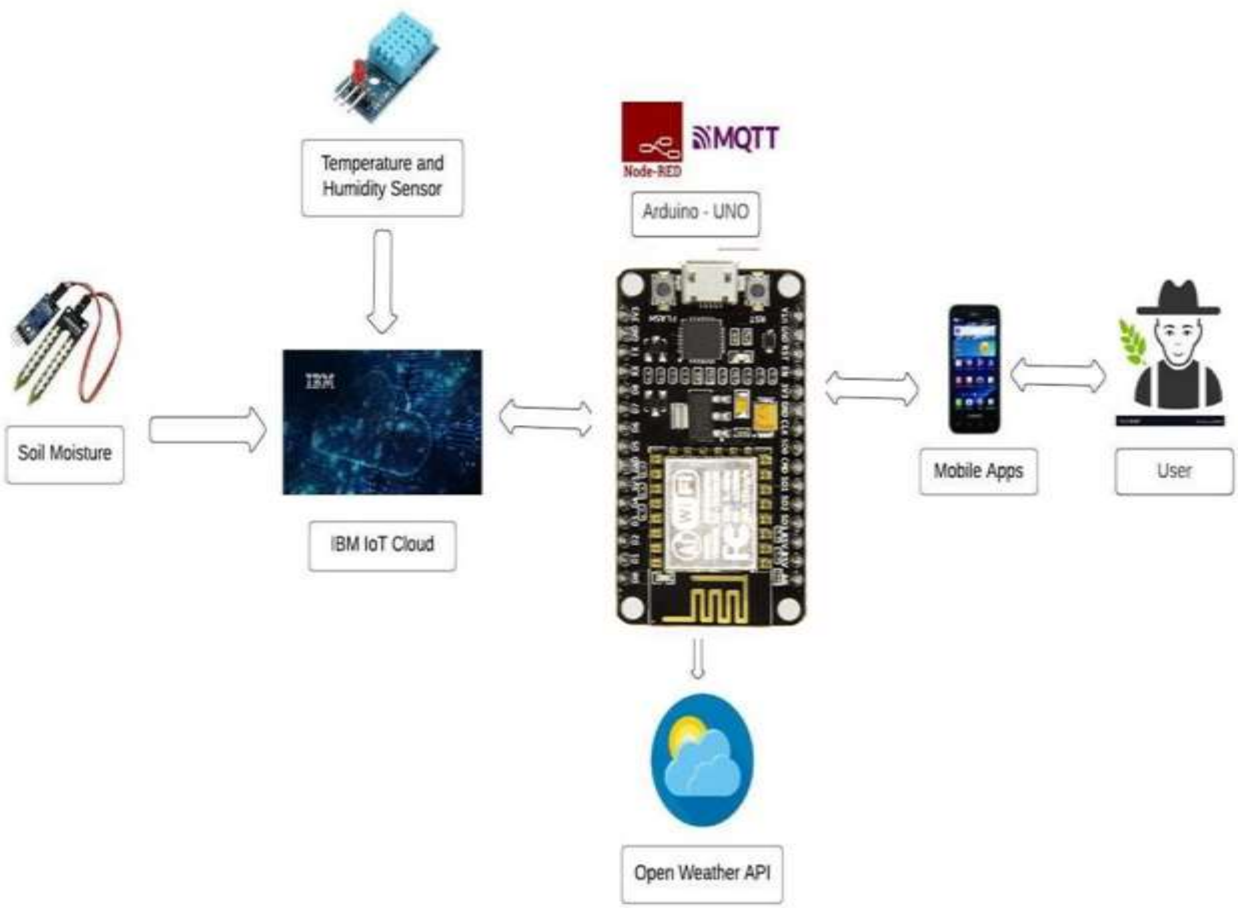
5. Project Design

5.1 Data Flow Diagram

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



5.2 Solution Architecture



- The different soil parameters (temperature, humidity, Soil Moisture) are sensed using different sensors, and the obtained value is stored in the IBM cloud.
- Arduino UNO is used as a processing unit that processes the data obtained from sensors and weather data from weather API.
- Node-red is used as a programming tool to wire the hardware, software, and APIs. The MQTT protocol is followed for communication.
- All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could make a decision through an app, whether to water the crop or not depending upon the sensor values. By using the app they can remotely operate the motor switch.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release	User Type
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password.	I can access my account / dashboard	High	Sprint-1	Customer (Mobile user)
		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1	
		USN-3	As a user, I can register for the application through Gmail		Medium	Sprint-1	
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1	
Customer (Web user)	Dashboard	USN-5	As a User can view the dashboard, and this dashboard include the check roles of access and then move to the manage modules.	I can view the dashboard in this smart farming application system.	High	Sprint 2	Customer (Web user)
		USN-6	User can remotely access the motor switch	In the smart farming app	High	Sprint 3	

6.Project Planning & Scheduling

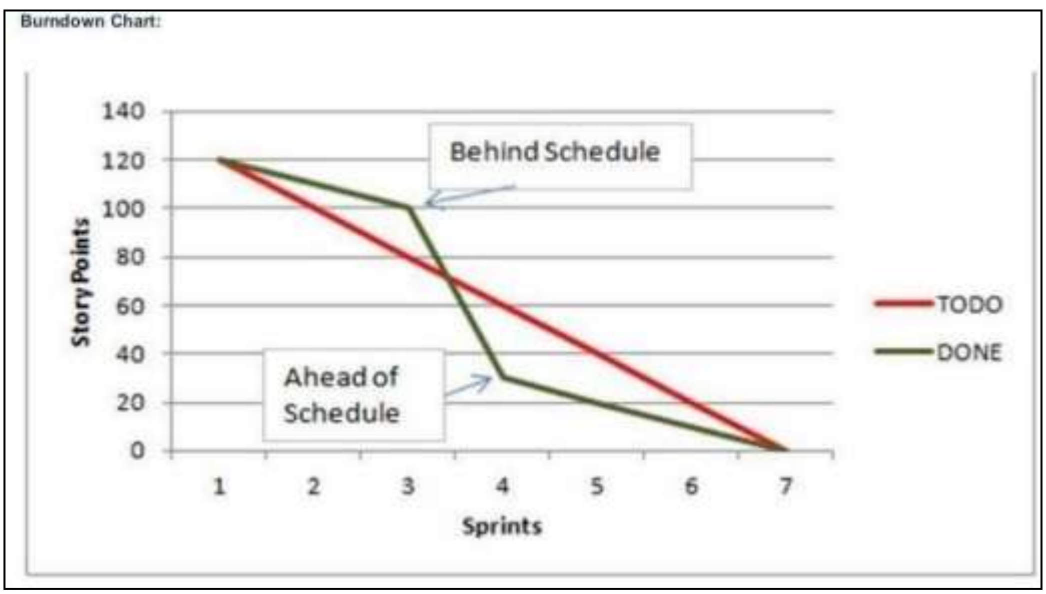
6.1 Sprint Planning & Estimation

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 code	2	High	Santhosh, harun ragavan
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform workflow for IoT Scenarios using Node-RED	2	High	Santhosh, harun ragavan, Gopinath, Ananthan
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmerproject using MIT App Inventor	2	High	Santhosh, Ananthan
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Santhosh, Gopinath
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Santhosh, harun ragavan,Ananthan, Gopinath

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	16 Nov 2022

6.3 JIRA Report



7. Coding & Solutioning

7.1 Feature - 1

Receiving commands from IBM cloud using C++ program

```
#include <ESP8266WiFi.h>
//#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#include "DHT.h"// Library for dht11
#define DHTPIN D3     // what pin we're connected to
#define DHTTYPE DHT11   // define type of sensor DHT 11
#define LED D0
#define pot A0
#define swat D5
#define twat D6
#define pone D7
#define ptwo D8

DHT dht (DHTPIN, DHTTYPE);// creating the instance by passing pin and typr of dht connected
int moister =0;
int swatv =0;
int twatv =0;
int ponev =0;
int ptwov =0;
void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);
//-------credentials of IBM Accounts------

#define ORG "1dzfs1"//IBM ORGANITION ID
#define DEVICE_TYPE "SMART_FORMER"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "6383319751"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "DQIhkT2xKA-Xk*Ztau"     //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

char ssid[] = "MY WIFI";
char pass[] = "96559655";

//-----------------------------------------
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();
  pinMode(LED,OUTPUT);
  pinMode(pot,INPUT);
  pinMode(swat,INPUT);
  pinMode(twat,INPUT);
  pinMode(pone,INPUT);
  pinMode(ptwo,INPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{
moister = analogRead(pot);
swatv = digitalRead(swat);
twatv = digitalRead(twat);
ponev = digitalRead(pone);
ptwov = digitalRead(ptwo);

  h = dht.readHumidity();
  t = dht.readTemperature();
  Serial.print("temp:");
  Serial.println(t);
  Serial.print("Humid:");
  Serial.println(h);

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



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

void PublishData(float temp, float humid,float moister) {
  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 += "," "\"moister\":";
  payload += moister;
  payload += "," "\"swat\":";
```

```
  payload += swatv;
  payload += "," "\"twat\":";
  payload += twatv;
  payload += "," "\"pone\":";
  payload += ponev;
  payload += "," "\"ptwo\":";
  payload += ptwov;
  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(ssid, pass);//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: ");
```

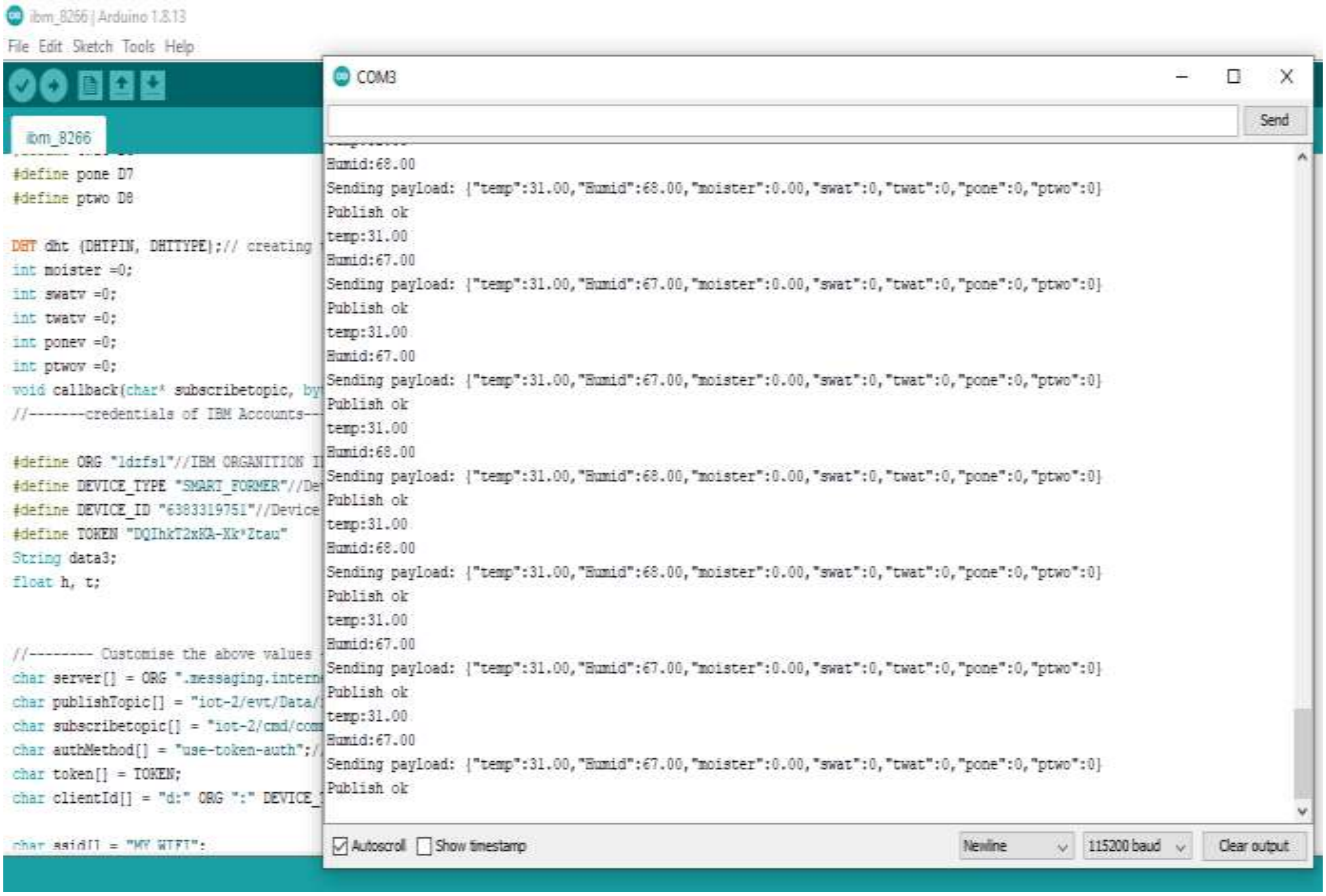
```
  Serial.println(WiFi.localIP());
}

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

Output

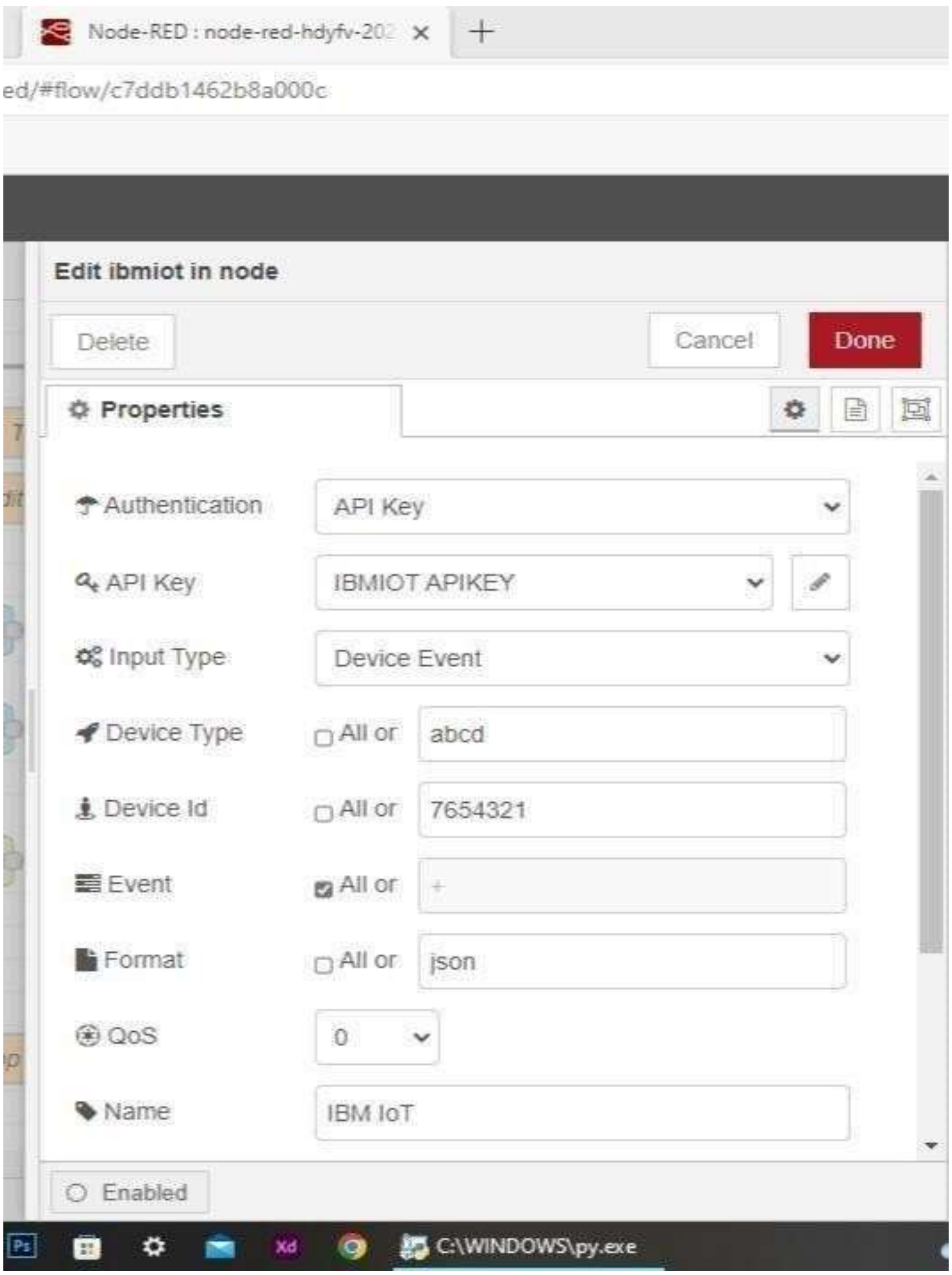


ibm_8266 | Arduino 1.8.13
File Edit Sketch Tools Help
ibm_8266
#define pone D7
#define ptwo D8
DHT dht (DHTPIN, DHTTYPE);// creating
int moister =0;
int swatv =0;
int twatv =0;
int ponev =0;
int ptwov =0;
void callback(char* subscribetopic, by
//-------credentials of IBM Accounts--
#define ORG "1dzfs1"//IBM ORGANITION I
#define DEVICE_TYPE "SMART_FORMER"//De
#define DEVICE_ID "6383319751"//Device
#define TOKEN "DQIhkT2xKA-Xk*Ztau"
String data3;
float h, t;
//-------- Customise the above values
char server[] = ORG ".messaging.intern
char publishTopic[] = "iot-2/evt/Data/
char subscribetopic[] = "iot-2/cmd/com
char authMethod[] = "use-token-auth";/
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_
char ssid[] = "MY WIFI";
COM3
Send
Humid:68.00
Sending payload: {"temp":31.00,"Humid":68.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
temp:31.00
Humid:67.00
Sending payload: {"temp":31.00,"Humid":67.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
temp:31.00
Humid:67.00
Sending payload: {"temp":31.00,"Humid":67.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
temp:31.00
Humid:68.00
Sending payload: {"temp":31.00,"Humid":68.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
temp:31.00
Humid:68.00
Sending payload: {"temp":31.00,"Humid":68.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
temp:31.00
Humid:67.00
Sending payload: {"temp":31.00,"Humid":67.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
temp:31.00
Humid:67.00
Sending payload: {"temp":31.00,"Humid":67.00,"moister":0.00,"swat":0,"twat":0,"pone":0,"ptwo":0}
Publish ok
Autoscroll
Show timestamp
Newline
115200 baud
Clear output

7.2 Feature – 2

Configuration of Node-Red to send commands to IBM cloud

ibmiot out node I used to send data from Node-Red to IBM Watson device. So, after adding it to the flow we need to configure it with credentials of our Watsondevice.



Here we add two buttons in UI
1 -> for motor on

2 -> for motor off

We used a function node to analyses the data received and assign command toeach number.

The Java script code for the analyses is:

if(msg.payload===1)

msg.payload={"command”:

“ON"}; else if(msg.payload===0)

msg.payload={"command”:

“OFF"};

Then we use another function node to parse the data and get the command and represent it visually with text node.

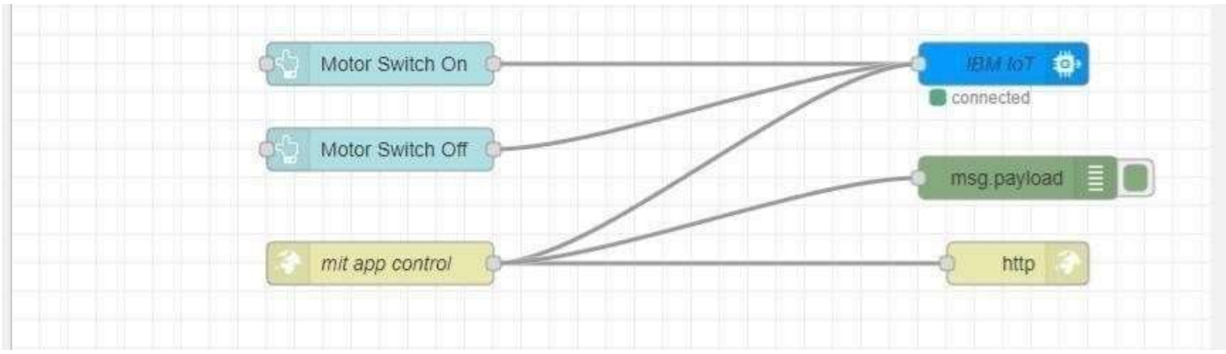
The Java script code for that function node is:

- var state=msg.payload;
  msg.payload = state.command;
- return msg;

The above images show the java script codes of analyser and state function nodes.

Then we add edit Json node to the conversion between JSON string & object and finally connect it to IBM IoT Out.

Edit JSON node needs to be configured like this



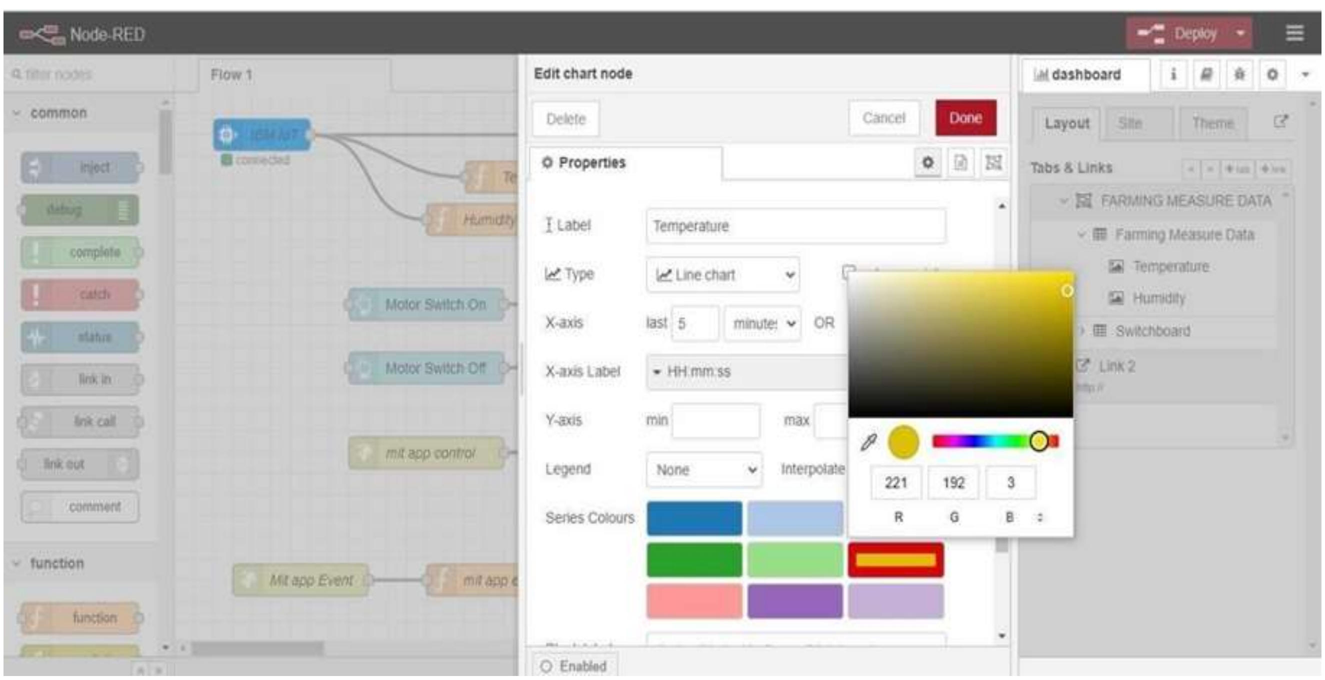
This is the program flow for sending commands to IBM cloud.

Adjusting User Interface

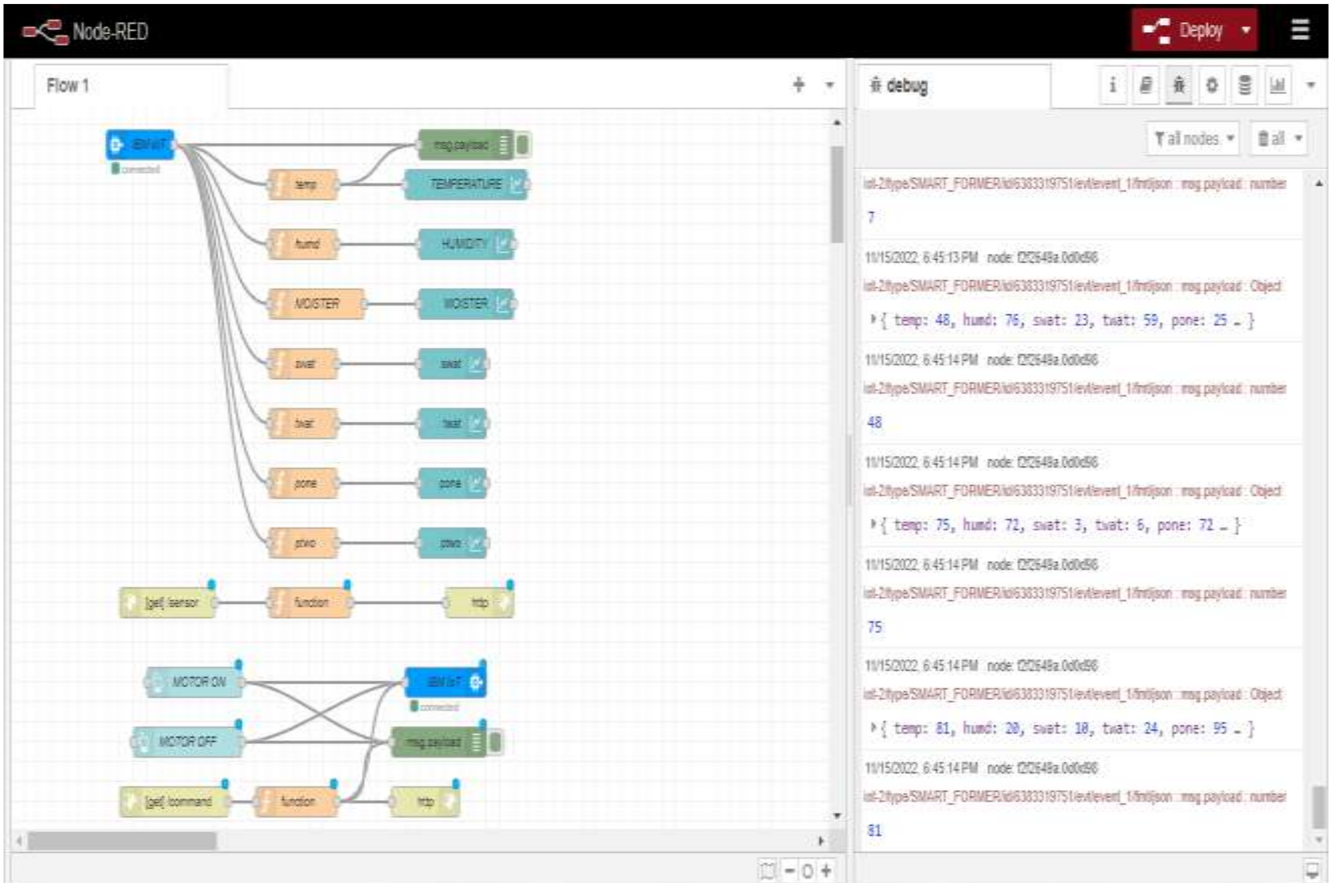
In order to display the parsed JSON data a Node-Red dashboard is created

Here we are using Gauges, text and button nodes to display in the UI and helps to monitor the parameters and control the farm equipment.

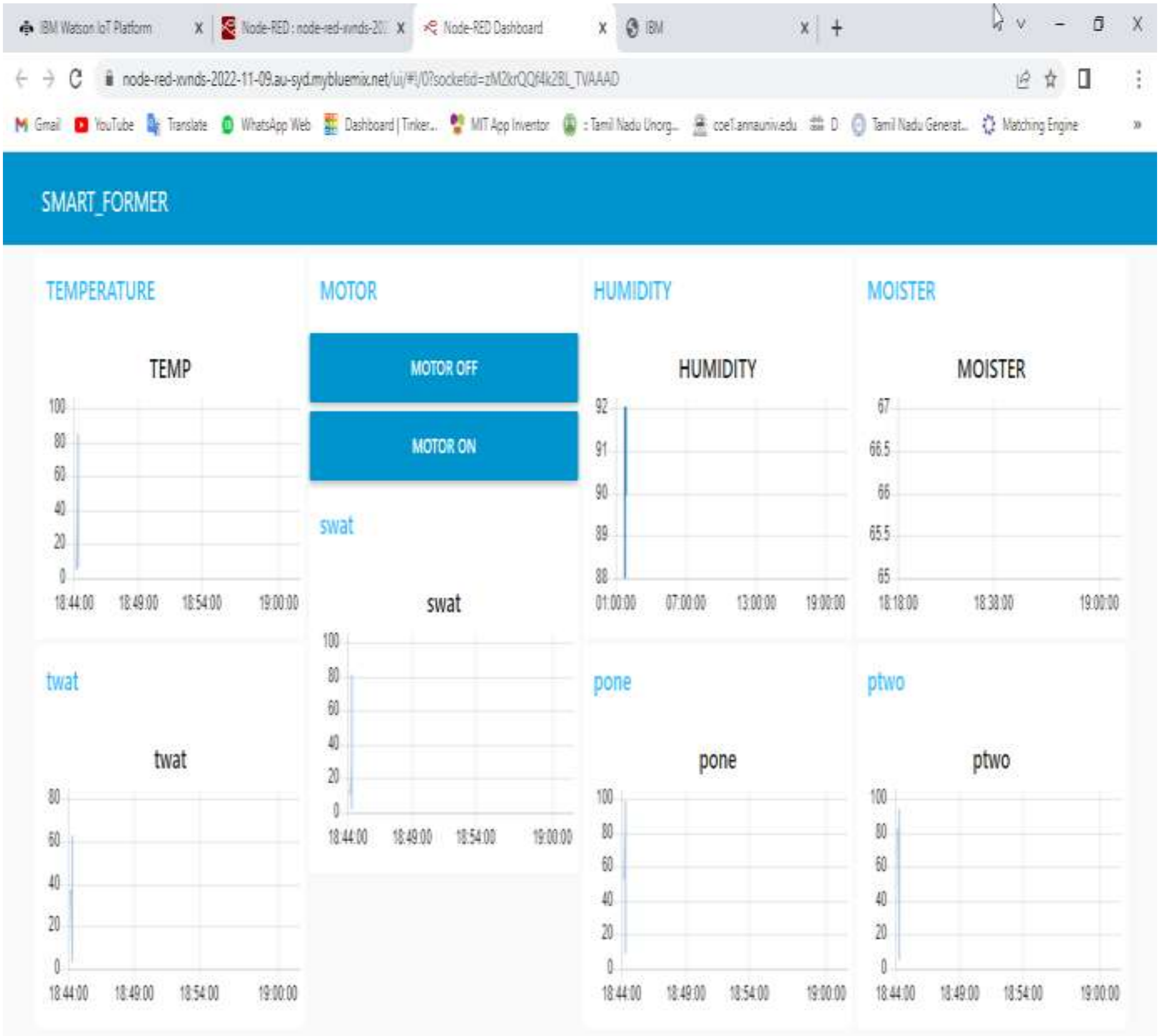
Below images are the Gauge, text and button node configurations.



Complete Program Flow



Web APP UI Home Tab



Mobile App UI

SMART FARMER APPLICATION

Blocks

SCREEN 2

SCREEN 2

WRONG PASSWORD


7:04
37%
SMART - விவசாயி
TEMPERATURE : 81
HUMIDITY : VARLA
MOISTER : VARLA
SUMP WATER: 10
TANK WATER : 24
PHASE 1 : 95
PHASE 2 : 67
MOTOR PUMP
MOTOR ON
MOTOR OFF

SCREEN 3

8.Testing

8.1 Test Cases

Shopenzer Testcases Testsceamrios ••• Exit Full S

Date	3-Nov-22
Team ID	PNT2022TMIDxxxxx
Project Name	Project - xxx
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	MIT App Inventor https://appinventor.mit.edu	Login popup should display		Fail	Steps not Clear to follow		Bug-1234	
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter Smart App 2.Verify login/Singup popup with below UI elements: a.Username text box b.password text box c.Submit button d.New customer? Create account link e.Last password? Recovery password	MIT App Inventor https://appinventor.mit.edu	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass				
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter MIT App Inventor URL (https://appinventor.mit.edu/) Smart app and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box	Username: IBM password: IBM	User should navigate to user account homepage	Working as Expected	Pass				
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL MIT App Inventor https://appinventor.mit.edu and smart app click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box	Username: chalam@gmail password: Testing123	Application should show 'Incorrect email or password' validation message.	Working as Expected	Pass				

8.2 User Acceptance Testing

1. 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).

Increasing control over production leads to **better cost management and waste reduction**. The ability to trace anomalies in crop growth or livestock health, for instance, helps eliminate the risk of losing yields. Additionally, automation boosts efficiency. Smart farming **reduces the ecological footprint of farming**. Minimized or site-specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases.

2. 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	8	3	2	2	16
Duplicate	1	0	2	0	3
External	2	3	0	1	6
Fixed	9	2	3	17	31
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	1	4	1	1	7
Totals	21	12	9	22	66

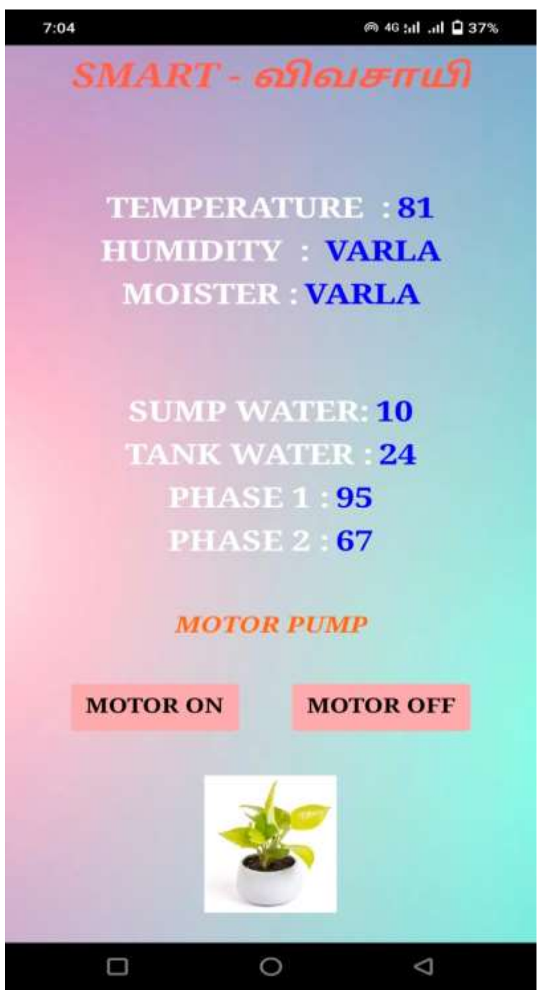
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	1	0	0	1

9.Result


SMART_FORMER
TEMPERATURE
TEMP
MOTOR
MOTOR OFF
MOTOR ON
swat
swat
HUMIDITY
HUMIDITY
MOISTER
MOISTER
twat
twat
pone
pone
ptwo
ptwo


7:04
37%
SMART - விவசாயி
TEMPERATURE : 81
HUMIDITY : VARLA
MOISTER : VARLA
SUMP WATER: 10
TANK WATER : 24
PHASE 1 : 95
PHASE 2 : 67
MOTOR PUMP
MOTOR ON
MOTOR OFF

10. Advantages & Disadvantages

Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labor cost.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of WebApp.

11. Conclusion

An IoT-based SMART FARMING SYSTEM for live monitoring of temperature,humidity, soil moisture, sump water, tank water, single phase, three phase is proposed using NodeMCU and cloud computing. The system has high efficiency and accuracy in acquiring live temperature and soil moisture data. The IoT-based smart farming system proposed in this report constantly assists farmers by providing accurate live feeds of ambient temperature and soil moisture for over 99 curated results, thus enabling farmers to increase their agricultural yields and help manage food production efficiently.

12. Future Scope

By collecting data from Sensor with IoT devices, we can learn about the “real state” of Crops. In future, IoT system in agriculture enables predictive analytics and helps you make better harvest decisions. It is important to use the latest information and communication technology to manage the family in order to improve the quantity and quality of products while optimizing the human labor force. In between Technologies available for today's glory: Soil, water, light, humidity and temperature control. Small Agricultural Products are designed to support field monitoring through the automation of automation systems using Sensors. As a result, Fame and associated volumes can easily monitor field conditions from anywhere.

13. Appendix

Links:

IBM cloud reference: https://cloud.ibm.com/

Github link : https://github.com/IBM-EPBL/IBM-Project-43512-1660717488

-IOT Watson simulator : https://wokwi.com/projects/347659185871127124

Node-Red : https://node-red-xvnds-2022-11-09.au-syd.mybluemix.net/red/#flow/d47a8f54bdef06ee