R.M.K. ENGINEERING COLLEGE

(An Autonomous Institution)

R.S.M. Nagar, Kavaraipettai-601 206

NOVEMBER 2022

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

TITLE OF THE PROJECT

SMART FARMER- IOT ENABLED SMART FARMING APPLICATION

PROJECT REPORT

TEAM ID : PNT2022TMID16026

Submitted by

HARSHINI T	(111719106053)
JANANI M	(111719106056)
JANANI M (9-9-2000)	(111719106057)
JAYANTHI A S	(111719106059)
JAYAVARTHNI R D	(111719106060)

CONTEXT

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

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, 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.

1.2 Purpose

The purpose of this project is to automate the irrigation process. Sensors can be used to monitor different parameters in the field. Different parameters like soil moisture, temperature and humidity can be sensed using sensors. Making arrangements to monitor these parameters through web or mobile applications. The decision for watering crops can be taken by the farmers based on the sensed parameters. The motors operating for irrigation can be controlled through the mobile application. The application can be used even if the farmer is far away from the field.

2. LITERATURE SURVEY

2.1 Existing problem

This system uses Raspberry pi as its processor and is connected to ADC with a soil moisture sensor producing output. Here, Ubidots cloud is connected to Raspberry pi through Wi-Fi connectivity. The data collected from sensors is transmitted to the cloud and gets stored in Ubidots server through Raspberry pi. For irrigation water will be turned ON when a specific condition is satisfied (i.e) When soil moisture value is below 14,000 the soil should continue to dry. The dangerous state is when the surface becomes too dry where soil moisture is below 18,000.

Soil moisture has a specific value which reacts with water after adding every 25ml of water, soil moisture value increases. The ADC value of soil moisture will be obtained by using ADS1015/ADS1115 library When the soil moisture value obtained will be calculated through the map function to obtain the appropriate percentage moisture content. Then the water pump will be active if the current time is at 7AM &6PM at the same time soil moisture level is below the desired level.

Disadvantages: Not able to run as a windows operating system and Impractical as a desktop computer

2.2 References

TITLE: AUTOMATED IRRIGATION SYSTEM - IOT BASED APPROACH

ABSTRACT OF THE PAPER 1 : Agriculture is a major source of earning of Indians and agriculture has made a big impact on India's economy. The development of crops for a better yield and quality delivery is exceptionally required. So suitable conditions and suitable

moisture in beds of crop can play a major role for production. Mostly irrigation is done by traditional methods of stream flows from one end to other. Such supply may leave varied moisture levels in field.The administration of the water system can be enhanced utilizing programmed watering framework This paper proposes a programmed water system with framework for the terrains which will reduce manual labour and optimizing water usage increasing productivity of crops. For formulating the setup, Arduino kit is used with moisture sensor with Wi-Fi module. Our experimental setup is connected with a cloud framework and data acquisition is done. Then data is analysed by cloud services and appropriate recommendations are given.

PUBLISHED IN:
https://ieeexplore.ieee.org/xpl/conhome/8502671/proceeding

AUTHORS: Dweepayan Mishra ,Arzeena Khan Rajeev Tiwari , Shuchi Upadhay.

TITLE: INTELLIGENT IRRIGATION SYSTEM - AN IOT BASED APPROACH

ABSTRACT OF THE PAPER 2: The Internet of Things (IOT) has been denoted as a new wave of information and communication technology (ICT) advancements. The IOT is a multidisciplinary concept that encompasses a wide range of several technologies, application domains, device capabilities, and operational strategies, etc. The ongoing IOT research activities are directed towards the definition and design of standards and open architectures which still have the issues requiring a global consensus before the final deployment. This paper gives an overview about IOT technologies and applications related to agriculture with comparison of other survey papers and proposes a novel irrigation management system. Our main objective of this work is for Farming where various new technologies yield higher growth of the crops and their water supply. Automated control features with the latest electronic technology using microcontroller which turns the pumping motor ON and OFF on

detecting the dampness content of the earth and GSM phone line is proposed after measuring the temperature, humidity, and soil moisture.

PUBLISHED IN:
https://ieeexplore.ieee.org/xpl/conhome/8053492/proceeding

AUTHORS: M. Newlin Rajkumar, S. Abinaya ,V. Venkatesa Kumar

TITLE: SOLAR POWERED SENSOR BASED IRRIGATION SYSTEM

ABSTRACT OF THE PAPER 3: This paper throws light on the development procedure of an embedded system for solar based Off-Grid irrigation systems. Solar power is absolutely perfect for use with irrigation systems. Using Solar panels, the sun's energy will be converted to electrical power and saved into batteries.When the sun is rising and shining, the solar panel will absorb the energy of the sun and the energy will stay in the battery. Light Detecting Resistors (LDR's) are placed on the solar panel which helps in tracking maximum intensity of sunlight. For generation of maximum energy, it is important to maintain solar panels face always perpendicular to the sun. This tracking movement of the panel is achieved by mounting the solar panel on the stepped motor.This stepped motor rotates the mounted panel as per signal received from the programmed microcontroller. The microcontroller used in this project is from the AVR family. Soil moisture sensor is placed inside soil to sense the moisture conditions of the soil. Based on moisture sensor values, the water pump is switched on and off automatically. When moisture level of the soil reaches too low, the soil moisture sensor is sending the signal to the microcontroller to start the pump by using stored solar energy.Same time, using GSM technique microcontroller is sending a message on farmers mobile about pump status. The microcontroller completes the above job as it receives signals from the soil moisture sensors, and these signals function as per program stored in the ROM of the microcontroller. The LDR's values, soil

moisture values, condition of the pump i.e., on/off are displayed on a 16x2 LCD which is interfaced to the microcontroller.

PUBLISHED IN:

https://www.irjet.net/archives/V3/i2/IRJET-V3I279.pdf

AUTHORS: Kavita Bhole , Dimple Chaudhari 17 4.4

TITLE: SENSOR BASED AUTOMATED IRRIGATION SYSTEM WITH IOT

ABSTRACT OF THE PAPER 4: India's population has reached beyond 1.2 billion and the population rate is increasing day by day then after 25-30 years there will be serious problems with food, so the development of agriculture is necessary. Today, the farmers are suffering from the lack of rains and scarcity of water. The main objective of this paper is to provide an automatic irrigation system thereby saving time, money & power of the farmer.The traditional farm-land irrigation techniques require manual intervention. With the automated technology of irrigation the human intervention can be minimized. Whenever there is a change in temperature and humidity of the surroundings these sensors senses the change in temperature and humidity and gives an interrupt signal to the microcontroller.

PUBLISHED IN:
https://ijcsit.com/docs/Volume%206/vol6issue06/ijcsit20150606104.pdf

AUTHORS: Karan Kansara ,Vishal Zaveri ,Shreyans Shah 18 4.5

TITLE: WIRELESS MONITORING OF SOIL MOISTURE, TEMPERATURE AND HUMIDITY USING ZIGBEE IN AGRICULTURE.

ABSTRACT OF THE PAPER 5: The main objective of the present paper is to develop a smart wireless sensor network (WSN) for an agricultural environment. Monitoring the agricultural environment for various factors such as soil moisture, temperature and humidity along with other factors can be of significance. A traditional approach to measure these factors in an agricultural environment meant individuals manually taking measurements and checking them at various times. This paper investigates a remote monitoring system using Zigbee. These nodes send data wirelessly to a central server, which collects the data, stores it and will allow it to be analyzed then displayed as needed and can also be sent to the client mobile.

PUBLISHED IN:

http://www.ijettjournal.org/archive/ijett-v11p296

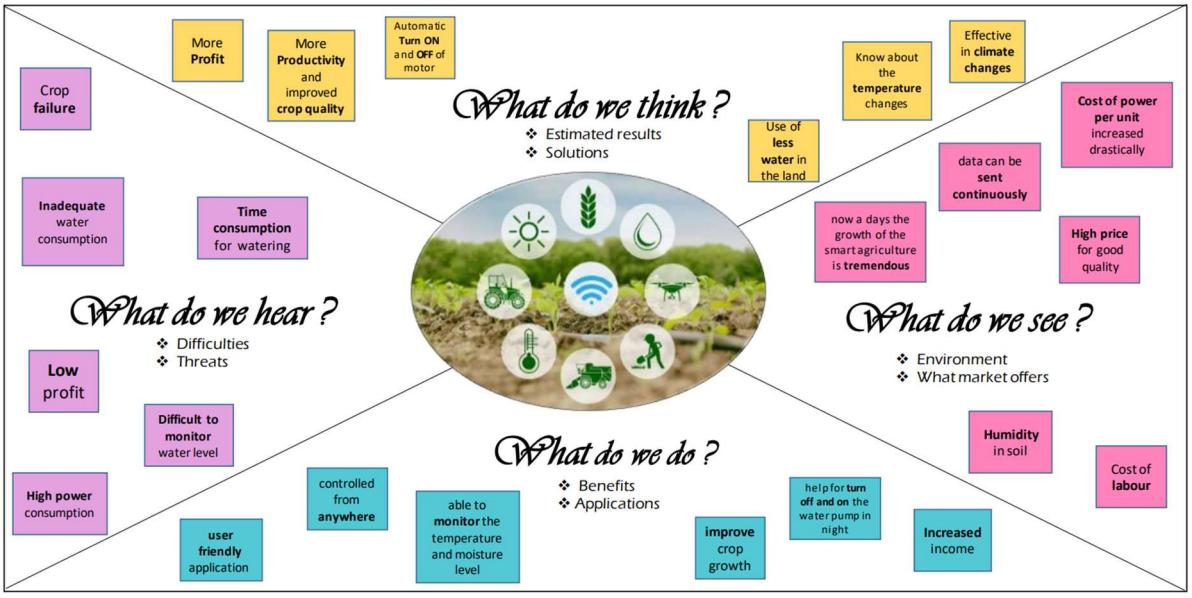
AUTHORS: C.H. Chavan , P.V. Karande

2.3 Problem Statement Definition

Ideally, each field should get just the right amount of water at just the right time. Under-watering causes crop stress and yield reduction. Overwatering can also cause yield reduction and consumes more water and fuel than necessary.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Problem statement

Ideally, each field should get just the right amount of water at just the right time. Under-watering causes crop stress and yield reduction. Overwatering can also cause yield reduction and consumes more water and fuel than necessary. Team ID PNT2022TMID16026 Project Name Smart Farmer - IoT Enabled Smart Farming Application

Ideas or solutions

- To automate the irrigation process
- Sensors can be used to monitor different parameters in the field

- Different parameters like soil moisture, temperature and humidity can be sensed using sensors
- Making arrangements to monitor these parameters through web or mobile applications
- The decision for watering crops can be taken by the farmers based on the sensed parameters
- The motors operating for irrigation can be controlled through the mobile application
- The application can be used even if the farmer is far away from the field

Abstract

To automate the farming process by monitoring various parameters and controlling it through web or mobile application for the ease of farmers

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Ideally, each field should get just the right amount of water at just the right time. Under-watering causes crop stress and yield reduction. Overwatering can also cause yield reduction and consumes more water and fuel than necessary.
2.	Idea / Solution description	In our proposed system, we are implementing an Automatic Smart Irrigation System Using IoT, which will be helpful for watering plants so that water can be saved efficiently by monitoring the soil moisture. Monitoring is done through our smart Phone.

3.	Novelty / Uniqueness	Using budget sensors which helps to monitor temperature, moisture, humidity effectively and efficiently.
4.	Social Impact / Customer Satisfaction	It is the best method for smart irrigation system and with this we can save time and manpower and thus producing better yield.
5.	Business Model (Revenue Model)	Since the productivity of crop increases the satisfaction of customers also increases and eventually the need for the application increases, which in turn increases the revenue.
6.	Scalability of the Solution	The application is scalable as we can improve or increase the performance when the problem arises.

3.4 Problem Solution fit

1. CUSTOMER SEGMENT(S) CS Who is your customer? Farmers are our customer	2. JOBS-TO-BE-DONE / J&P PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. To make their job easier • Soil moisture	3.TRIGGERS By seeing growth of automation • Crop quality • Cost of power and water • Quantity of crop production 4. EMOTIONS: BEFORE / AFTER

	• Temperature • Humidity • Automatic irrigation	Before: Worried about irrigation and loss. After: Increased profit and low water and power consumption
5. AVAILABLE SOLUTIONS They tried to control the usage of water at the field but they couldn't do it efficiently all the time.	6. CUSTOMER CONSTRAINTS Network availability, knowledge about using the application, budget are several constraints	7. . BEHAVIOUR The customer will reach us to know how to analyze the soil and sow the seeds accordingly to get maximum yield.
8.CHANNELS OF BEHAVIOUR Online: Managing irrigation through our application. Offline: Making sure that the setup required for the application is installed properly	9. PROBLEM ROOT CAUSE No knowledge about the soil moisture, humidity which is the major factor which determines the crop growth.	10. YOUR SOLUTION Our aim is to provide maximum crop yield with effective use water and power.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Verifying by Email Verifying by OTP
FR-3	Temperature measurement	DHT11 Temperature and Humidity Sensor is used for measuring temperature since **high temperature can damage the roots** resulting in substantial reduction in shoot growth.
FR-4	Humidity measurement	DHT11 Temperature and Humidity Sensor is used for measuring relative humidity which is **important to make photosynthesis possible.**
FR-5	Soil moisture measurement	YL69 Soil moisture sensor is used. Soil **moisture is the critical parameter in agriculture**. If there is a shortage or over abundance of water, plants may die.
FR-6	Irrigation of soil if needed	If there is shortage in soil moisture then motor is turned on for irrigation to **improve crop growth and quality.**

4.2 Non-Functional requirements

Non-functional Requirements:

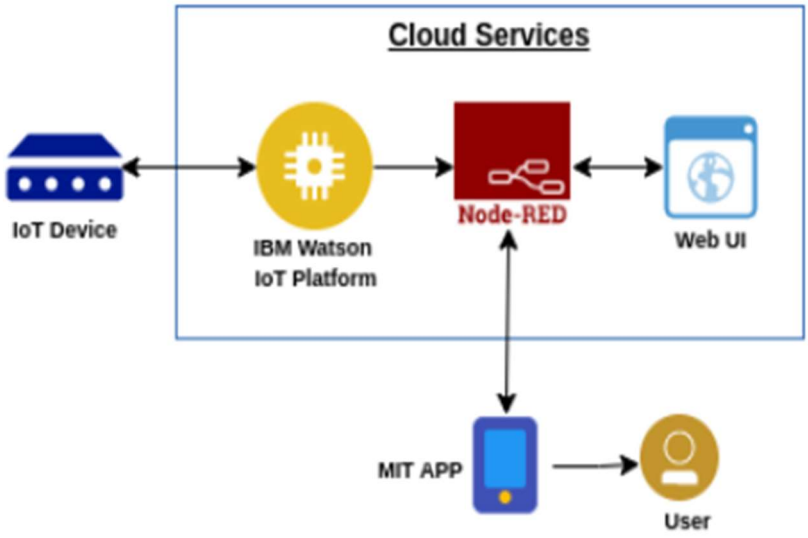
Following are the non-functional requirements of the proposed solution.

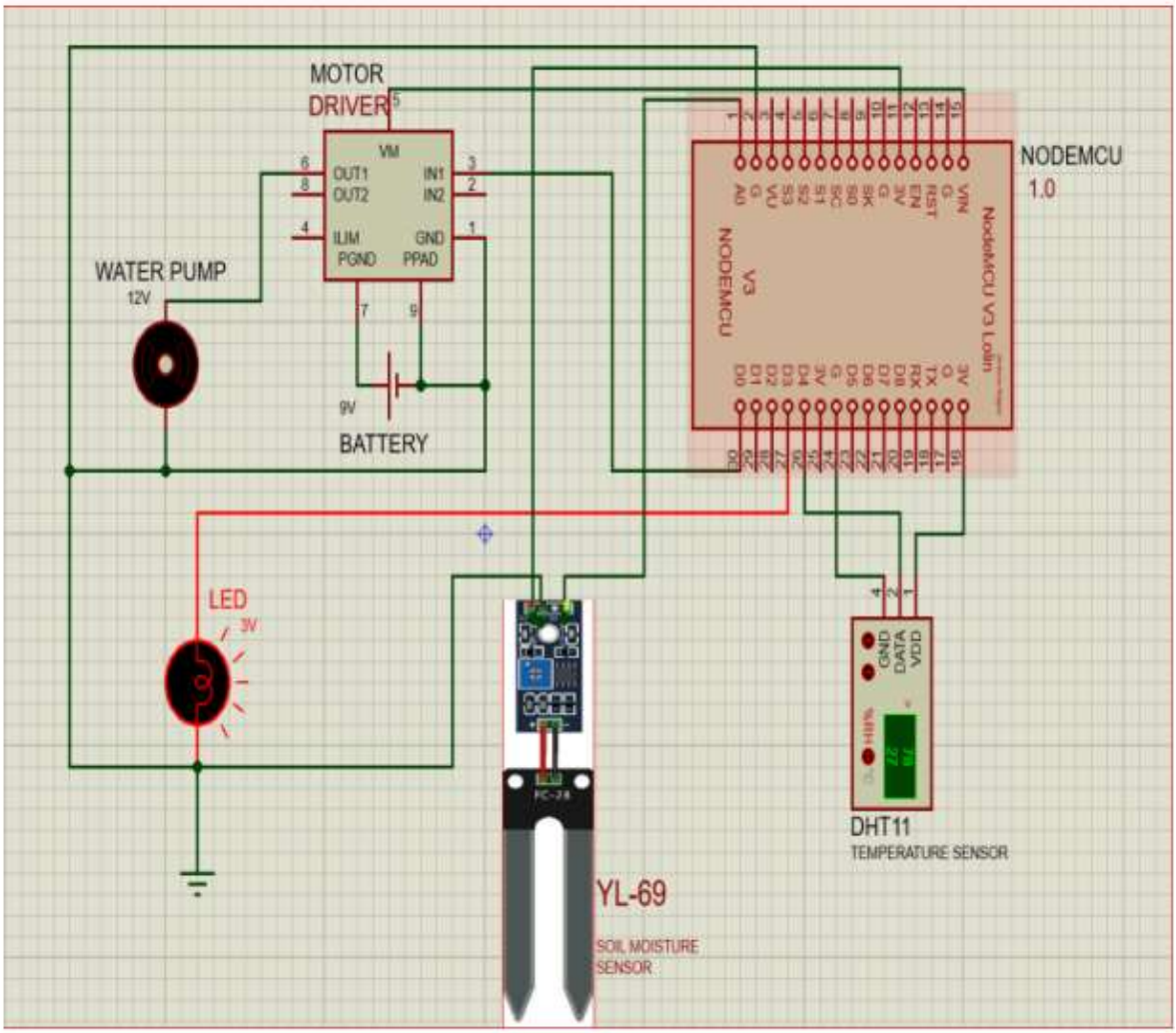
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability is a quality attribute that assesses how easy user interfaces are to use. It is a measure that the user feels easy to access the project.
NFR-2	**Security**	Ensure that all the data within the system will be protected against theft, malware attacks or unauthorised access.
NFR-3	**Reliability**	The degree to which the result of a measurement is accurate and longer Life Span.
NFR-4	**Performance**	It should be effective to monitor plant growth.
NFR-5	**Availability**	It must be available for 24/7 and should be easy to alter the soil moisture at home.
NFR-6	**Scalability**	Scalability is the ability of a device to adapt to the changes in the environment and meet the changing needs in the future. The proposed work can be integrated with new components in future if needed.

5. PROJECT DESIGN

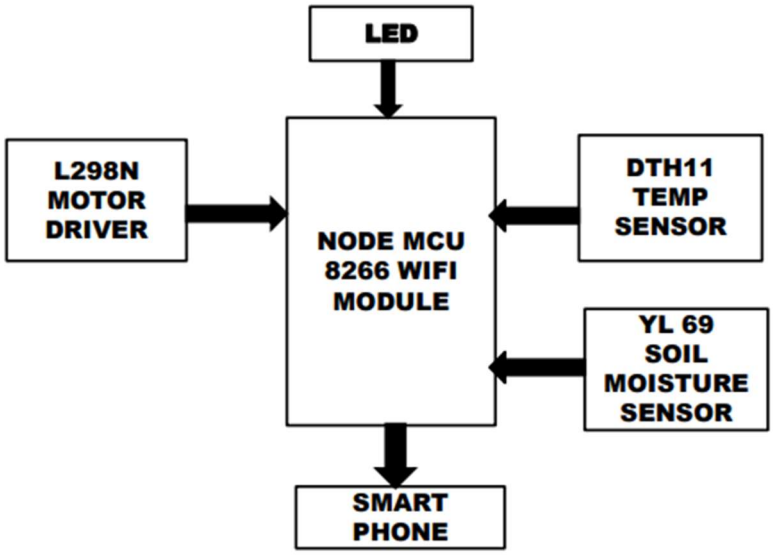
5.1 Data Flow Diagrams

Data Flow Diagrams: 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 & Technical Architecture



Technical Architecture:

1.The three soil parameter measurements are Temperature, Soil moisture, and humidity which are made using various sensors and recorded in the IBM cloud.
2. Using an Arduino UNO as a processing unit the data from the sensors and weather are processed. 3. The programming tool used to write the hardware, software and APIs is NODE-RED. The MQTT protocol is followed for the communication.
4. The user is given access to all the collected data through a smartphone application created with the aid of MIT App Inverter. Depending on the sensor results, the user might decide whether or not to irrigate the crop using an app. The motor switch can be controlled remotely by utilising the app.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	1	Can register for the application by entering my email, password, and confirming my password.	Can access my account / dashboard	High	Sprint-1
		2	Will receive confirmation email once I have registered for the application	Receive confirmation email & click confirm	High	Sprint-1
		3	As a user, I can register for the application through Facebook	Can register & access the dashboard with Facebook Login	Low	Sprint-2
		4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	5	As a user, I can log into the		High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
			application by entering email & password			
	Dashboard					
Customer (Web user)	Registration	1	Can register for the application by entering my email, password, and confirming my password.	Can access my account / dashboard	High	Sprint-1
		2	Will receive confirmation email once I have registered for the application	Receive confirmation email & click confirm	High	Sprint-1
		3	As a user, I can register for the application through Facebook	Can register & access the dashboard with Facebook Login	Low	Sprint-2
		4	As a user, I can register for the application through Gmail		Medium	Sprint-1
		5	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer Care Executive						
Administrator						

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 the Arduino and sensors with python code	2	High	Jayanthi Harshini
Sprint-2	Software	USN-2	Creation of device in the IBM Watson IoT platform, checking the workflow for IoT scenarios using Node-Red	1	High	Janani (6056) Jayavarthni
Sprint-3	MIT App Inventor	USN-3	To develop an application for the Smart Farmer - IoT Enabled Smart Farming Application project using MIT App Inventor	2	Medium	Janani (6057) Janani (6056)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Dashboard	USN-4	To design the Modules and test the application	2	Medium	Jayavarthni Jayanthi
Sprint-5	Web UI	USN-5	To make the user interact with the software.	1	High	Harshini Janani (6057)

6.2 Sprint Delivery

Sprint Delivery – 1

Introduction

Agriculture is the backbone of the Indian Economy"- said Mahatma Gandhi. The main aim of our project is to help farmers with a Web App to monitor Temperature, soil moisture, humidity and to control water motor remotely via internet without going to their field.

Problem Statement

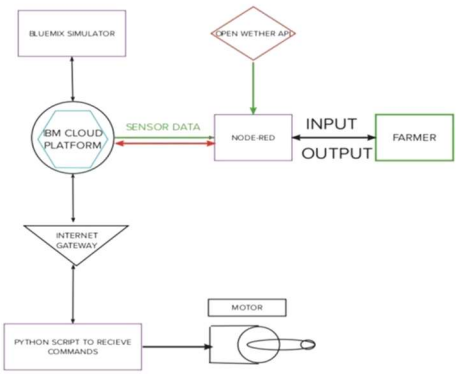
Farmers should be present in their field anytime irrespective of their health, climatic conditions even without considering their family time. They have to check the soil moisture, Temperature, Humidity before watering the crops and also ensure that the crops are well watered.

Proposed Solution

We aim to help the Farmers and provide easier working environment also accurate. We introduce IOT services to them which connect cloud services and internet to ensure that farmers can work remotely via internet. Also, He can monitor the field parameters and control the devices in farm.

Theoretical Analysis

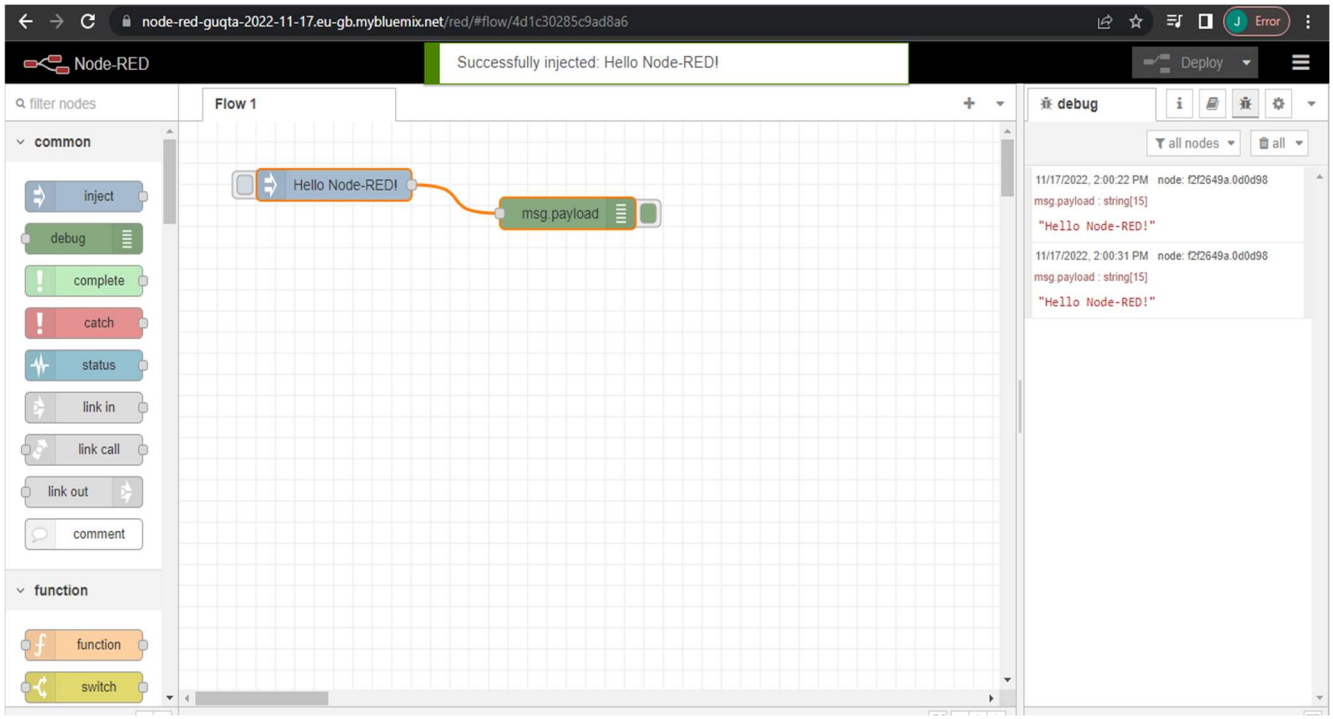
Block Diagram



Required Software Installation

Node-Red

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



First install npm/node.js

Open cmd prompt

Type => npm install node-red

To run the application :

Open cmd prompt

Type=>node-red

Then open http://localhost:1880/ in browser

Installation of IBM IoT and Dashboard nodes for Node-Red

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required 1. IBM IoT node 2. Dashboard node

IBM Watson IoT Platform

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage. IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices.

Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token

Create API key and store API key and token elsewhere

Python IDE

Install Python3 compiler Install any python IDE to execute python scripts

Arduino Code In Wokwi:

```
#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
#define LED 2

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

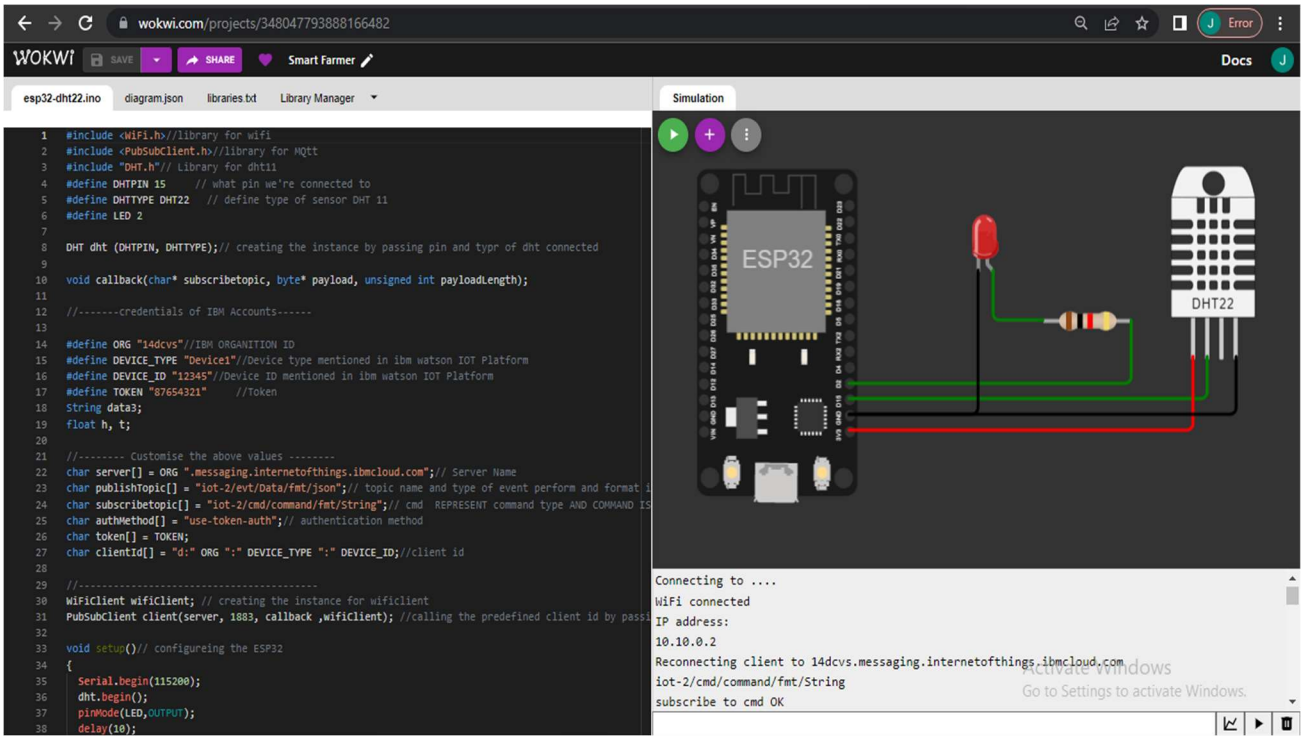
```
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);

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

#define ORG "14dcvs"//IBM ORGANITION ID
#define DEVICE_TYPE "Device1"//Device type mentioned in ibm
watson IOT Platform
#define DEVICE_ID "12345"//Device ID mentioned in ibm watson
IOT Platform
#define TOKEN "87654321"     //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();
  pinMode(LED,OUTPUT);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{

  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();
  }
}



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

void PublishData(float temp, float humid) {
  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 += "}";


  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);
digitalWrite(LED,HIGH);
  }
```

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



Sprint Delivery – 2

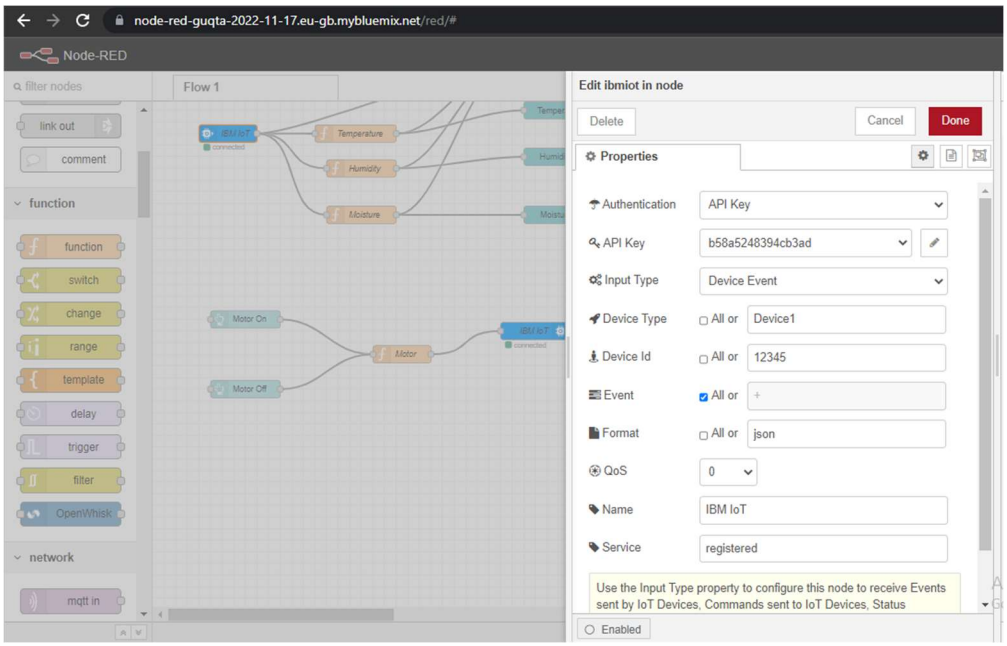
Building Project Connecting IOT Simulator to IBM Watson IOT Platform

- Open link provided in above section 4.3
- Give the credentials of your device in IBM Watson IOT Platform
- Click on connect My credentials given to simulator are:
- Org ID: 14dcvs
- Api key : a-14dcvs-xzoonjld1n
- Device type: Device1
- Authentication token: FSaB@(rp7jt2hUXduI
- Device ID: 12345
- Device Token: 87654321
- You can see the received data in graphs by creating cards in Boards tab

Configuration of Node-Red to collect IBM cloud data

The node IBM IOT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red

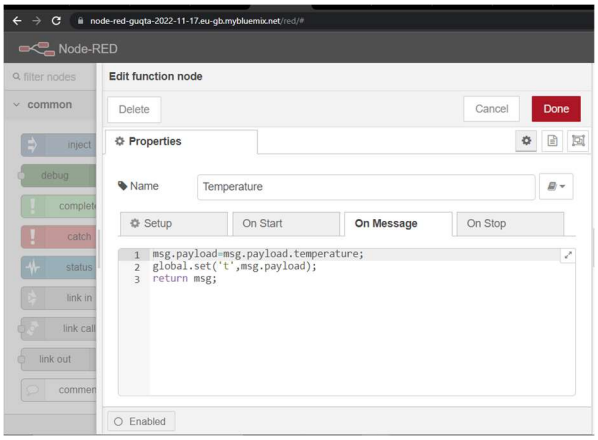
Once it is connected Node-Red receives data from the device, Display the data using debug node for verification. Connect function node and write the Java script code to get each reading separately.



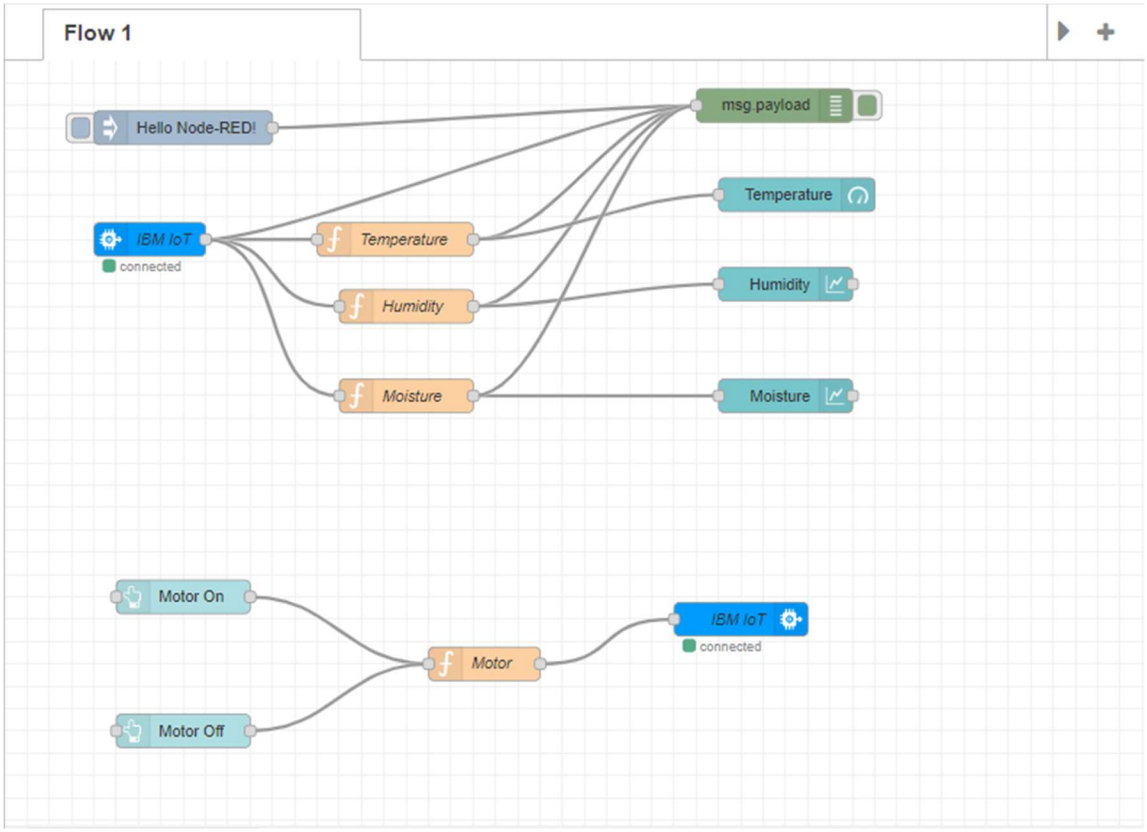
The Java script code for the function node is:

```
msg.payload = msg.payload.temperature;
return msg;
```

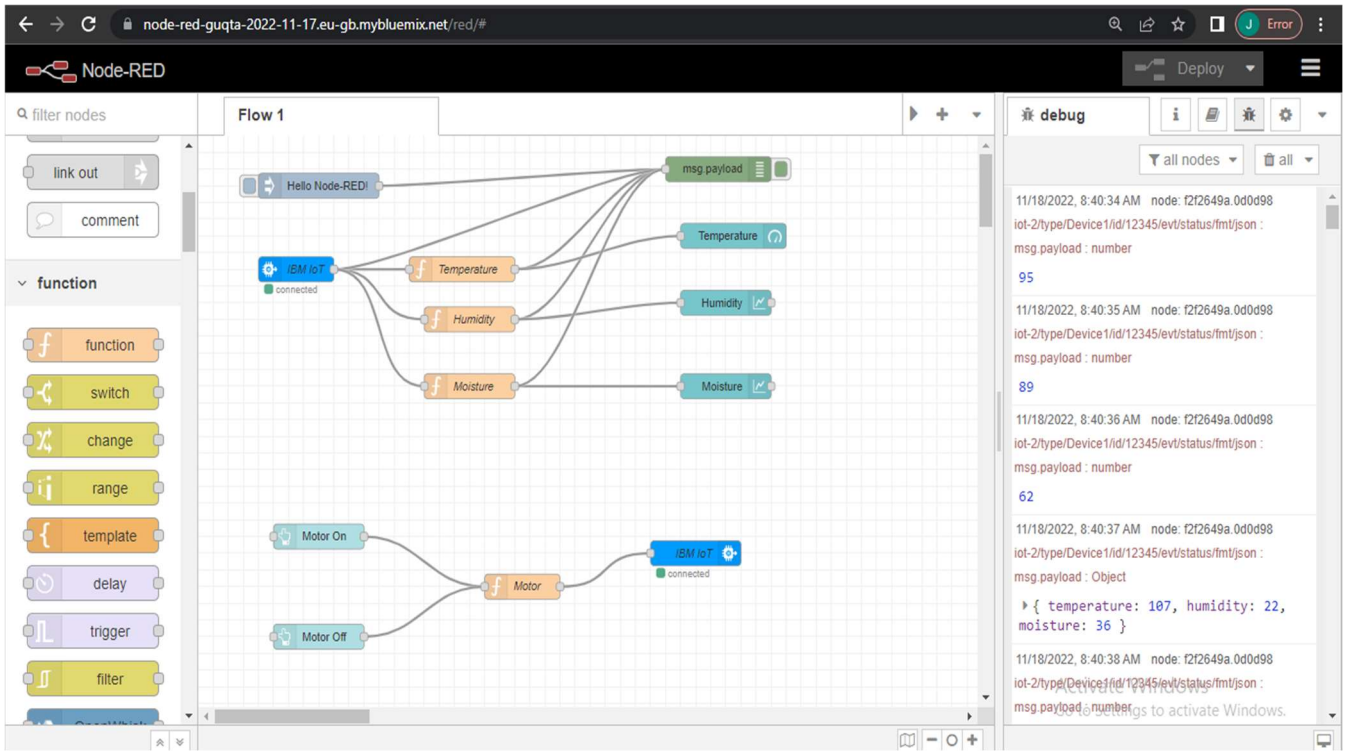
Finally connect Gauge nodes from dashboard to see the data in UI.



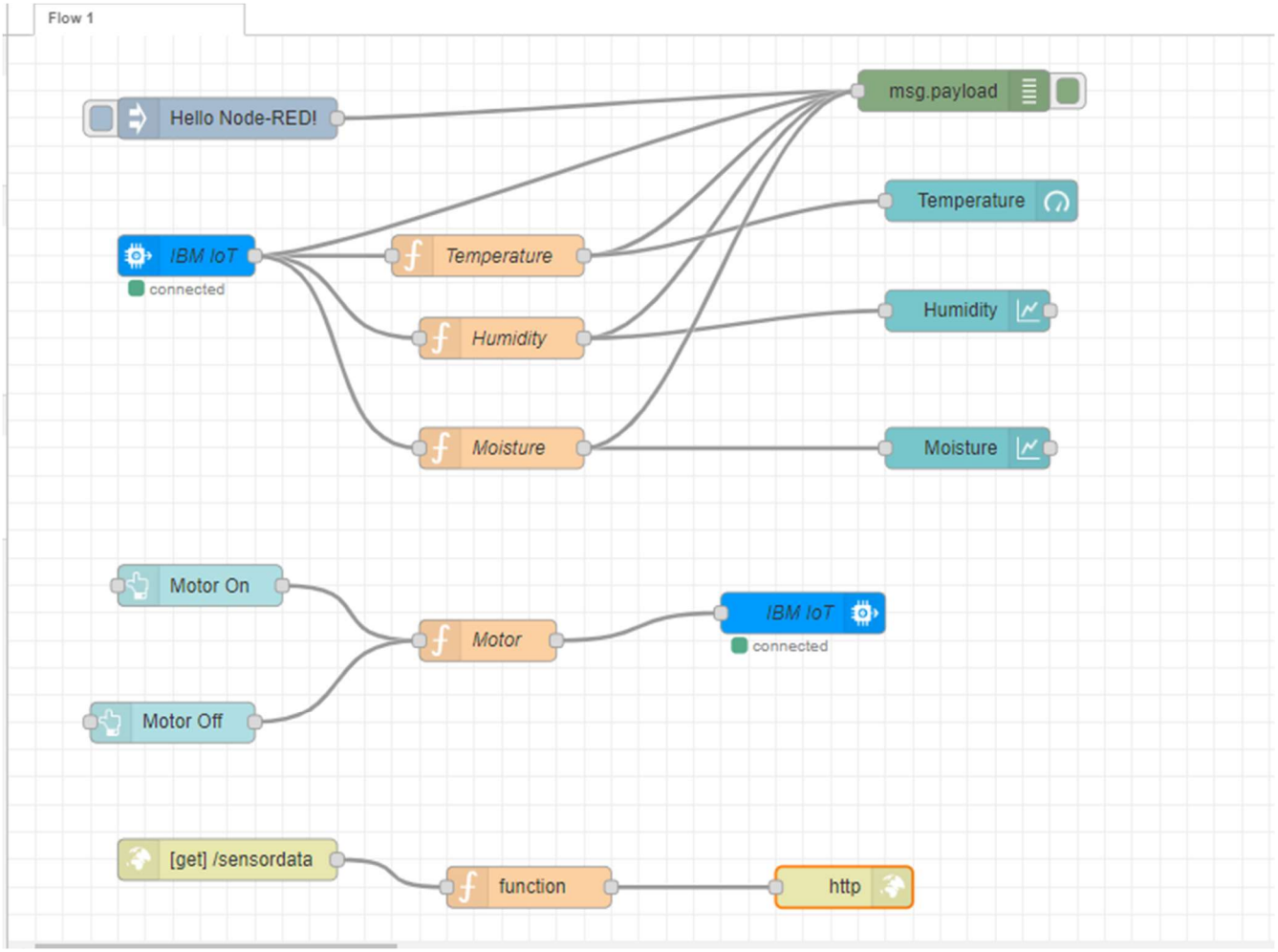
Nodes connected in following manner to get each reading separately.



Data received from the cloud in Node-Red console.



Data from Node-Red to MIT app Inventor



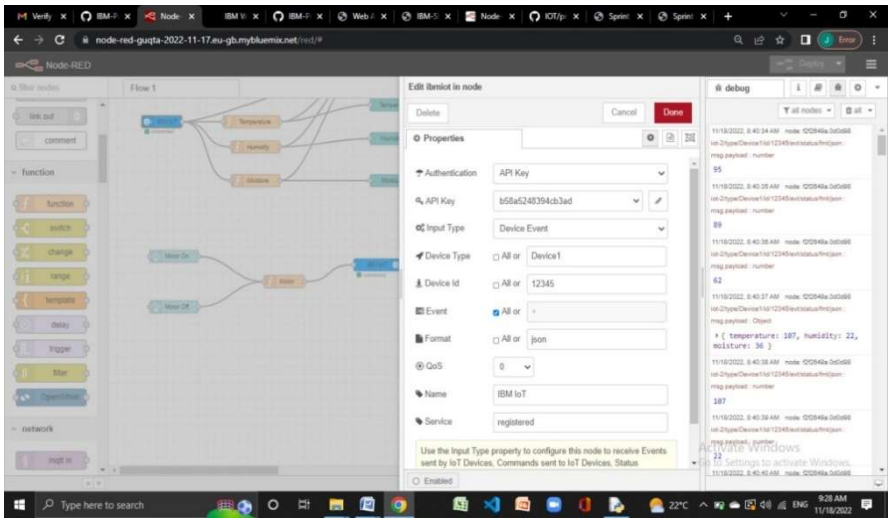
Sprint delivery – 3

Configuration of Node-Red to send commands to IBM cloud

We used IBM IOT in node to get IBM Watson to Node Red.IBM out node 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 Watson device.

Here we add two buttons in UI

1 -> for motor on
2 -> for motor off



PROGRAM:

Sensor data:

```
msg.payload={
    "temperature":global.get('t'),
    "humidity":global.get('h'),
    "moisture":global.get('m')
}
return msg;
```

Temp:

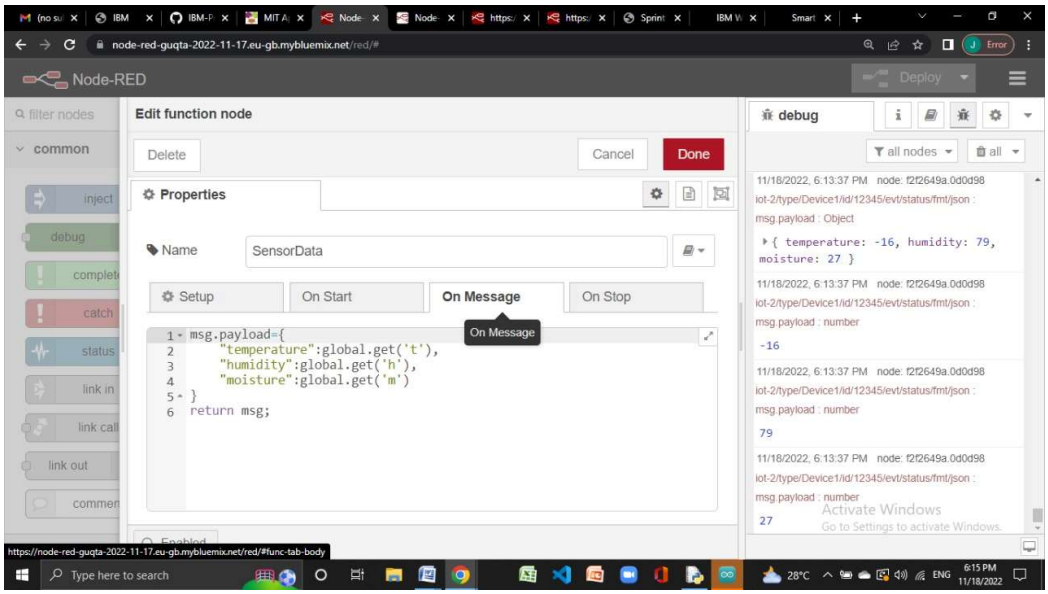
```
msg.payload=msg.payload.temperature
global.set('t',msg.payload);
return msg;
```

Motor:

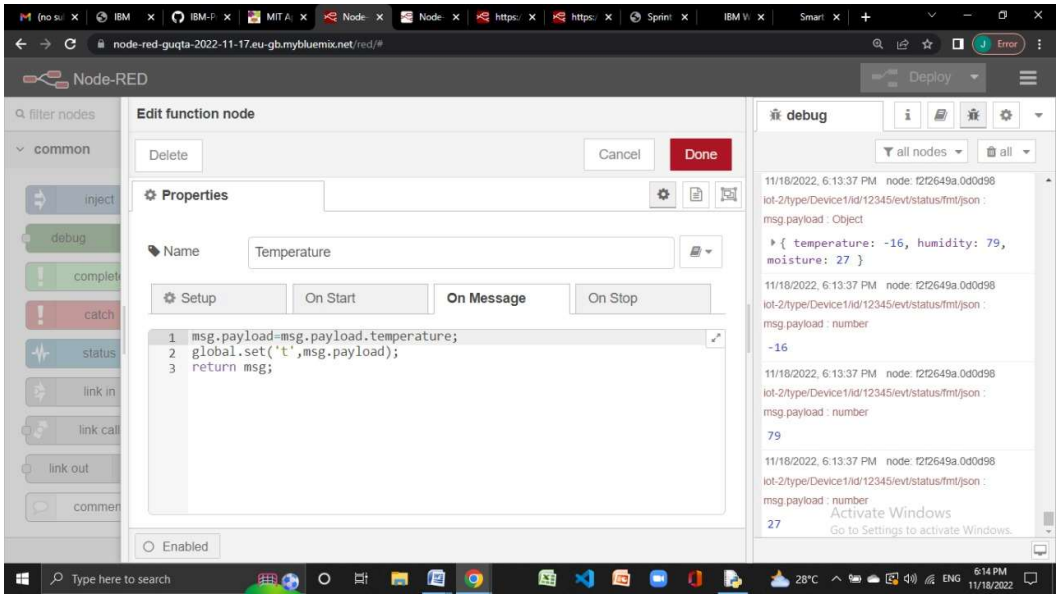
```
msg.payload={
   "command": msg.payload
}
return msg;
```

The data from NODE RED to MIT APP INVENTOR. The following data are:

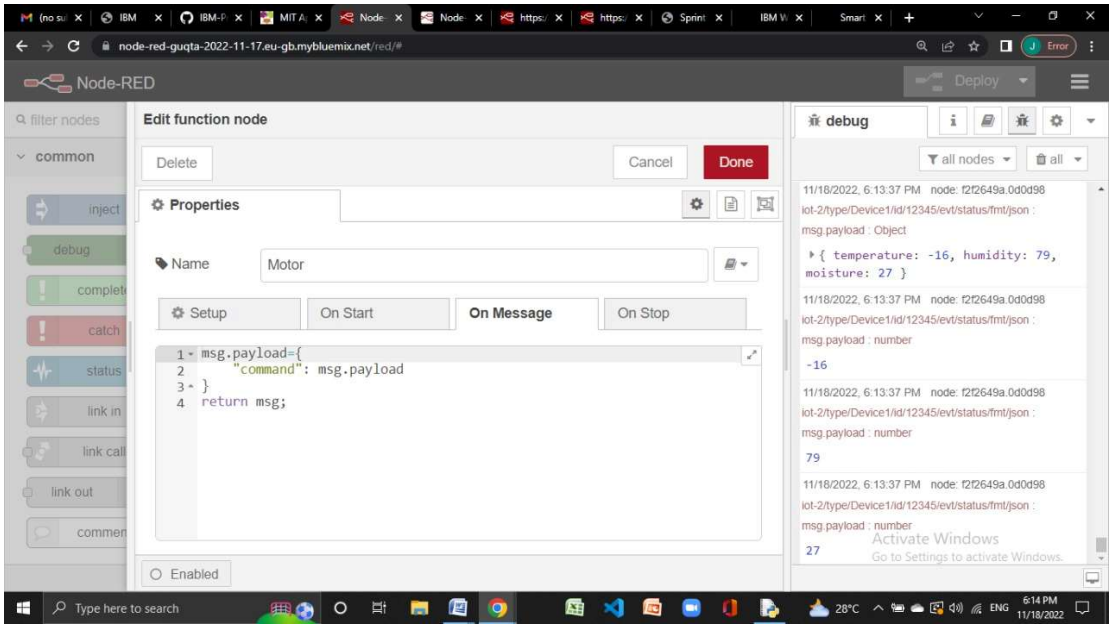
SENSOR DATA:

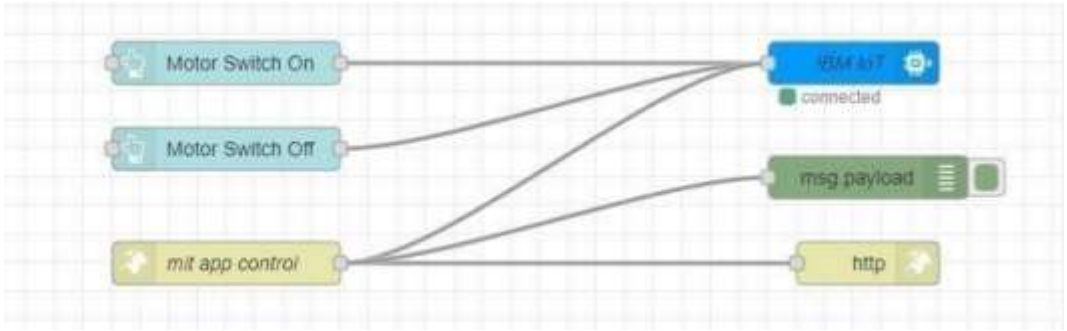


TEMPERATURE:



MOTOR:

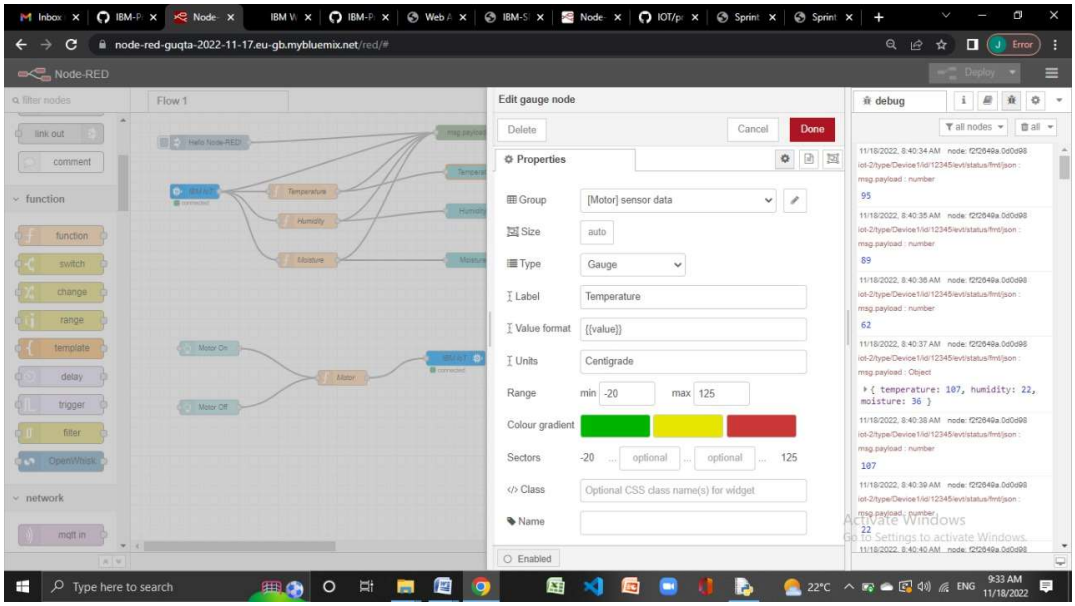




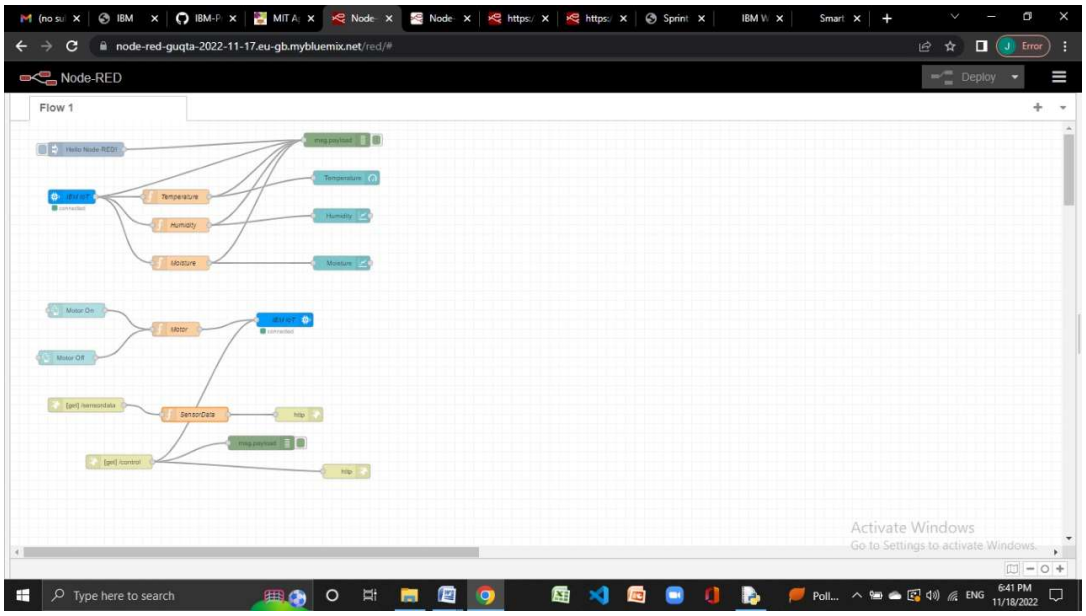
This is the program flow for sending commands to IBM cloud.

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.

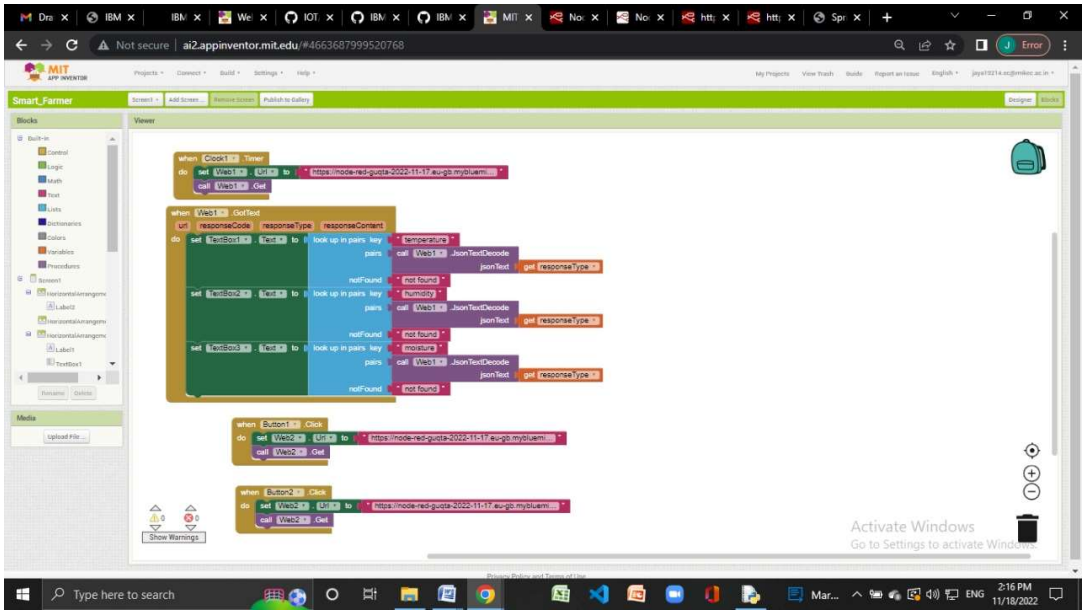


Complete Program Flow



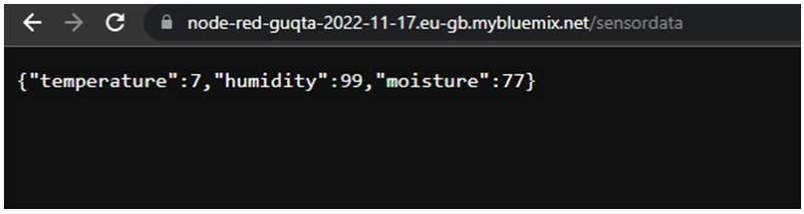
MOBILE APP WEB :

BLOCK DIAGRAM

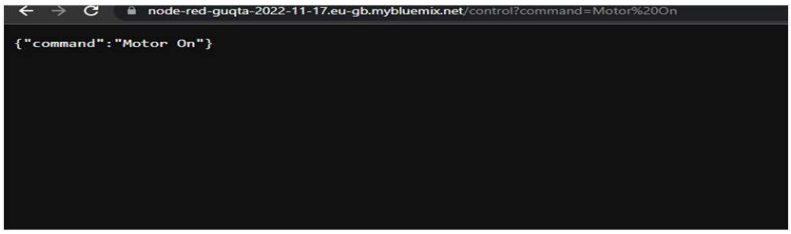


To get information from NODE RED to MIT APP. The following are used:

SENSOR DATA



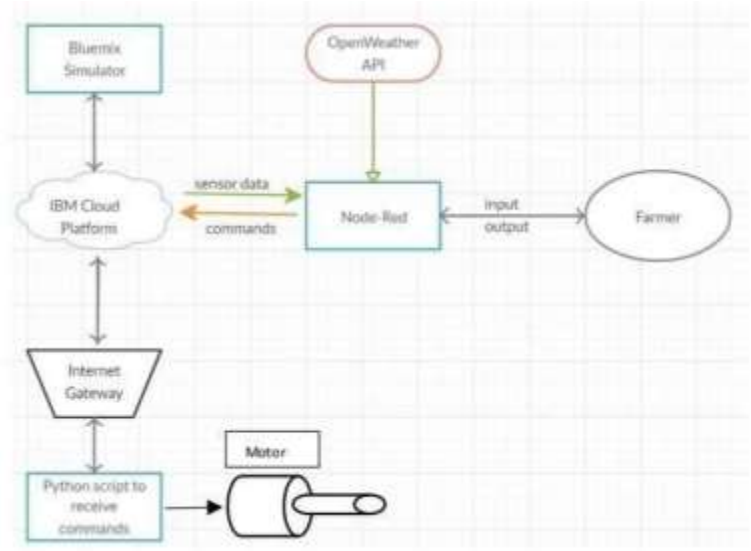
COMMAND:



Sprint delivery – 4

Receiving commands from IBM cloud using Python program

FLOWCHART



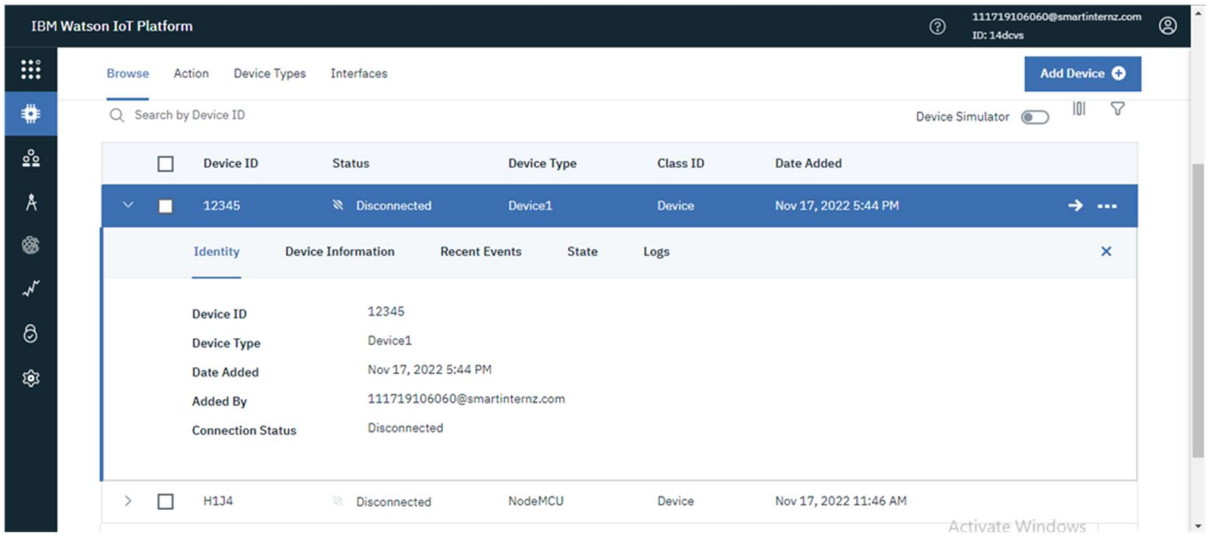
7. CODING & SOLUTIONING

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "14dcvs",
        "typeId": "Device1",
        "deviceId":"12345"
    },
    "auth": {
        "token": "87654321"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="Motor On"):
        print("****///Motors ARE ON///****")
    else:
        print("****///Motors ARE OFF///****")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    Mois=random.randint(20,120)
    myData={'temperature':temp, 'humidity':hum, 'moisture':Mois}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

8. TESTING

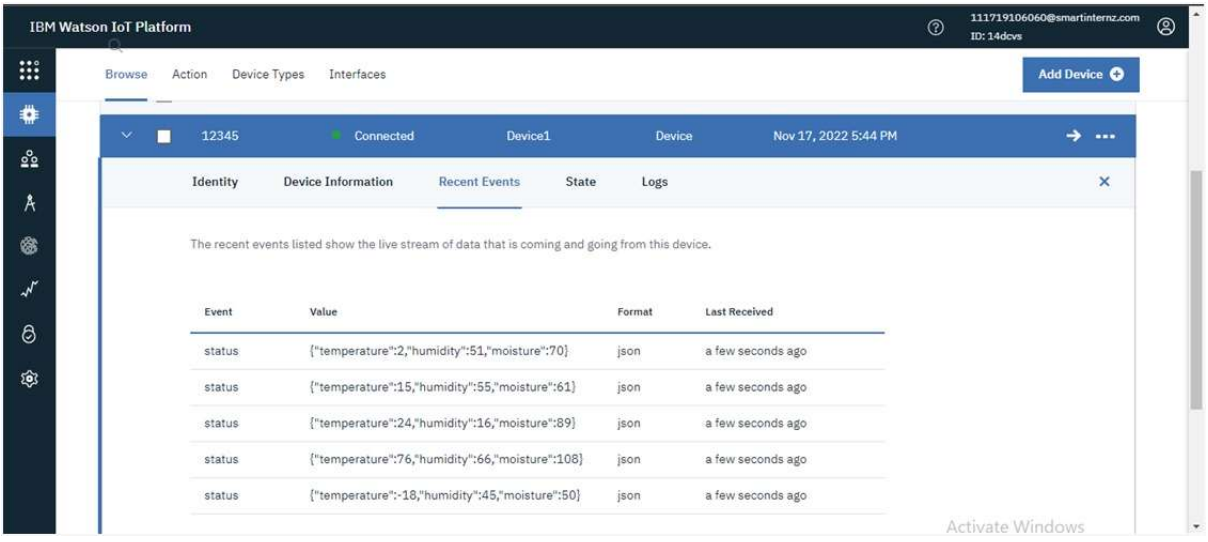
8.1 Test Cases

Device creation

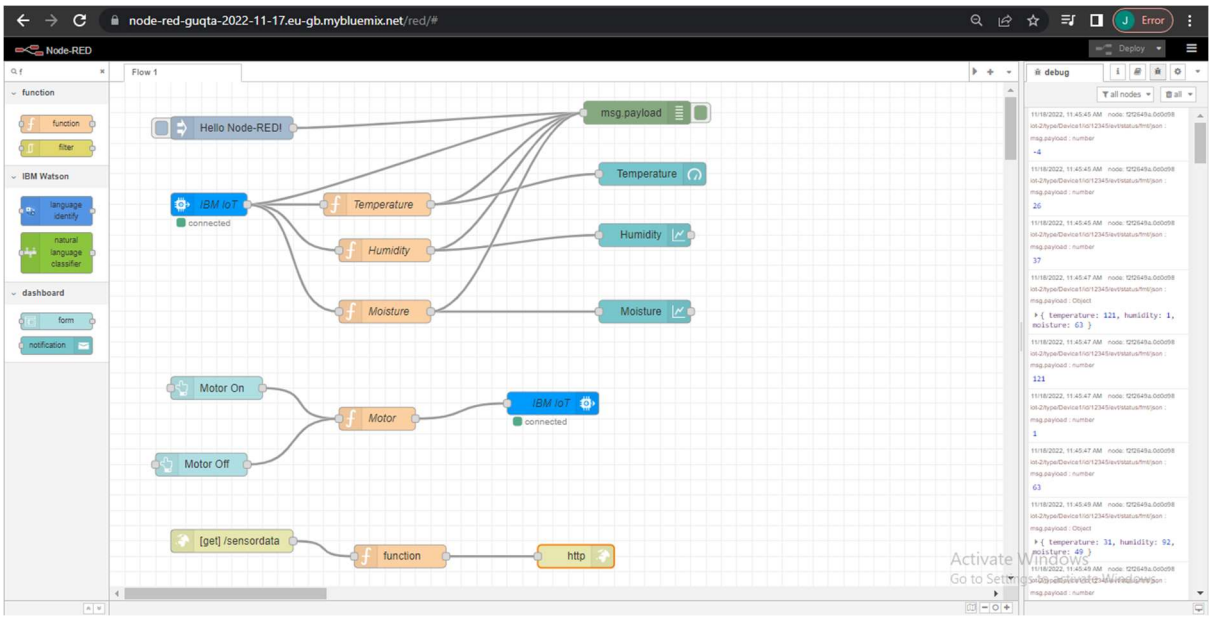


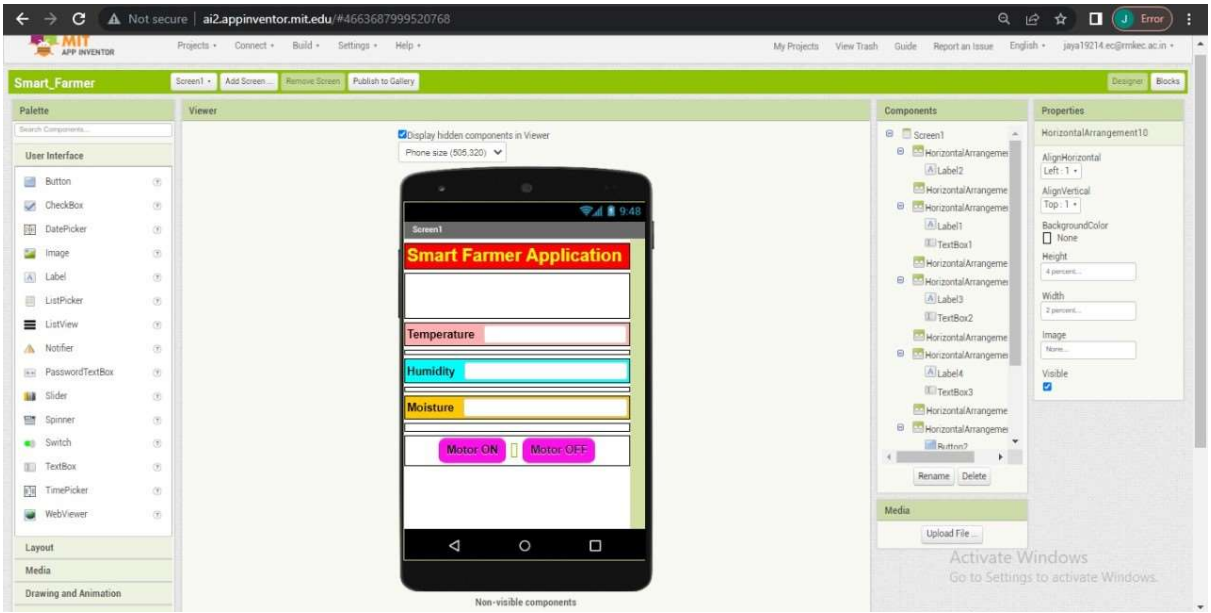
Python Script

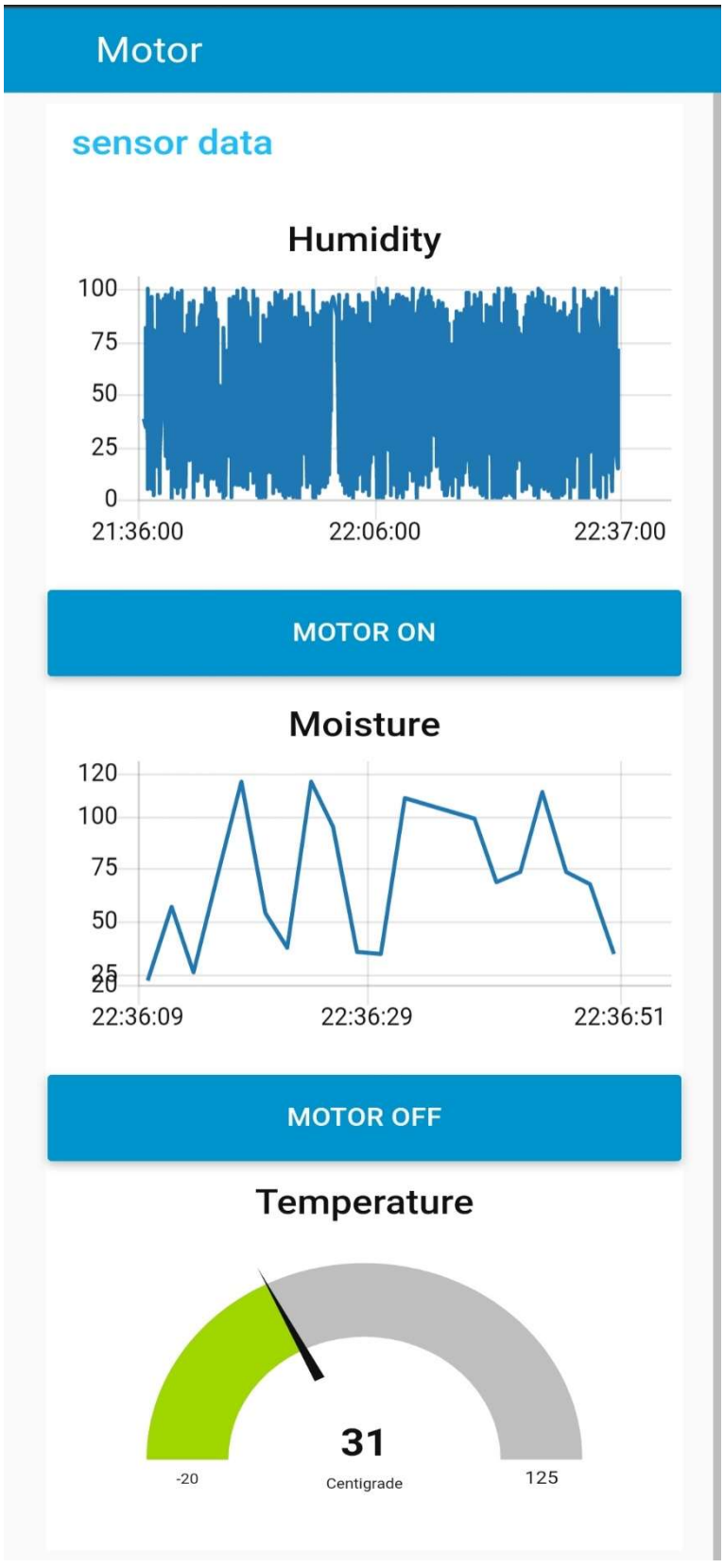

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:/Users/ANAND/Desktop/pythoncode.py ==========
2022-11-18 18:06:01,244 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:14dcvs:Device1:12345Published data Successfully: %s
{'temperature': 75, 'humidity': 58, 'moisture': 22}
Published data Successfully: %s {'temperature': 44, 'humidity': 33, 'moisture': 39}
Published data Successfully: %s {'temperature': 11, 'humidity': 62, 'moisture': 93}
Published data Successfully: %s {'temperature': 102, 'humidity': 10, 'moisture': 32}
Published data Successfully: %s {'temperature': 107, 'humidity': 51, 'moisture': 32}
Published data Successfully: %s {'temperature': -7, 'humidity': 58, 'moisture': 77}
Published data Successfully: %s {'temperature': 97, 'humidity': 90, 'moisture': 23}
Published data Successfully: %s {'temperature': 58, 'humidity': 100, 'moisture': 21}
Activate Windows
Go to Settings to activate Windows.
Ln: 10 Col: 0


IBM Watson IoT Platform
111719106060@smartinternz.com
ID: 14dcvs
Browse
Action
Device Types
Interfaces
Add Device
12345
Connected
Device1
Device
Nov 17, 2022 5:44 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
status
{"temperature":2,"humidity":51,"moisture":70}
json
a few seconds ago
status
{"temperature":15,"humidity":55,"moisture":61}
json
a few seconds ago
status
{"temperature":24,"humidity":16,"moisture":89}
json
a few seconds ago
status
{"temperature":76,"humidity":66,"moisture":108}
json
a few seconds ago
status
{"temperature":-18,"humidity":45,"moisture":50}
json
a few seconds ago
Activate Windows

8.2 User Acceptance Testing

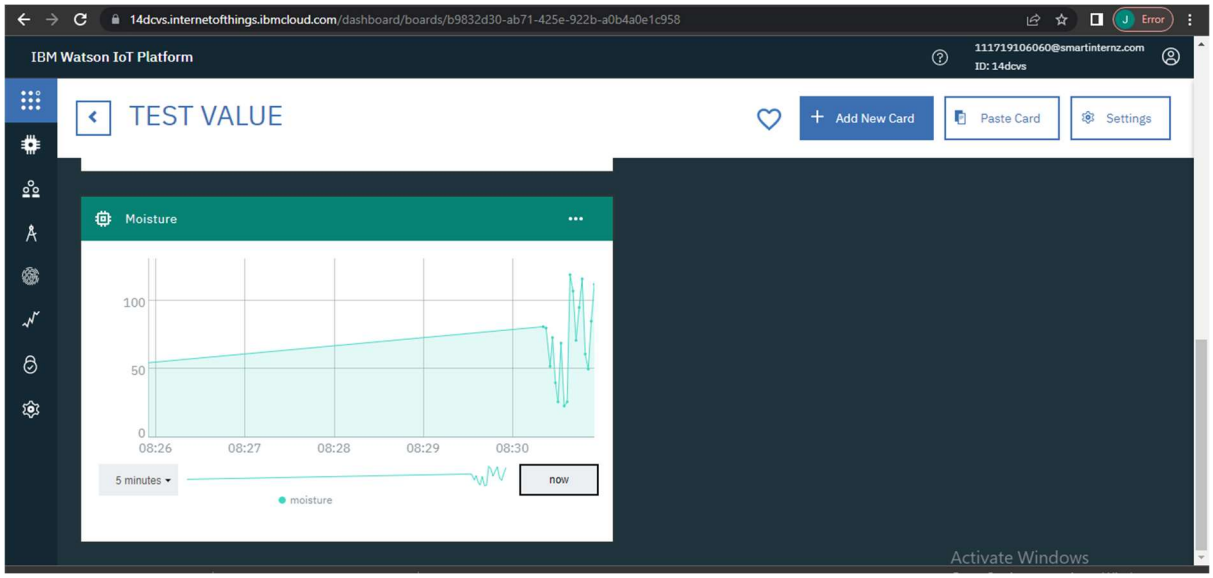


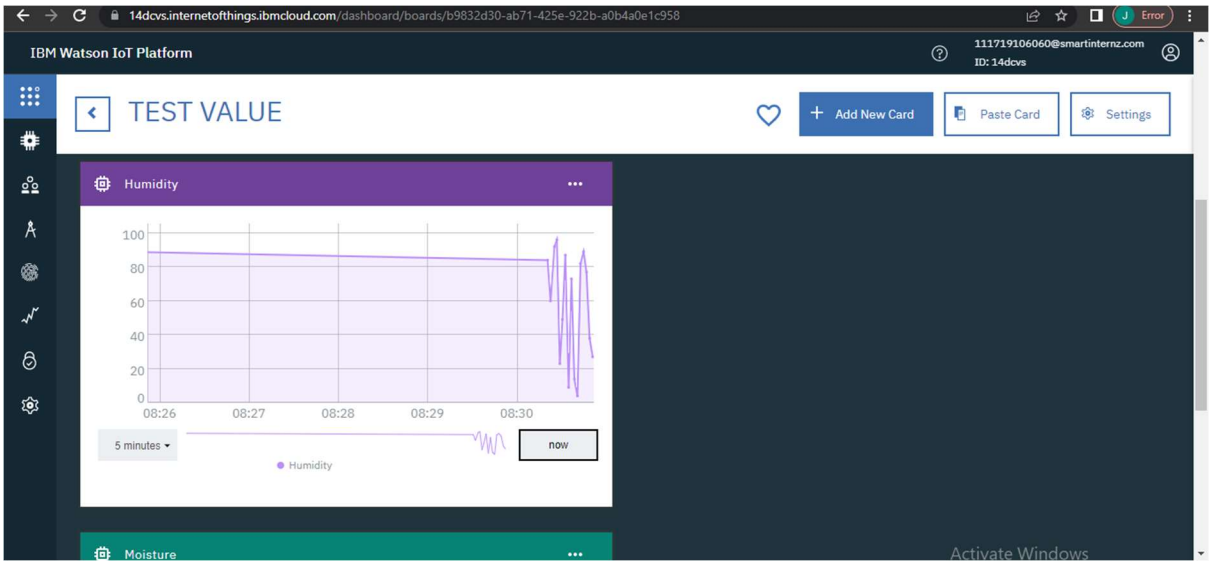


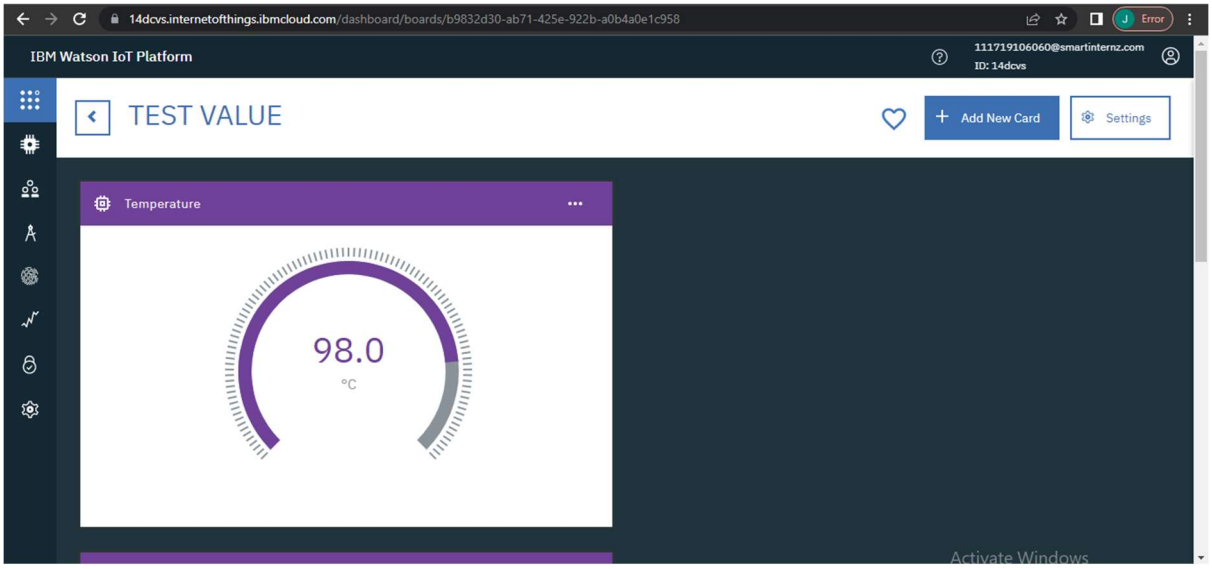

Motor
sensor data
Humidity
100
75
50
25
0
21:36:00
22:06:00
22:37:00
MOTOR ON
Moisture
120
100
75
50
25
20
22:36:09
22:36:29
22:36:51
MOTOR OFF
Temperature
31
Centigrade
-20
125

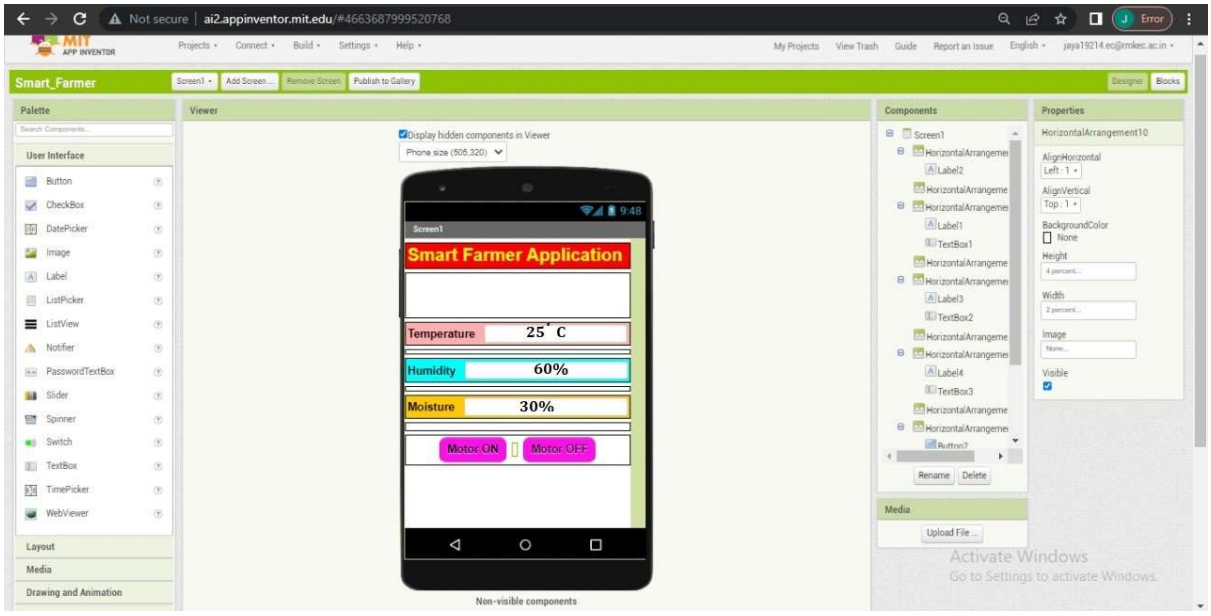
9. RESULTS

9.1 Performance Metrics






14dcvs.internetofthings.ibmcloud.com/dashboard/boards/b9832d30-ab71-425e-922b-a0b4a0e1c958
IBM Watson IoT Platform
111719106060@smartinternz.com
ID: 14dcvs
TEST VALUE
Add New Card
Settings
Temperature
98.0
°C
Activate Windows


Not secure | ai2.appinventor.mit.edu/#4663687999520768
Smart_Farmer
Palette
Viewer
Components
Properties
Smart Farmer Application
Temperature
25 C
Humidity
60%
Moisture
30%
Motor ON
Motor OFF
Non-visible components
Activate Windows
Go to Settings to activate Windows.

10. ADVANTAGES & DISADVANTAGES

Advantages :

- 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.
- A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labour 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.

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

The project concludes that this system is easy to implement and time, money, and manpower saving solution for irrigating fields. It will be comfortable for farmers to operate the irrigation at remote locations i.e. from home. A farmer should visualize his agricultural land's moisture content from time to time and whether the water level of the source is sufficient or not. The IOT based irrigation system displays the values of the sensors continuously in smart phones and farmers can operate them anytime from and anywhere. This will save time and avoid the problem of continuous vigilance. Not only this, it will also control the consumption of water for the irrigation of the field, thus preventing the water wastage and would help in sustaining productivity, increasing the yield.

An IoT based irrigation system aims to utilize the features of embedded systems to make agriculture simple. Having sensors connected to the controller, the system reads the soil moisture and temperature of the soil and then the sensed data is processed in the controller. The microcontroller is the decision maker of this system. It checks for moisture value and the temperature. Initially, the threshold moisture and temperature value must be defined. When the sensed moisture value goes above the threshold value, the controller checks for the temperature. Only if the sensed temperature value is higher than the threshold value, irrigation is done and the user is acknowledged. This is because all crops can withstand the dry soil moisture condition if the temperature is moderate. This would conserve the water for irrigation. Sending SMS to the user about the field enables the user to remotely monitor the agriculture area. The SMS includes the warning and suggestion to the affected system.

12. FUTURE SCOPE

● The system can not only used in field by the farmers but can be used to solve other problems where continuous monitoring of water supply is required like in a garden, or a personal small field, or in the watering the stadium when necessary etc.

● This project can be made further more innovative by adding – controlling and monitoring the sprinkles, checking the faults in the irrigation network and correcting them remotely and visualization the live working of integrated systems in the field area by mobile.

● Also the future aspect of this model can be made into an intelligent system, wherein the system predicts user actions, rainfall pattern, time to harvest and many more features which will make the system independent of human operation.

● Systems can also be upgraded to Real Time systems, such that users receive real time updates and status of condition of the field. Thereby, enabling the user to take immediate action in case of any problems

13. APPENDIX

Source Code

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "14dcvs",
        "typeId": "Device1",
        "deviceId":"12345"
    },
    "auth": {
        "token": "87654321"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="Motor On"):
        print("****///Motors ARE ON///****")
    else:
        print("****///Motors ARE OFF///****")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    Mois=random.randint(20,120)
    myData={'temperature':temp, 'humidity':hum, 'moisture':Mois}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

GitHub & Project Demo Link

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-30601-1660150205

Project Demo Link

https://clipchamp.com/watch/mFEDIL8tikQ