PANIMALAR INSTITUTE OF TECHNOLOGY

CHENNAI – 600 123

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SMART FARMER-IOT ENABLED SMART FARMING APPLICATION

A NALAIYA THIRAN PROJECT REPORT

Submitted by

KUPPASWAMY MEGHANA	(211519104076)
HARITHA. P	(211519104053)
KEERTHANA. K.P	(211519104069)
VITHYASREE. S	(211519104186)

In the 7th semester

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

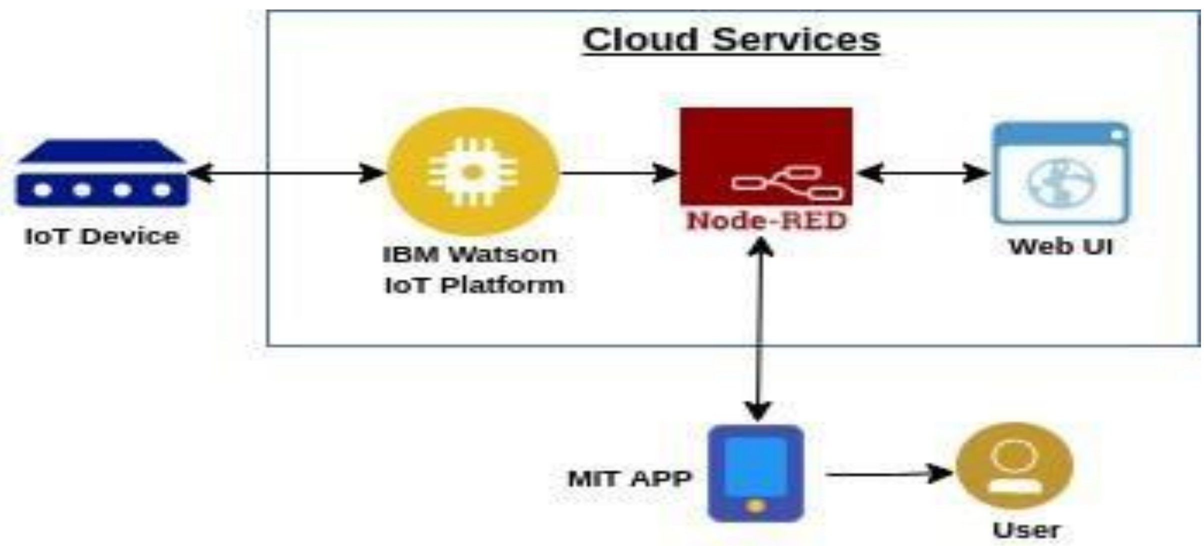
During the academic year 2022-2023

TABLE OF CONTENTS

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.



1.2 PURPOSE:

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.

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM:

The smartirrigator receives signal from smart farm sensingsystem throughzigbee module. The sensed data is transferred towards central database from which all crop details are analyzed and transferred to irrigator system to performautomatic actions.

All these operations will be controlled through any smart device placed remotely and the interfacing sensors are used to perform operations along with Wi-Fi, actuators and other hardware devices.The biggest challenges faced by IoT in the agricultural sector are lack of information, high adoption costs, and security concerns, etc. Most of the farmers are not aware of the implementation.

2.2REFERENCE:

1. TITLE: Smart Farming System Using Sensors.
AUTHOR: Chetan Dwarkani M, Ganesh Ram R, Jagannathan S, and R. Priyatharshini
YEAR: 2019

2.TITLE: Sensor Networks for Smart Farming

AUTHOR: Sinung Suakanto,Ventje J. L. Engel, Maclaurin Hutagalung

YEAR: 2019

3.TITLE: Intelligent Farming by using ArduinoTechnology

AUTHOR: Narayut Putjaika, Sasimanee Phusae, AnupongChen-Im
YEAR: 2020

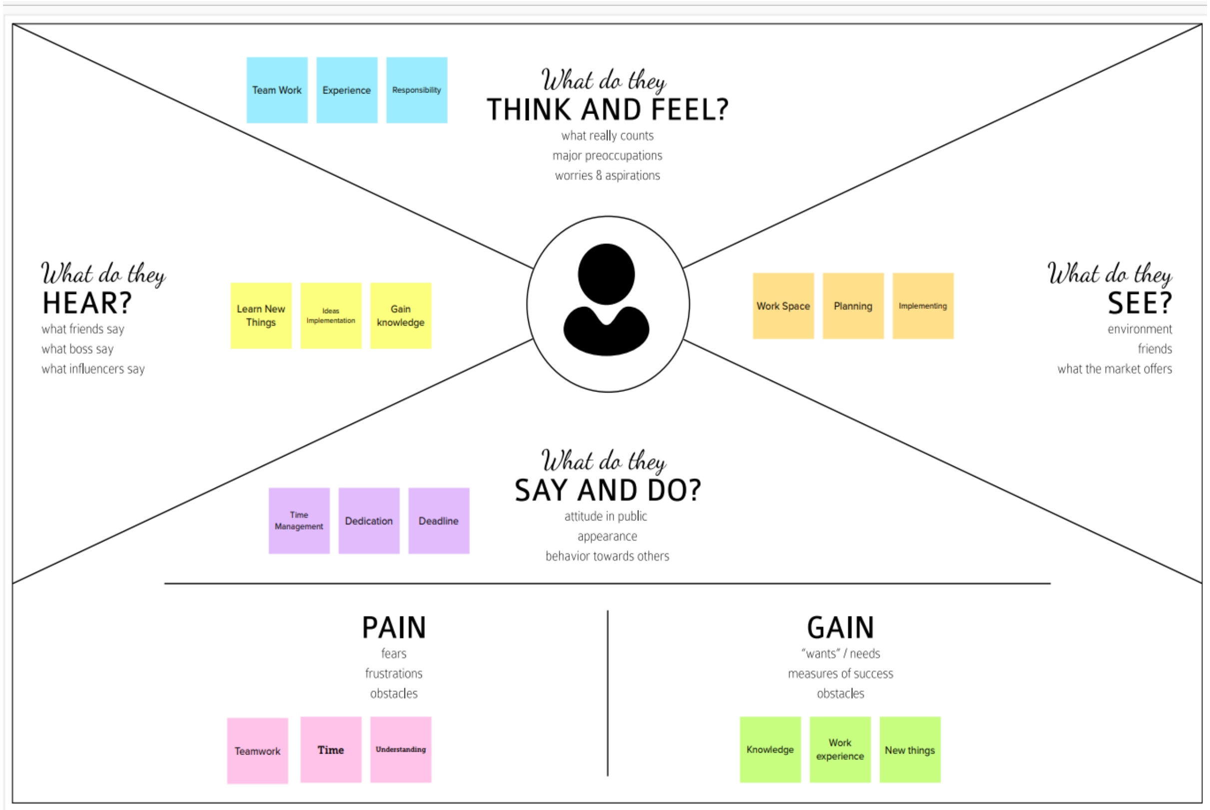
2.3 PROBLEM STATEMENT DIFINITION:

Farmers are to be present at farm for its maintenance irrespective of the weather.They have to ensure that the crops are well watered and the farm status is monitored by them physically. Farmer have to stay most of the time in field in order to get a good yield. In difficult times like in the presence of pandemic also they have to work hard in their fields risking their lives to provide food for the country.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem ant the person who is experiencing it The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation & Brainstorming:

TEAM IDEAS:

KUPPASWAMY MEGHANA:

- Automate irrigation process using temperature of soil.
- Automate irrigation using measurement of moisture of soil.

HARITHA P:

- We can use sensors on sensing.
- We can sense and program the moisture level.

KEERTHANA K.P:

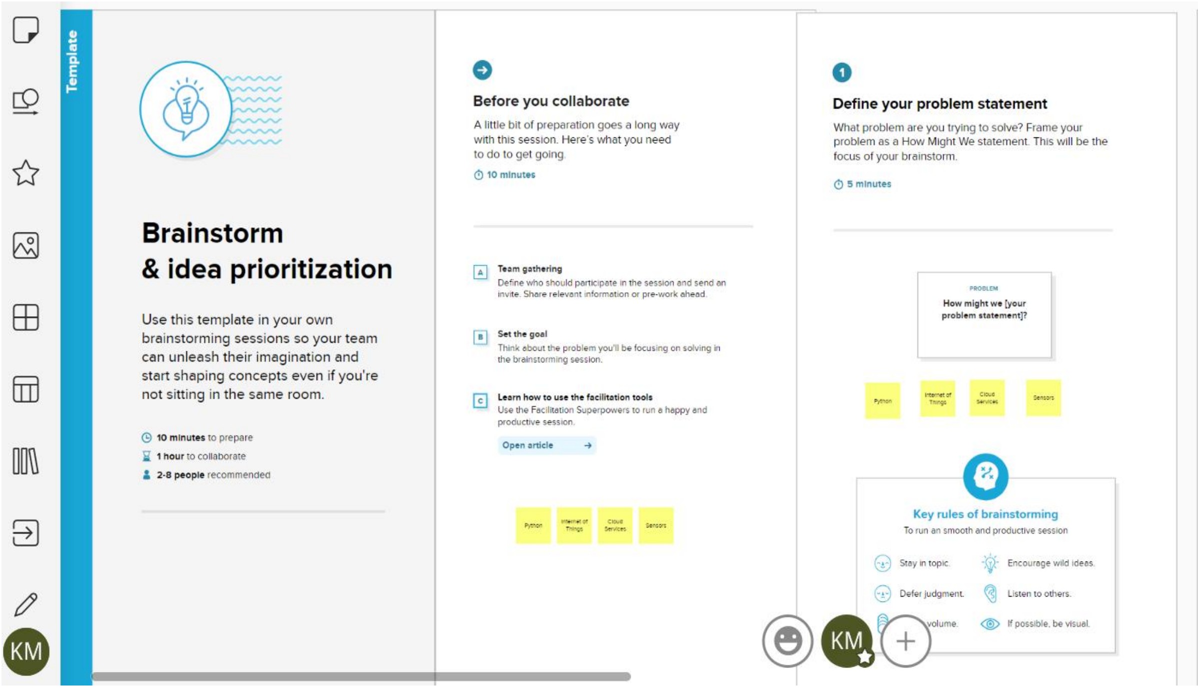
- We can simplify the drip irrigation into time controlled irrigation.
- Automate irrigation using any Robots.

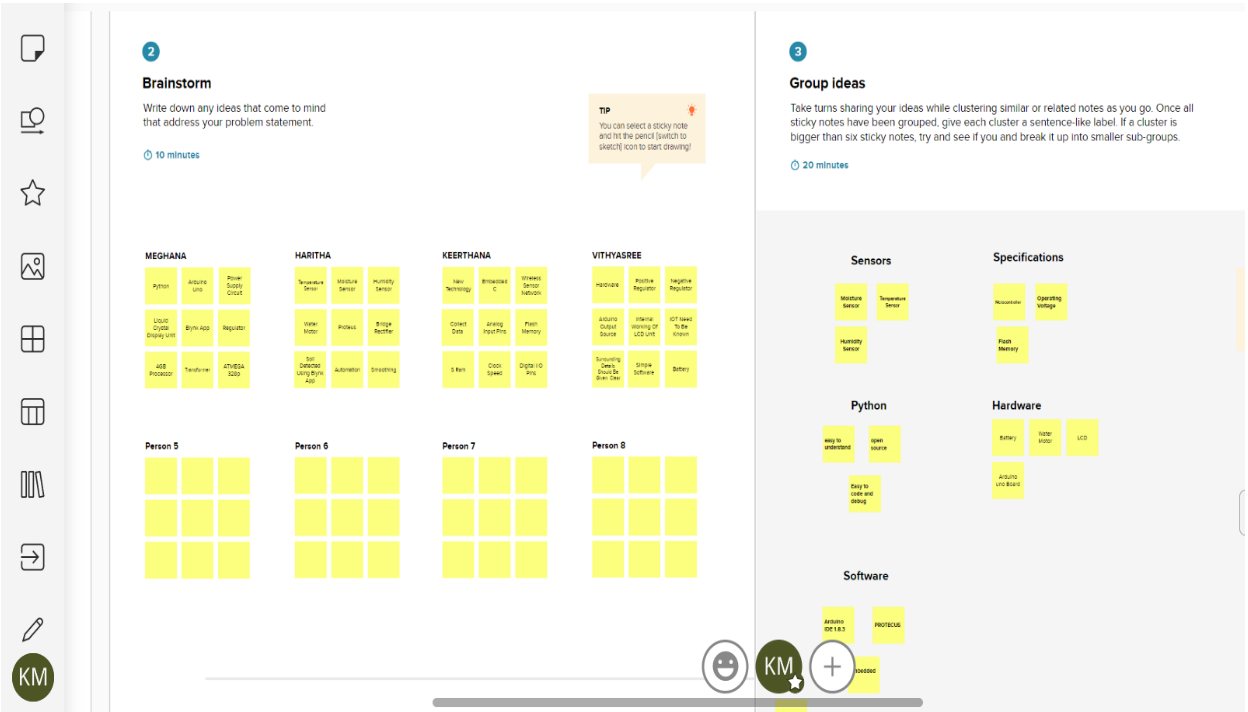
VITHYASREE S:

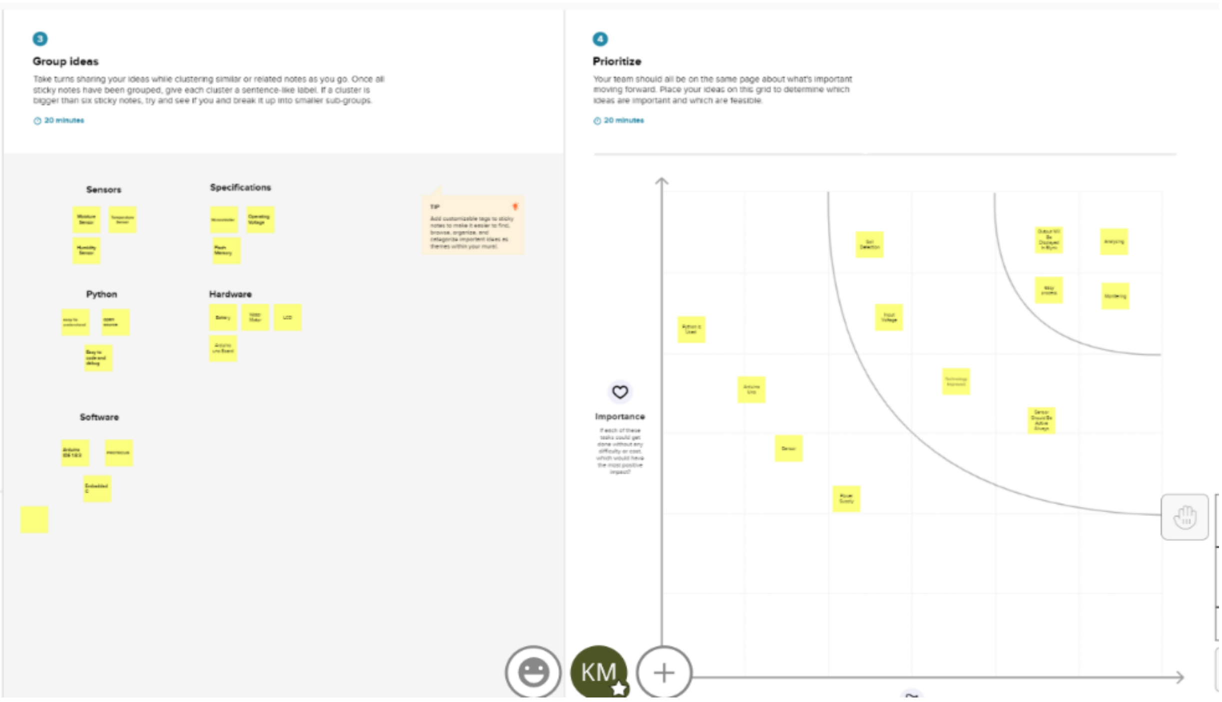
- We can automate and design Audino for programming.
- We can make good design and programming of soil moisture and temperature.

Best Three Ideas:-

- Automate irrigation using measurement of moisture of soil.
- We can sense and program the moisture level.
- We can automate and design Audino for programming.


Template
Brainstorm
& idea prioritization
Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.
10 minutes to prepare
1 hour to collaborate
2-8 people recommended
Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes
Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.
Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.
Open article
Define your problem statement
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes
How might we [your problem statement]?
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
If possible, be visual.
KM


Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
TIP
MEGHANA
HARITHA
KEERTHANA
VITHYASREE
Person 5
Person 6
Person 7
Person 8
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.
20 minutes
Sensors
Specifications
Python
Hardware
Software
KM



3.3 <u>Proposed Solution:</u>

Proposed Solution Template:

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

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to be solved)	To incorporate the process of working and also elevate the smart farming using IOT enabled smart Farming technique since the traditional Farming technique I very Complex one.
2.	Idea / Solution description	To automate irrigation in accordance to the amount of moisture present in soil
3.	Novelty / Uniqueness	Automation of irrigation to amount of moisture

4.	Social Impact / Customer Satisfaction	The problems faced by the farmers in the process of irrigation gets solved and this full fills and saves their crops from over irrigation
5.	Business Model (Revenue Model)	The process of fulfilling this process brings revolution in drip irrigation systems also makes a revolutionary change in market
6.	Scalability of the Solution	The design scale of solution has been planned in a compact manner

3.4 Problem Solution fit:

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS	6. CUSTOMER CONSTRAINTS CC	5. AVAILABLE SOLUTIONS A
	Who is your customer? i.e working parents of 0-5 y.o kids	What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.	Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e pen and paper is an alternative to digital note taking
	Proprietary systems in which the farmer is part of a highly integrated food supply chain	Available Devices, Network Connection, Fit to environment	Livestock tracking and Geo fencing, Smart logistics and warehousing, Smart pest management. Pros: It follows farmers to maximize yields using minimum resources. Cons: Rural part of most of the developing countries do not fulfil this requirement.

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explored different sides.

The integration of these sensors and tying the sensor data to the analytics driving automation and response activities.

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e customers have to do it because of the change in regulations.

Poor Internet Connectivity in Farms
High Hardware Costs
Disrupted Connectivity to the cloud

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

By installing an IOT-powered farming solution, the farmers can interconnect their smartphones and monitor their field activities remotely.

Identify strong TR & EM

3. TRIGGERS

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

A system that is built for monitoring the crop field with the help of sensors and automating the irrigation system.

4. EMOTIONS: BEFORE / AFTER

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

The hardware and materials used to develop prototype allowed us to make an efficient and accurate, as well as cheap product. It is economical and easily installable for farmer as well.

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

Current solution is precision agriculture. It uses digital technologies. By using sensors to collect data on weather, soil moisture, crop health and real-time locational asset tracking.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Online: It uses robots, drones, remote sensors for monitoring crops, surveying, mapping the fields and providing data.

Offline: The application of sensors and automated irrigation practices can help monitor agricultural land, temperature, soil moisture.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Measure Temperature	Soil thermometers are the most common Tool for measuring soil temperature. The voltage across the diode terminals
FR-2	Measure soil moister	Sensor for soil scanning and water, light, humidity and temperature management
FR-3	Calculating the date and time	Time of day: Between 1 and 2 p.m. Depth :4 inches below the soil surface Soil Location: Same area of field, soil type weather and precipitation
FR-4	Irrigating the soil if needed	A moisture supply for plant growth which also transports essential nutrients. A flow of water to leach or dilute salts in the soil

4.2 NON-FUNCTIONAL REQUIREMENT:

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

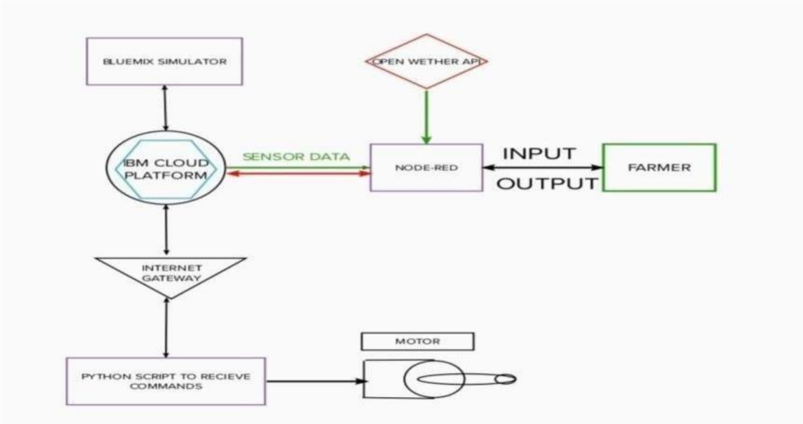
Non-functional Requirements:

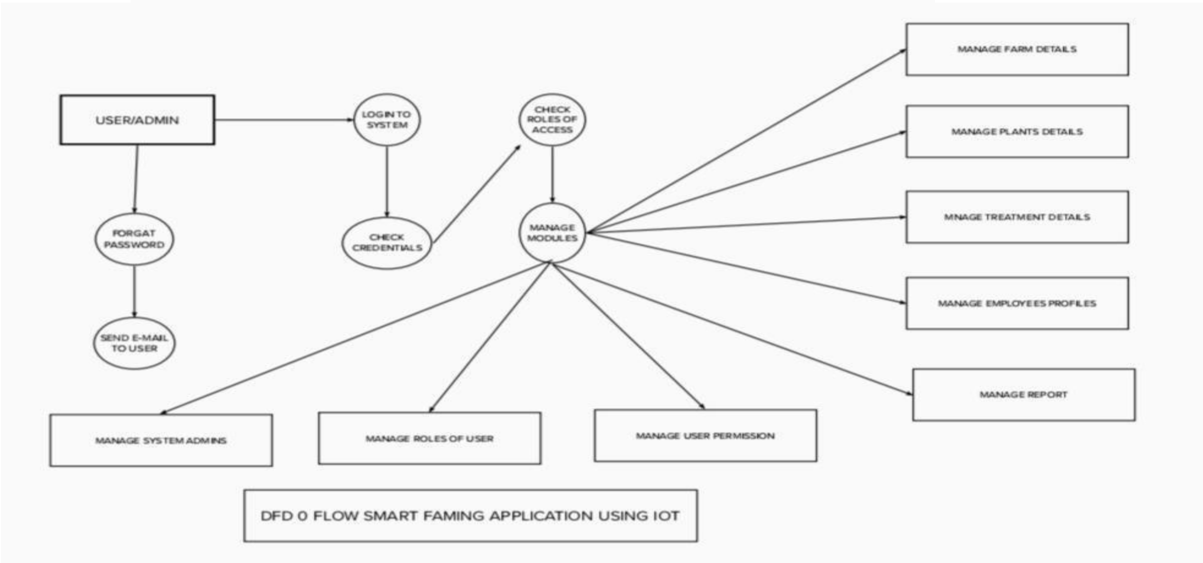
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Smart farming has enabled farmers to reduce waste and enhance productivity with the help of sensors and automation of irrigation systems. Further with the help of these sensors, farmers can monitor the field conditions from anywhere.
NFR-2	**Security**	The adoption of sensor based technologies and cloud supported smart applications in agriculture has unleashed opportunities for adversaries to orchestrate cyber attacks. Therefore, it is important to first understand major security and privacy issues in smart farming domain before discussing specific cyber attacks.
NFR-3	**Reliability**	An adaptive network mechanism is designed to improve the network performance of the system, in order to achieve a more reliable smart farm system.
NFR-4	**Performance**	The system was tested on okra plants and their vegetative traits were measured for 30 days. The result revealed good performance which proves that the developed system is suitable for smart farming system.
NFR-5	**Availability**	With smart sensors that monitor every aspect of everyday work automatically, IOT technology for agriculture allows farmers to automate real-time data collection to increase production volumes, reduce costs and manage expenses and improve overall efficiency in many different aspects of agriculture.
NFR-6	**Scalability**	It 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.

5. PROJECT DESIGN

5.1 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.

The different soil parameters temperature, soil moistures and then humidity are sense during different sensors and obtained value is stored in the IBMcloud.

Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weatherAPI.

5.2 Solution & Technical Architecture :

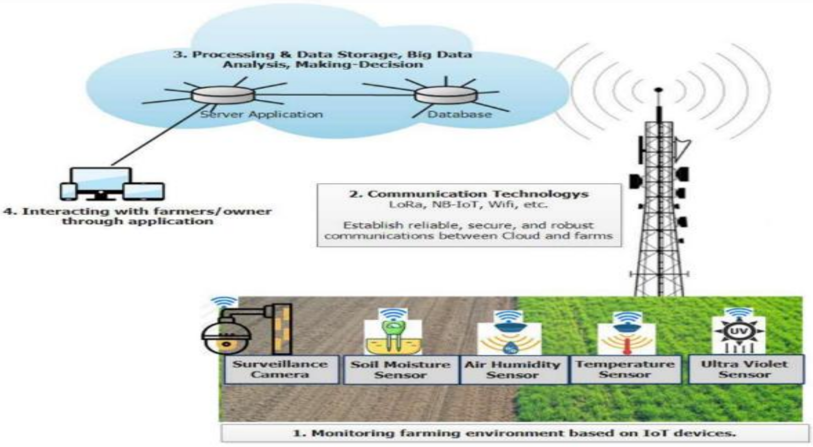


Figure 1: An illustration of IoT ecosystems' architecture for smart agriculture.

5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once 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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access with Gmail.	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive the login credentials.	High	Sprint-1
	Dashboard	USB-6	As a user, I can check data in visualized graphs on the application developed by mile sight.	I can check the data in graph.	Medium	Sprint-1
Customer (Web user)	Software	USB-7	As a user, I can also use a third-party application server or develop my own application.	I would recommend that I can use my own data.	Low	Sprint-2
Customer Care Executive	Connectivity	USB-8	Those graphs are useful to farm operators to modify their crop health management system.	I agree that it is useful to modify the system.	Medium	Sprint-1
Administrator		USB-9	The gateways collect data from sensors and pushes them to the things network platform.		High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	3	High	KUPPASWAMY MEGHANA HARITHA P
Sprint-1	Software	USN-1	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	5	Medium	KEERTHANA K P VITHYASREE S
Sprint-2	Simulation creation	USN-2	Connecting Arduino python code in laptop with Arduino Breadboard.	4	High	KUPPASWAMY MEGHANA KEERTHANA K P
Sprint-3	Login	USN-3	Then Connecting Hotspot in mobile phone through Blynk App.	2	Medium	VITHYASREE S KUPPASWAMY MEGHANA
Sprint-4	Dashboard	USN-4	If the sensor is inserted inside ground surface then the output will viewed in Blynk app.	4	High	KEERTHANA K P HARITHA P

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		3 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		8 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		10 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

7. CODING & SOLUTIONING

7.1 Feature 1:

```
import wiotp.sdk.device
import time
import os
import datetime
import random myConfig ={
        "identity": {

        "orgId": "0hzydu",

        "typeId": "NodeMCU",

        "deviceId": "12345"

   },

   "auth": {

      "token": "12345678"

   }

}
client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
 client.connect ()
def myCommandCallback (cmd) :
    print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
    m=cmd.data['command']

    if (m=="motoron"):
```

```
      print("Motor is switchedon")

    elif (m=="motoroff"):


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

7.2 Feature 2:

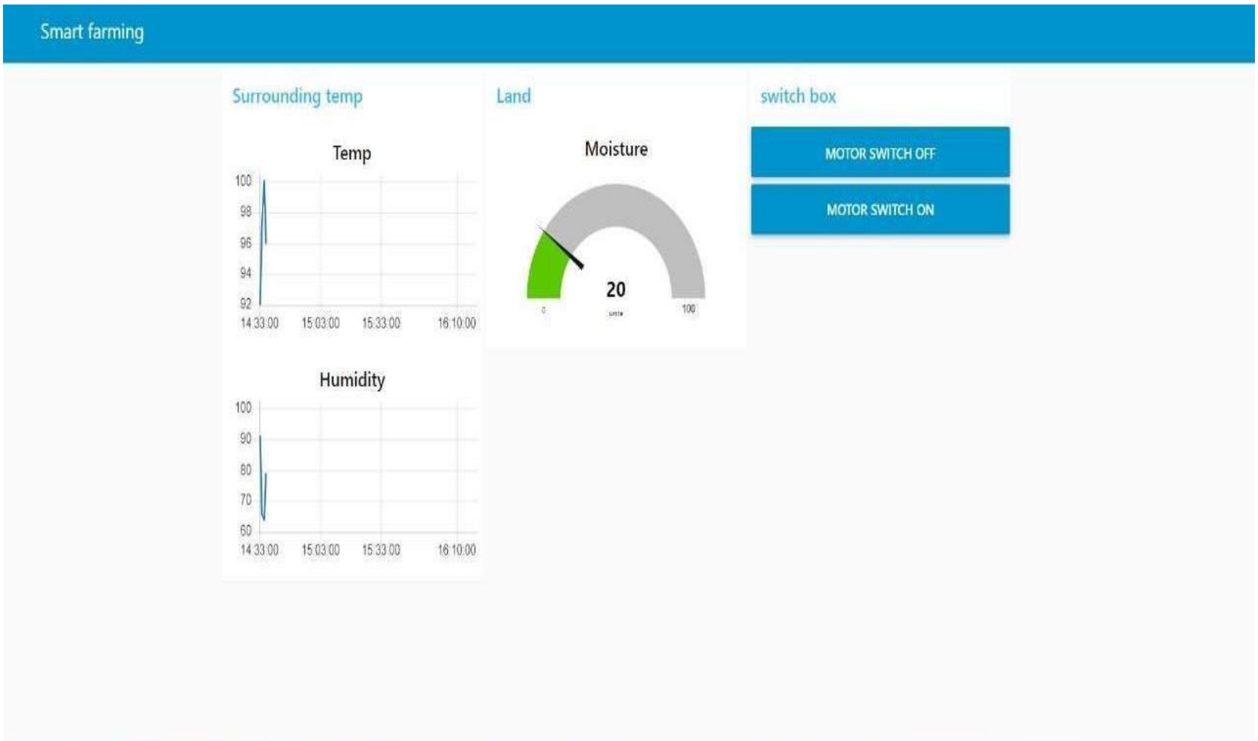
```
/* Plant Watering System

The circuit: -Water pump Power supply: 4.5~12V DC

 Interface: Brown +; Blue - -

Temperature/moisture sensor Power supply: 3.3-5v –

Moisture sensor Power supply: 3.3-5v */

#include "DHT.h"

 #define DHTPIN 2 // what digital pin we're connected to

 #define DHTTYPE
  DHT22 // DHT 22 (AM2302),
  AM2321
  DHT dht(DHTPIN, DHTTYPE);

  const int SOIL_MOISTURE_SENSOR_PIN = A0;

  const int WATER_PUMP_PIN = 4;

  const int dry = 520;
   const int wet = 270;
   const int moistureLevels = (dry - wet) / 3;

// TODO: Should we have a counter so if it waters for X times, then take a break?

// OPTIMIZE: how dry to start watering and for how long.

 const int soilMoistureSartWatering = 400;

 const int soilMoistureStopWatering = 300;
```

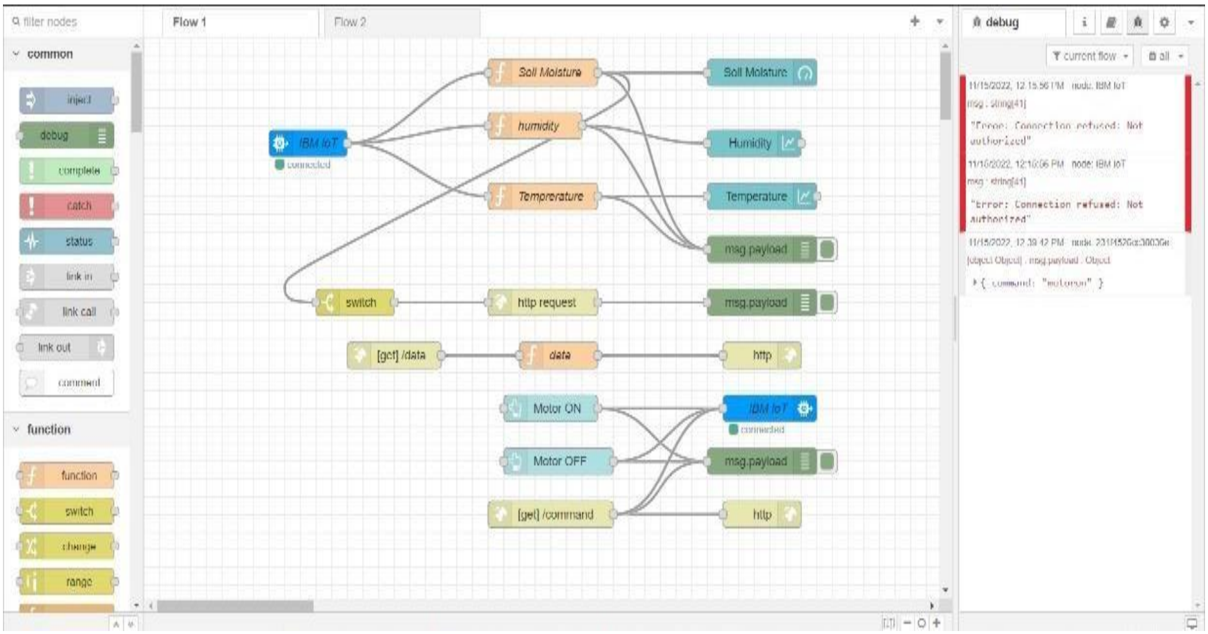
```
// 60 seconds const long waterDuration = 1000L
* 60L;
// 60 seconds const long sensorReadIntervals = 1000L *
60L;
// 2 hr

const long waterIntervals = 1000L * 60L * 60L * 2;
 long lastWaterTime = waterIntervals - 1;
 boolean isWatering = false;
 void setup()

{

 Serial.begin(9600);

 pinMode(WATER_PUMP_PIN, OUTPUT);

 waterPumpOff();

dht.begin();

}

void loop()

{

 mainLoop ();

}

void mainLoop() { float temperature = getTemperature(); float humidity = getHumidity();

long soilMoisture = analogRead(SOIL_MOISTURE_SENSOR_PIN);

Serial.println("Soil Moisture: " + readableSoilMoisture(soilMoisture) + ", " + soilMoisture);

Serial.println("Temperature: " + String(temperature) + " *F");
```

```
Serial.println("Humidity: " + String(humidity) + " %");



if (millis() - lastWaterTime > waterIntervals)

{

waterPlants(soilMoisture); lastWaterTime = millis();

}

delay(sensorReadIntervals);

}

void waterPlants(int soilMoisture) {

// Should this take a moving avg of the soilMoisture?

// Can get outliers on the right after watering. if (soilMoisture > soilMoistureSartWatering) {
isWatering = true
```

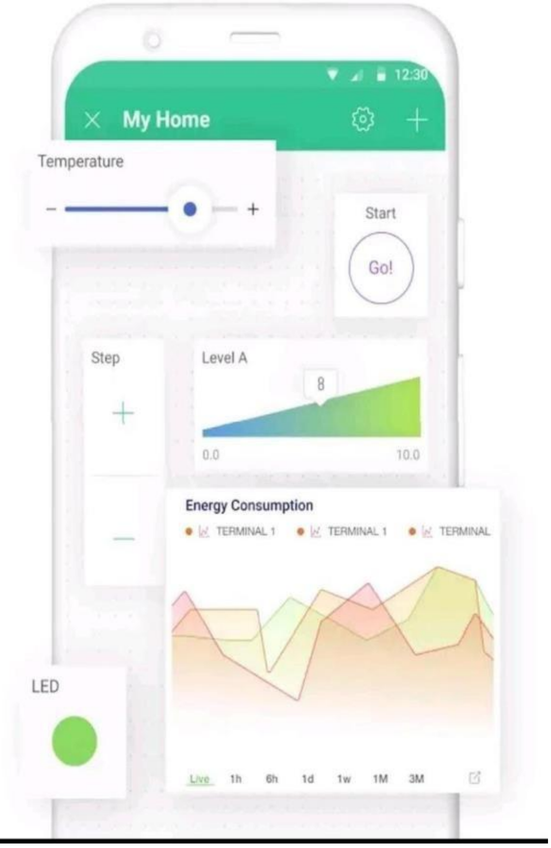
8. TESTING

8.1 Test Cases:



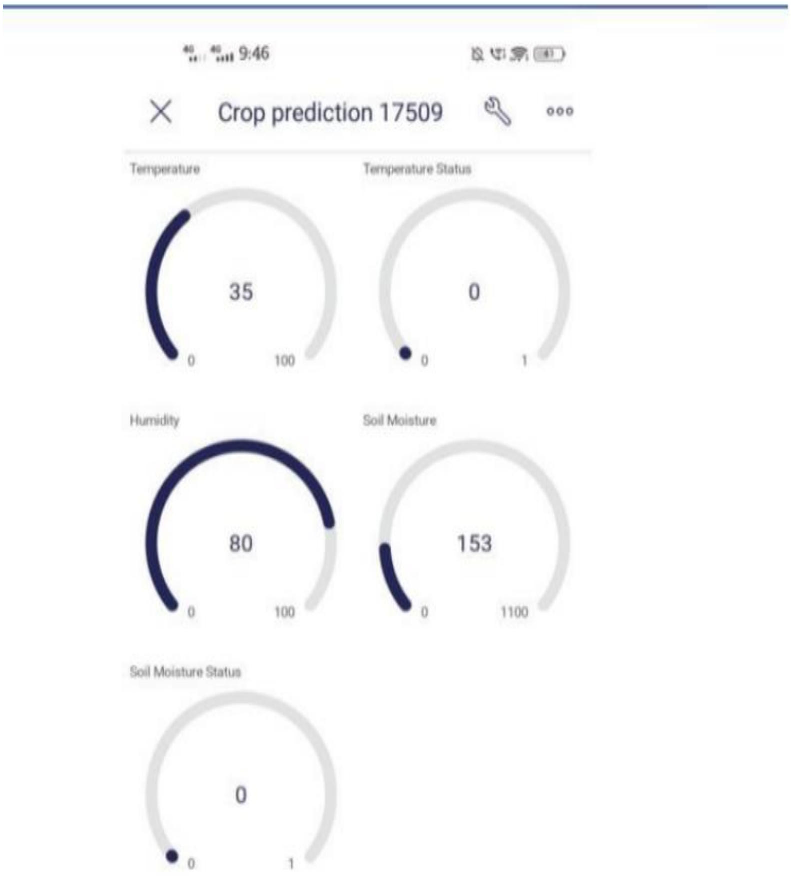


8.2 User Acceptance Testing:



9. RESULTS

9.1 PERFORMANCE METRICS:



10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

1.High efficient

2.User friendly

3.Easy to install

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

DISADVANTAGES:

1.Less efficient

2.Covers short area.

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

11. CONCLUSION

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

12. FUTURE SCOPE

- In future due to more demand of good and more farming in less time, for betterment of the crops and reducing the usage of extravagant resources like electricity and water IOT can be implemented in most of the places.

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

- We can create few more models of the same project, so that the farmer can have information of an entire.

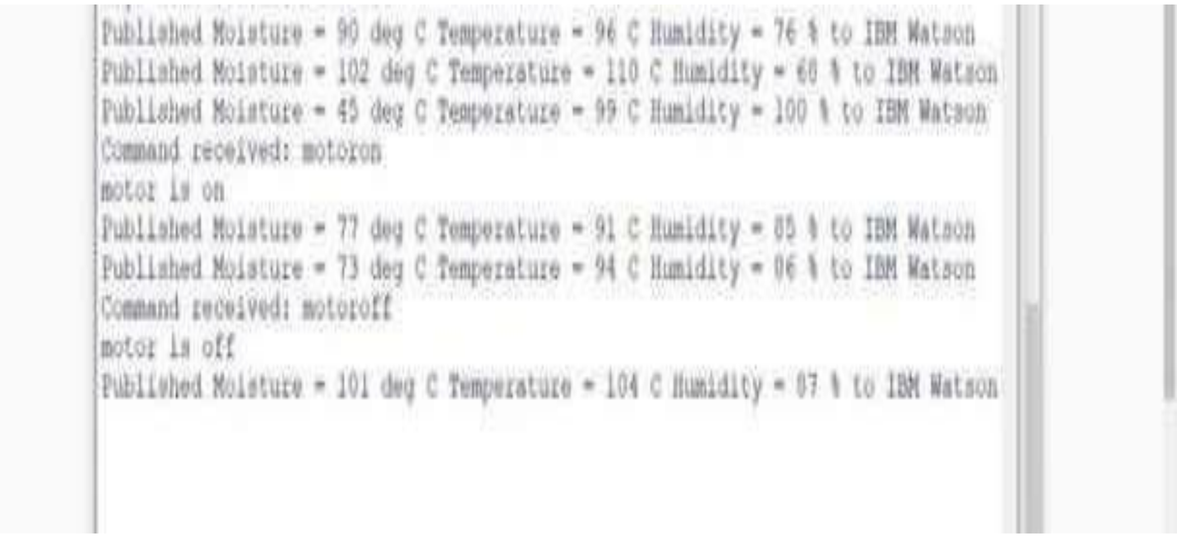
13. APPENDIX

SOURCE CODE:

```
import wiotp.sdk.device
import time
import os
import datetime
import random myConfig ={
   "identity": {

        "orgId": "0hzydu",

         "typeId": "NodeMCU",

         "deviceId": "12345"

   },

   "auth": {

       "token": "12345678"

   }

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

 client.connect () def myCommandCallback (cmd) :

   print("Message received from IBM IoT Platform: %s"

%cmd.data['command'])    m=cmd.data['command']

  if (m=="motoron"):

       print("Motor is switchedon")
```

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

OUTPUT:



```
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 96 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

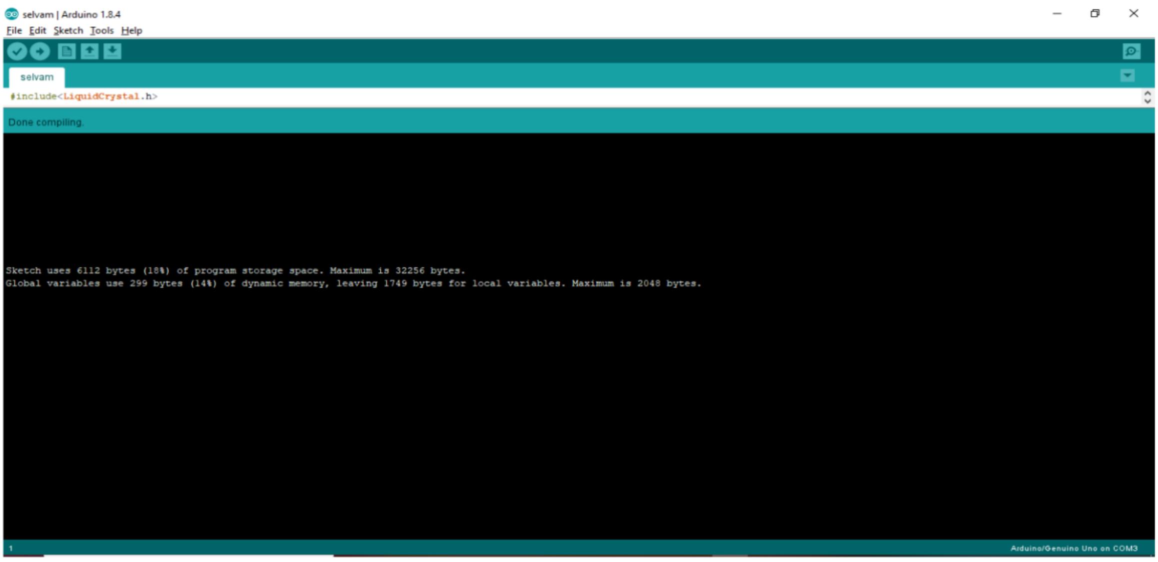
PYTHON CODE:

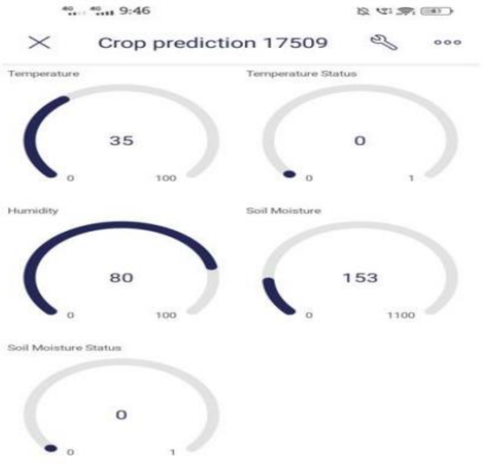
```
// Fill-in information from your Blynk Template here

#define BLYNK_TEMPLATE_ID "TMPLEHXAYFKX"

#define BLYNK_DEVICE_NAME "Crop prediction"

// aravindkarthick2630@gmail.com

//pw: iotproject2022

#define BLYNK_FIRMWARE_VERSION  "0.1.3"

#define BLYNK_PRINT Serial

//#define BLYNK_DEBUG

//#define APP_DEBUG

// Uncomment your board, or configure a custom board in Settings.h

//#define USE_SPARKFUN_BLYNK_BOARD

#define USE_NODE_MCU_BOARD

//#define USE_WITTY_CLOUD_BOARD

int c;

//Change the virtual pins according the rooms

#define VPIN_1    V0  //temp

#define VPIN_2    V1  //temp stat

#define VPIN_3    V2  //huminity

#define VPIN_4   V3  //moist

#define VPIN_5   V4  //moist status
```

```
#include "BlynkEdgent.h"

int t,a,h,m,b;

BLYNK_CONNECTED()

{

Blynk.syncVirtual(VPIN_1);

  Blynk.syncVirtual(VPIN_2);

  Blynk.syncVirtual(VPIN_3);

   Blynk.syncVirtual(VPIN_4);

    Blynk.syncVirtual(VPIN_5);

}

void setup()

{

  Serial.begin(9600);

  delay(100);

  BlynkEdgent.begin();

}

void loop()

{

  BlynkEdgent.run();

  if (Serial.available()>0)

  {
```

```
c=Serial.read();

// Serial.print(a);

if(c=='T'){  t=Serial.parseInt(); Blynk.virtualWrite(VPIN_1,t); }

else if(c=='A'){  a=Serial.parseInt(); Blynk.virtualWrite(VPIN_2,a); }

else if(c=='H'){  h=Serial.parseInt(); Blynk.virtualWrite(VPIN_3,h); }

 else if(c=='M'){  m=Serial.parseInt(); Blynk.virtualWrite(VPIN_4,m); }

 else if(c=='B'){  b=Serial.parseInt(); Blynk.virtualWrite(VPIN_5,b);

 }

}
```

OUTPUT:





DEMO LINK:

https://drive.google.com/file/d/1Iv0OaIUgYFpW5PriTu4X1pC3VCEWDVrt/view?usp=sharing

GITHUB LINK:
https://github.com/IBM-EPBL/IBM-Project-1329-1658384306