SMART FARMER - IOT ENABLED SMART FARMING APPLICATION

Team ID: **PNT2022TMID43644**
Team Leader : **AMALDAS K K**
Team Members :

1) **ABISHEAK A**
2) **RESHMA R**
3) **NAYANA P**

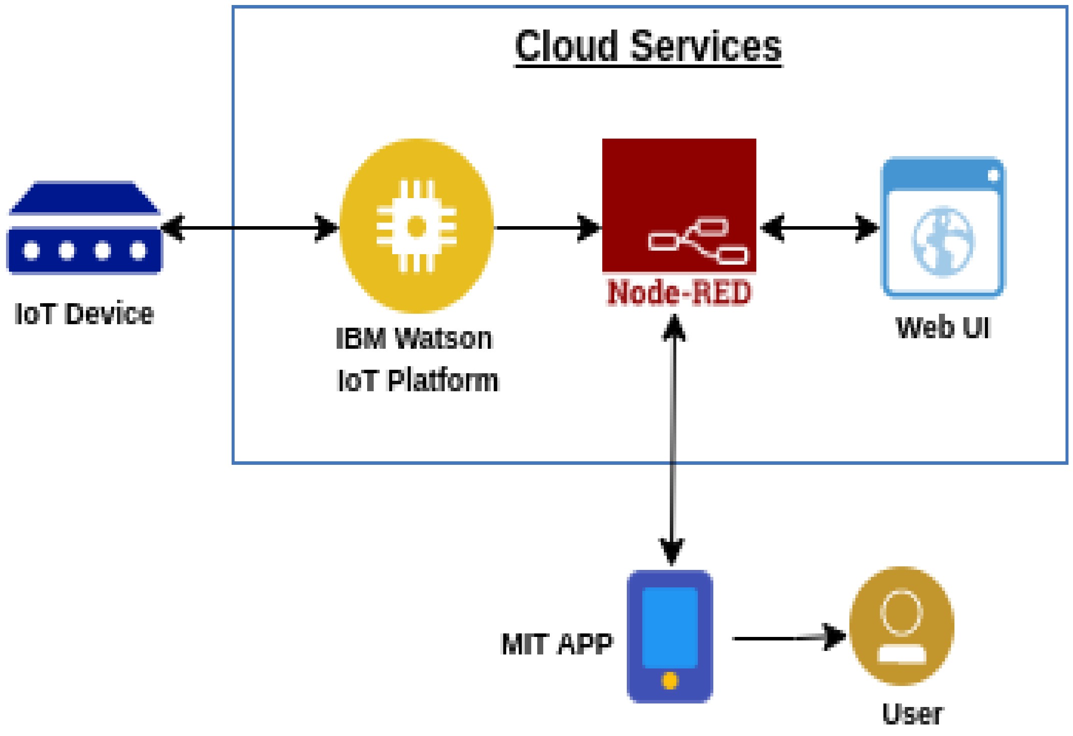
CATEGORY : Internet Of Things

Skills Required :
Python,IBM Cloud,IBM IoT Platform,IBM Nodered,IBM Cloudant DB

PROJECT DESCRIPTION

- 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

TECHNICAL ARCHITECTURE



PROJECT OBJECTIVES

By the end of this project you will:

- Gain knowledge of Watson IoT Platform
- Connecting IoT devices to the Watson IoT platform and exchanging the sensor data
- Explore python client libraries of Watson IoT Platform
- Gain knowledge on IBM Cloudant DB
- Configuring APIs using Node-RED for communicating with a mobile application
- Creating a Mobile Application through which the user interacts with the IoT device

PROJECT FLOW

- The parameters like temperature, humidity, and soil moisture are updated to the Watson IoT platform
- The device will subscribe to the commands from the mobile application and control the motors accordingly
- APIs are developed using Node-RED service for communicating with Mobile Application
- A mobile application is developed using the MIT App inventor to monitor the sensor parameters and control the motors

To accomplish this, we have to complete all the activities and tasks listed below:

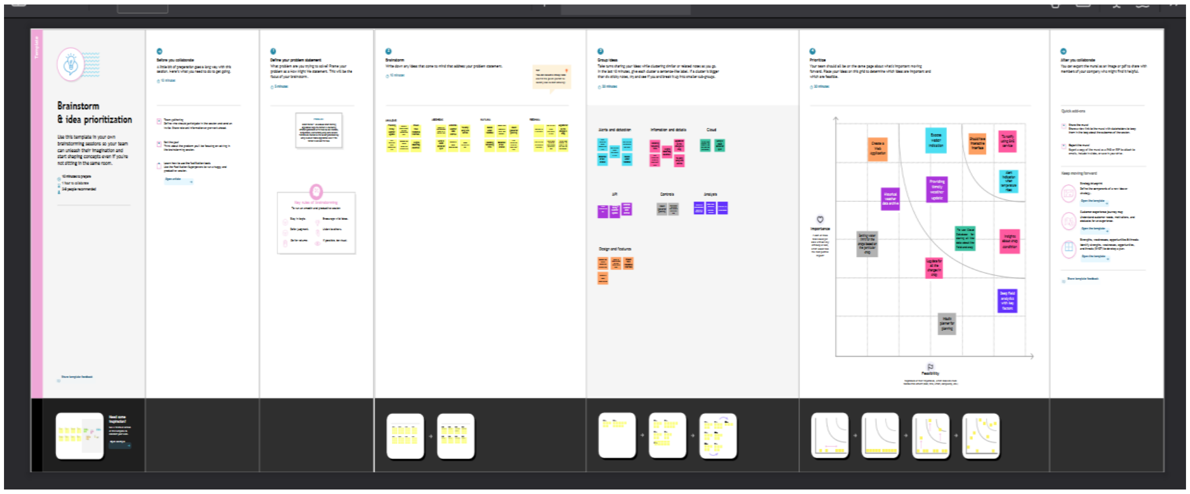
- Create and configure IBM Cloud Services
    - Create IBM Watson IoT Platform
    - Create a device & configure the IBM IoT Platform
    - Create Node-RED service
    - Create a database in Cloudant DB to store all the sensor parameters
- Develop a python script to publish and subscribe to the IBM IoT platform
- Configure the Node-RED and create APIs for communicating with mobile application
- Develop a mobile application to display the sensor parameters and control the motors

DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IoT PLATFORM

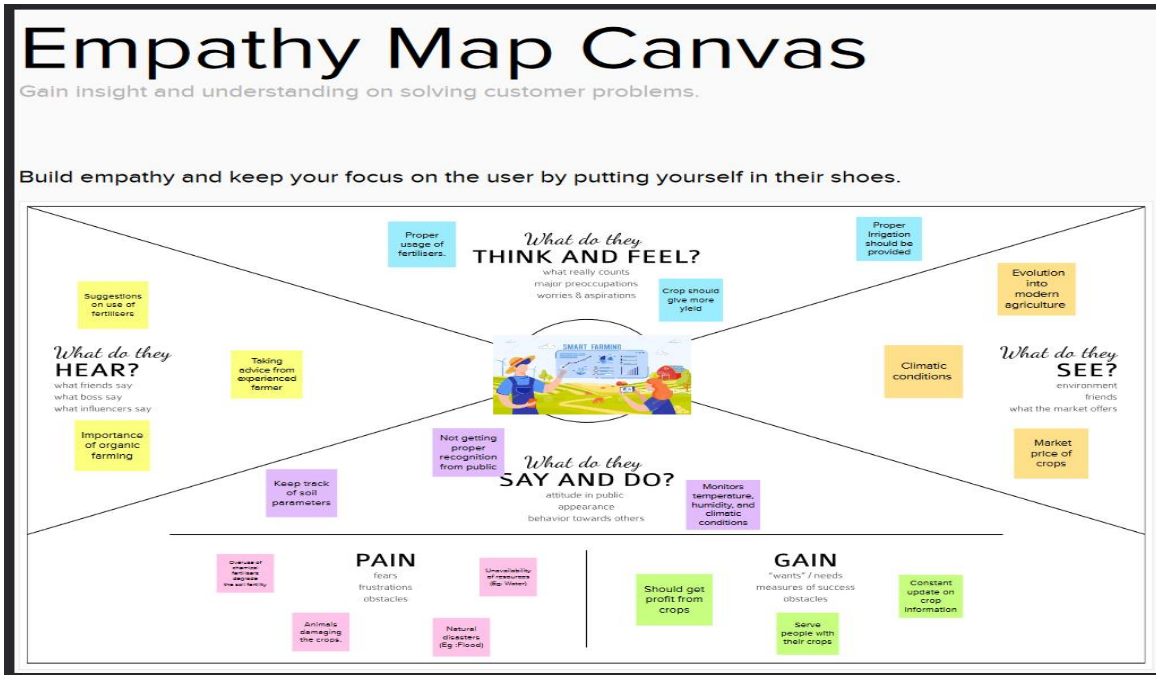
```
from pickle import TRUE import
wiotp.sdk.device import time
import os import datetime
import random

myConfig = {"identity" : {"orgId" : "ka1ghd", "typeId" : "NodeMCU", "deviceId" :
"250402"}, "auth" : {"token" : "9487936557"}}

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 switched on")

   elif(m == "motoroff"):         print("Motor
is switched off")

   print(" ")
while TRUE:
   soil = random.randint(0, 100)     temp =
random.randint(-20, 125)
   hum = random.randint(0, 100)

   myData = {'soil_moisture' : soil, 'temperature' : temp, 'humidity' : hum}

   client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)

   print("Published data successfully : ", myData)

   time.sleep(2)

   client.commandCallback = myCommandCallback

client.disconnect()
```

IDEATION PHASE

IDEATION AND BRAINSTORMING



EMPATHY MAP


Empathy Map Canvas
Gain insight and understanding on solving customer problems.
Build empathy and keep your focus on the user by putting yourself in their shoes.
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Proper usage of fertilisers.
Proper Irrigation should be provided
Crop should give more yield
What do they HEAR?
what friends say
what boss say
what influencers say
Suggestions on use of fertilisers
Taking advice from experienced farmer
Importance of organic farming
What do they SEE?
environment
friends
what the market offers
Evolution into modern agriculture
Climatic conditions
Market price of crops
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Not getting proper recognition from public
Keep track of soil parameters
Monitors temperature, humidity, and climatic conditions
SMART FARMING
PAIN
fears
frustrations
obstacles
Overuse of chemical fertilisers degrade the soil fertility
Unavailability of resources (Eg: Water)
Animals damaging the crops.
Natural disasters (Eg :Flood)
GAIN
"wants" / needs
measures of success
obstacles
Should get profit from crops
Serve people with their crops
Constant update on crop information

LITERATURE SURVEY

S NO	TITLE	Authors	Abstract	Drawbacks

1	Smart farm and monitoring system for measuring the Environmental condition using wireless sensor network - IOT Technology in farming	Tharindu Madushan Bandara Mansoor RAZA	Internet of things (IoT) gives a new proportion of smart farming and agriculture territory. Because with the development of the current world, the internet of things field has peaked with modern technology and modern techniques. In the modern world, IoT is used in every domain like smart city, smart university, etc. This paper is about the implementation of the smart farm. IoT concept helps in cost-efficient farming activities like crop and other resource management. With a wireless sensor network, it is easy to connect with every sensor node placed in the farming environment. Also, with the wireless sensor network, it can connect with long-distance ranges. With the help of a sensor network, it can collect the data from the farming environment and analyze it according to the pre-defined values. The proposed system used IoT sensors to collect the data are soil moisture sensors, temperature sensors, water volume sensors, etc. According to the existing system analysis, the proposed solution contains a smart farm environment and a real-time monitoring system with the wireless sensor network for node connectivity. The proposed system provides a more reliable and flexible smart concept for the farmers, and it is a simple architecture that contains the IoT sensors that collect the data from the farm field and transfer those data through wireless sensor network to the central server and according to the input data, the primary server assigning the task to the paticular devices.	Needs more investment and does not provide specific analysis for farmers.

PROJECT DESIGN PHASE I

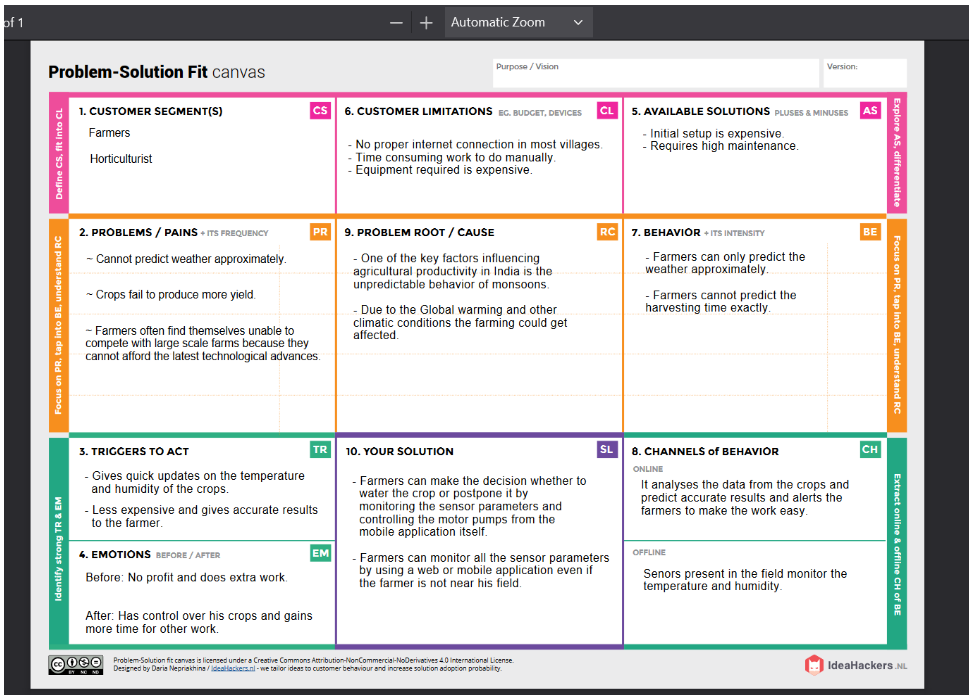
PROPOSED SOLUTION

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

S.No.	**Parameter**	**Description**

1.	Problem Statement (Problem to be solved)	Smart Farmer – Iot enabled smart farming application helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors.
2.	Idea / Solution description	IoT-based remote sensing utilizes sensors placed along with the farms like weather stations for gathering data, which is transmitted to analytical tools for analysis.
3.	Novelty / Uniqueness	1. Alerts the farmers through SMS messages in extreme conditions. 2. Uses cloud services for synchronisation.
4.	Social Impact / Customer Satisfaction	1. Gives timely update for the farmers regarding temperature and humidity levels of the fields. 2. It will be helpful for small scale farm and organic farm.
5.	Business Model (Revenue Model)	Subscription based application for providing analysis of crops and fields.
6.	Scalability of the Solution	1. Easy and simple setup is required and less number of connections and sensors are used for efficient performance. 2. Everything can be controlled from anywhere through cloud.

PROBLEM SOLUTION FIT



of 1 — + Automatic Zoom

Problem-Solution Fit canvas

Purpose / Vision

Version:

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Farmers

Horticulturist

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- No proper internet connection in most villages.
- Time consuming work to do manually.
- Equipment required is expensive.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

- Initial setup is expensive.
- Requires high maintenance.

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

~ Cannot predict weather approximately.

~ Crops fail to produce more yield.

~ Farmers often find themselves unable to compete with large scale farms because they cannot afford the latest technological advances.

9. PROBLEM ROOT / CAUSE RC

- One of the key factors influencing agricultural productivity in India is the unpredictable behavior of monsoons.

- Due to the Global warming and other climatic conditions the farming could get affected.

7. BEHAVIOR + ITS INTENSITY BE

- Farmers can only predict the weather approximately.

- Farmers cannot predict the harvesting time exactly.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

- Gives quick updates on the temperature and humidity of the crops.
- Less expensive and gives accurate results to the farmer.

4. EMOTIONS BEFORE / AFTER EM

Before: No profit and does extra work.

After: Has control over his crops and gains more time for other work.

10. YOUR SOLUTION SL

- Farmers 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.

- Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field.

8. CHANNELS of BEHAVIOR CH

ONLINE

It analyses the data from the crops and predict accurate results and alerts the farmers to make the work easy.

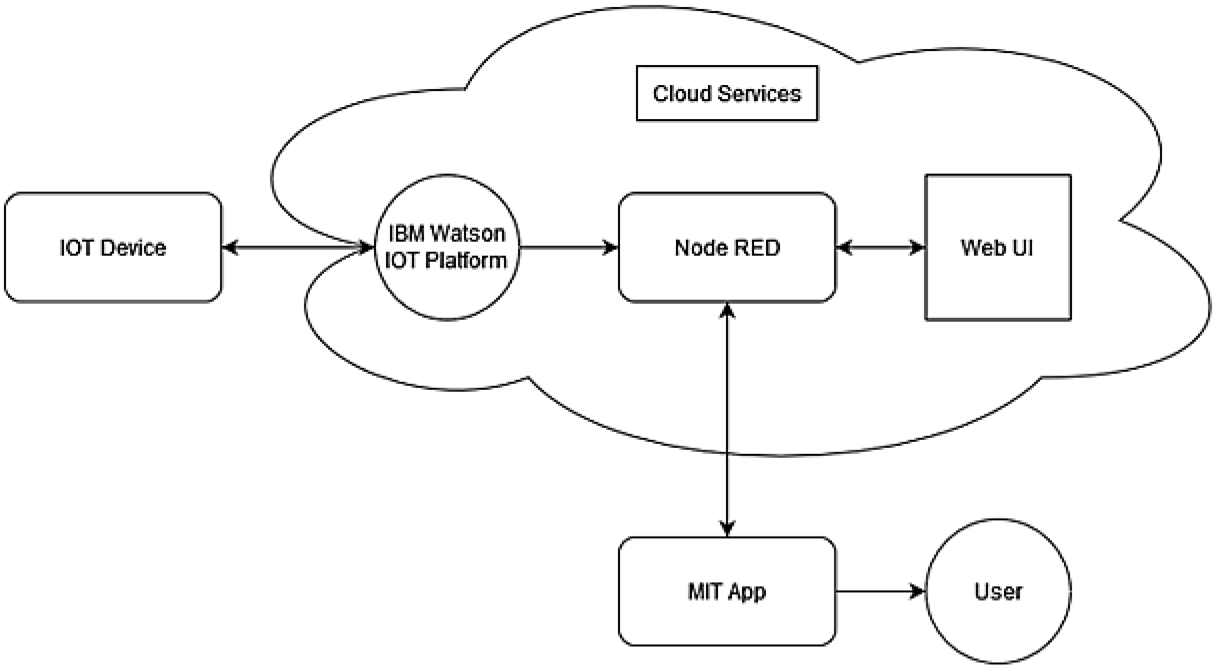
OFFLINE

Senors present in the field monitor the temperature and humidity.

Extract online & offline CH of BE

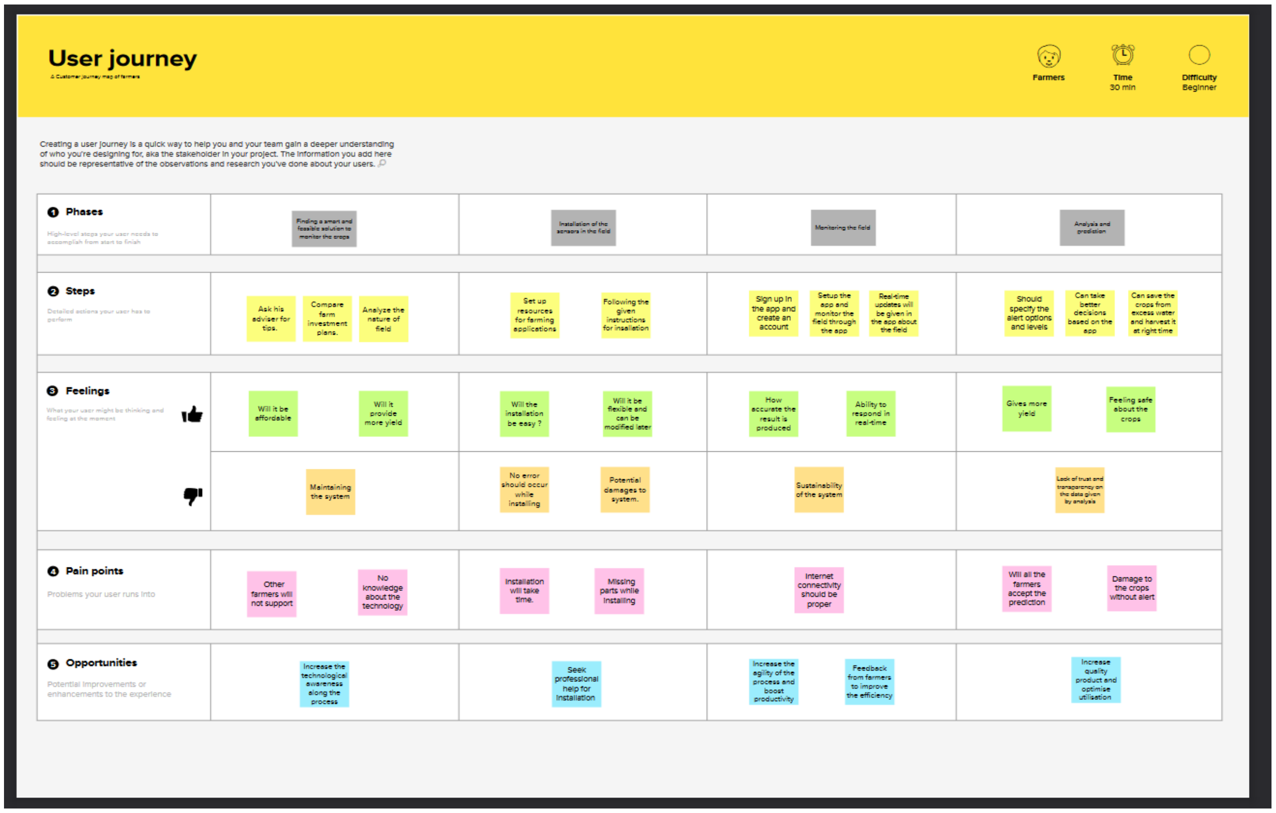
IdeaHackers.NL

SOLUTION ARCHITECTURE



PROJECT DEVELOPMENT PHASE II

CUSTOMER JOURNEY



User journey
A Customer journey map of farmers

Farmers | Time 30 min | Difficulty Beginner

Creating a user journey is a quick way to help you and your team gain a deeper understanding of who you're designing for, aka the stakeholder in your project. The information you add here should be representative of the observations and research you've done about your users.

1 Phases High-level steps your user needs to accomplish from start to finish	Finding a smart and feasible solution to monitor the crops	Installation of the sensors in the field	Monitoring the field	Analysis and prediction
2 Steps Detailed actions your user has to perform	Ask his adviser for tips. / Compare farm investment plans. / Analyze the nature of field	Set up resources for farming applications / Following the given instructions for insallation	Sign up in the app and create an account / Setup the app and monitor the field through the app / Real-time updates will be given in the app about the field	Should specify the alert options and levels / Can take better decisions based on the app / Can save the crops from excess water and harvest it at right time
3 Feelings What your user might be thinking and feeling at the moment 👍	Will it be affordable / Will it provide more yield	Will the installation be easy ? / Will it be flexible and can be modified later	How accurate the result is produced / Ability to respond in real-time	Gives more yield / Feeling safe about the crops
👎	Maintaining the system	No error should occur while installing / Potential damages to system.	Sustainability of the system	Lack of trust and transparency on the data given by analysis
4 Pain points Problems your user runs into	Other farmers will not support / No knowledge about the technology	Installation will take time. / Missing parts while installing	Internet connectivity should be proper	Will all the farmers accept the prediction / Damage to the crops without alert
5 Opportunities Potential improvements or enhancements to the experience	Increase the technological awareness along the process	Seek professional help for installation	Increase the agility of the process and boost productivity / Feedback from farmers to improve the efficiency	Increase quality product and optimise utilisation

FUNCTIONAL REQUIREMENT

Functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy and effective to use Easy to learn and clean UI
NFR-2	**Security**	All the details are protected from unauthorized access. Detection and identification of any malfunction of sensors.
NFR-3	**Reliability**	It gives the accurate results.
NFR-4	**Performance**	It requires low power consumption and low data transmission rates.

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN

FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login with Email Id and password Check Credentials.
FR-4	Sensor Data	To display all the data from different sensors like temperature sensor, humidity sensor etc.
FR-5	Weather Info	To show the climatic weather conditions of the location in real-time and other data using a weather API.
FR-6	Calendar and Notes	Add information about the plants and add remainder in the calendar.
FR-7	Alerts and Notification	Checks for any abnormalities and alerts the user.
FR-8	Logout	After checking all the details, user can exit.

Non-functional Requirements:

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

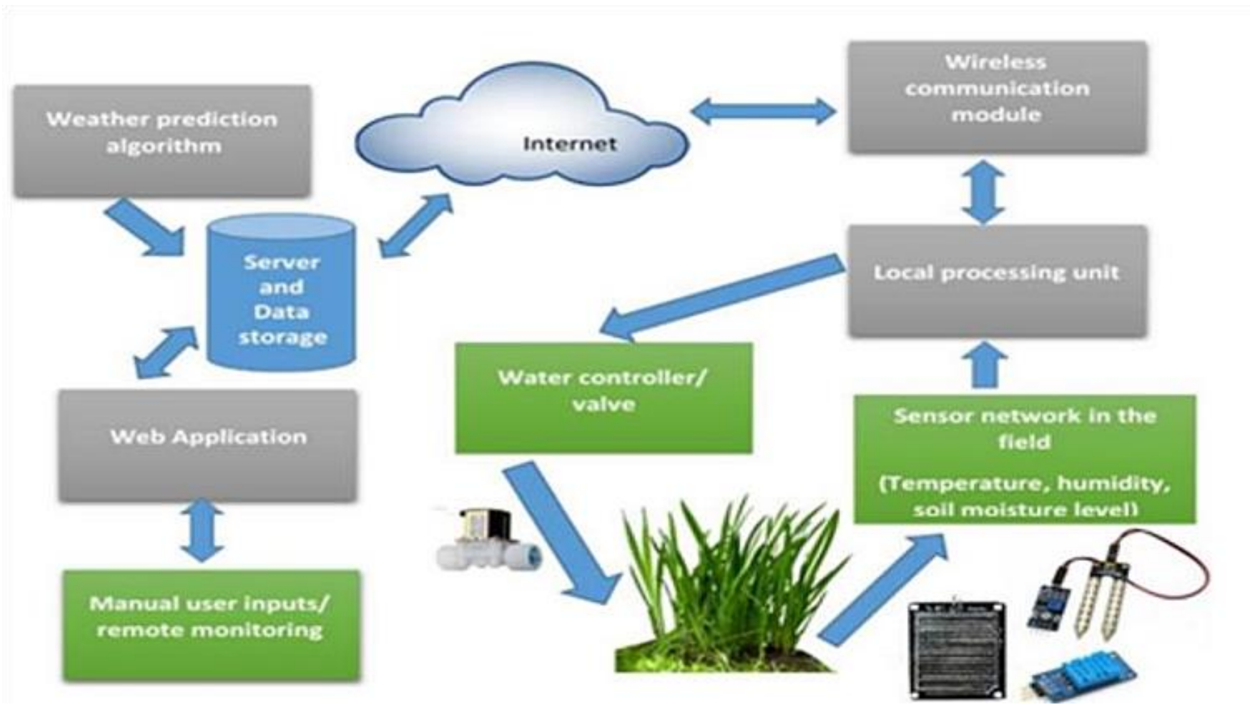
NFR-5	**Availability**	With minimum internet connectivity it is accessible all time and all the data are synced to the cloud.
NFR-6	**Scalability**	It supports third party sensors and can be easily scalable for large scale farming.

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 enter and leaves the system, what changes the information, and where data is stored.

Example: DFD Level 0 (Industry Standard)


Weather prediction algorithm
Internet
Wireless communication module
Server and Data storage
Local processing unit
Water controller/ valve
Web Application
Sensor network in the field (Temperature, humidity, soil moisture level)
Manual user inputs/ remote monitoring

USER STORIES

Use the below template to list all the user stories for the product.

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	Medium	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-1
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user I want to see everything in single widget		Medium	Sprint-2

		USN-7	As a user I want a organised widgets section		High	Sprint-2
		USN-8	As a user I want a graphical/pictorial representation		Low	Sprint-2
Customer (Web User)	Dashboard	USN-9	As a user I want a graphical representation of data for better understanding		High	Sprint-2
		USN-10	As a user I want to see a dashboard where I can customise myself	Dashboard with customisation	Low	Sprint-2
Customer (Mobile and Web)	IoT Device Setup	USN-10	Have to use a least sensor and get better output		High	Sprint-2
		USN-11	As a user, I need a low cost IoT devices for farming		High	Sprint-2
		USN-12	As a user, I need a multiple sensors for various data		High	Sprint-2
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Release**
Customer Care Executive	User Problems	USN-13	As a user, I don’t how to use the application	Manual guide will be there	Medium	Sprint-3
		USN-14	As a user, I need my application to work on most of the mobiles		High	Sprint-3
		USN-15	As a user, I am facing issue in the application	Query form will be there	High	Sprint-3
Administrator	Query Clarification	USN-16	As a admin, I give solutions to their queries		High	Sprint-3
	Particular Access	USN-17	As a admin, I give access only to authorised person		High	Sprint-3
	Connection with IoTdevices	USN-18	As a admin, I ensure the correct working of the devices. If any problem arises it will be shared to user		Medium	Sprint-4
Customer (Mobile user)	Application	USN-19	As a user, I need to control my devices	Commands for devices	High	Sprint-4
		USN-20	As a user, I need a events for better productivity		Low	Sprint-4
		USN-21	As a user, I need a more info about plants inside a application		Medium	Sprint-4

TECHNICAL ARCHITECTURE:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

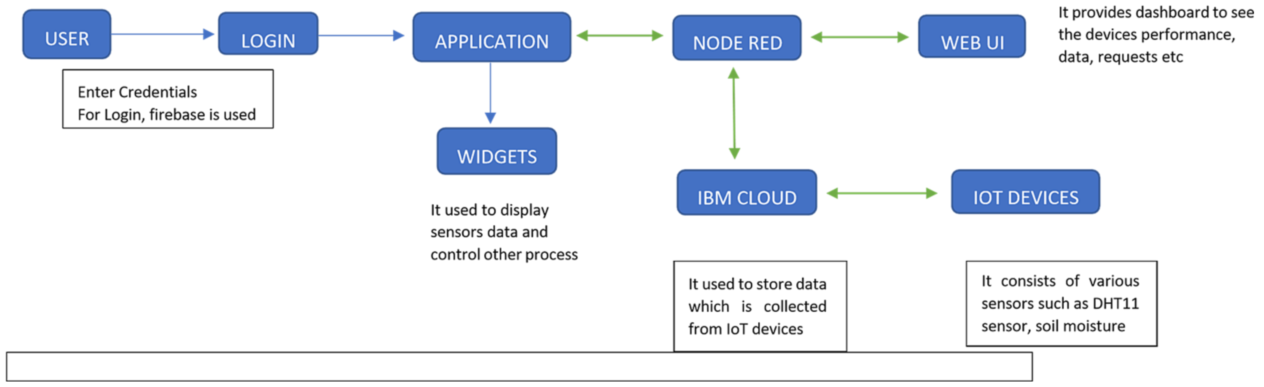

USER
LOGIN
APPLICATION
NODE RED
WEB UI
It provides dashboard to see the devices performance, data, requests etc
Enter Credentials For Login, firebase is used
WIDGETS
It used to display sensors data and control other process
IBM CLOUD
IOT DEVICES
It used to store data which is collected from IoT devices
It consists of various sensors such as DHT11 sensor, soil moisture

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Mobile app. In our application, were data are displayed using widgets like structure. Users interacts with widgets to additional info	MIT App Inventor
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data base type	Firebase is Nosql database
6.	Cloud Database	Database Service on Cloud	Firebase, IBM Watson IoT Cloud Platform
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of the API is get to weather information	Open Weather API
9.	External API-2	Purpose of the API is to connect with firebase for login purpose	Firebase API
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, IBM Cloud, Firebase

11.	DHT11 sensor, Soil Moisture sensor	It used to monitor the soil, temperature, humidity.	

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Node Red, MIT App Inventor, Arduino IDE Node Red for connecting with application, MIT App Inventor for building app, Arduino is open source electronics platform to build hardware and software.	It is a software, which helps in connecting and building application. Node Red, MIT App Inventor, Arduino IDE.
2.	Security Implementations	HTTPS Connections, X-Force Red IoT Testing	Encryptions, Secured Connection
3.	Scalable Architecture	Architecture is scalable from 10 devices to 300 devices easily and account is also scalable upto thousand connections. For very high scalability we need to upgrade our cloud plan.	Firebase, IBM Cloud
4.	Availability	Availability of our application is 24/7 because which use a cloud technology. Firebase will use commercially reasonable efforts to make Firebase available with a Monthly Uptime Percentage of at least 99.95% and distributed servers.	Firebase, IBM Cloud
5.	Performance	No of requests is 2 requests per 20 seconds or 4 requests per 30 second and sometimes user request will be added with respective to the requests	MIT App Inventor, Node Red, Cloud

PROJECT PLANNING PHASE

PREPARE MILESTONE AND ACTIVITY LIST

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	9 October 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem Statements.	9 October 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	16 November 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	16 November 2022
Problem Solution Fit	Prepare problem - solution fit Document.	16 November 2022
Solution Architecture	Prepare solution architecture Document.	

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	17 November 2022
Functional Requirement	Prepare the functional requirement document.	17 November 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	17 November 2022
Technology Architecture	Prepare the technology architecture diagram.	
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	18 November 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	19 November 2022

SPRINT DELIVERY PLAN

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

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	10	High	AMALDAS K.K ABISHEAK A NAYANA P RESHMA R

			confirming my password.			
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password.	10	High	AMALDAS K.K ABISHEAK A NAYANA P RESHMA R
Sprint-2	Cloud Services	USN-3	Creation of cloud services for collection and processing of data from the IoT devices.	20	High	AMALDAS K.K ABISHEAK A NAYANA P RESHMA R
Sprint-3	Dashboard	USN-4	Creation of a web application to monitor the data from the field through cloud.	10	High	AMALDAS K.K ABISHEAK A NAYANA P RESHMA R
Sprint-3	Mobile App	USN-5	Creation of a mobile application to integrate it with cloud services and fast SMS service.	10	High	AMALDAS K.K ABISHEAK A NAYANA P RESHMA R
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-4	Setup and Installation of devices	USN-6	IoT devices will be installed in the field and the necessary setup will be done.	20	High	AMALDAS K.K ABISHEAK A NAYANA P RESHMA R

Use the below template to create product backlog and sprint 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		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		19 Nov 2022

Project Tracker, Velocity & Burndown Chart

Velocity:

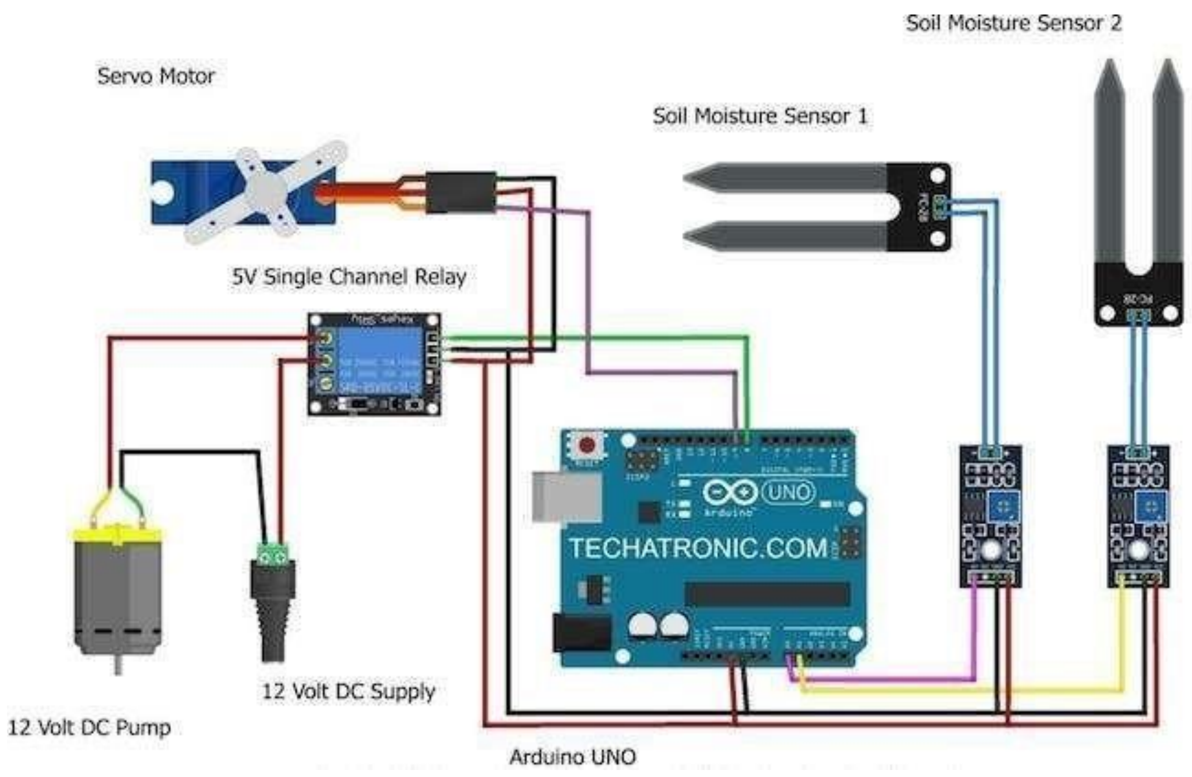
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint).
Let’s calculate the team’s average velocity
(AV) per iteration unit (story points per day)

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

PROJECT DEVELOPMENT PHASE

SPRINT 1

```
#include <Servo.h> Servo
myservo; int m=0; int n=0; int
pos = 0; void setup() {
pinMode(A0, INPUT_PULLUP);
pinMode(A1, INPUT_PULLUP);
pinMode(8,OUTPUT);
Serial.begin(9600);
myservo.attach(9);
digitalWrite(8, HIGH);
} void loop() { int m=
analogRead(A0);
int n= analogRead(A1);
Serial.println(m);
delay(10);
Serial.println(n);
delay(200); if (m>=980)
{ myservo.write(90);
digitalWrite(8, LOW);
delay(1000); } else
if(m<=970) {
digitalWrite(8, HIGH);
} if (n>=980)
{myservo.write(0);
digitalWrite(8, LOW);
delay(1000); } else
if(n<=970)
{ digitalWrite(8, HIGH);
} else{ digitalWrite(8,
HIGH);
}
}
```



SPRINT 2:

This Smart Irrigation System is used to help farmers in the irrigation process. The System provides data on the parameters which can be used to monitor the condition of the field to maintain and protect the crops. The parameters like temperature, soil moisture, the water level in the field, etc., can be accessed through the system. The sensors in the system monitor the parameters and provide them to the farmer to take the necessary measures.

Program:

```
 #include <Adafruit_LiquidCrystal.h> //Includes the library for LCD Display
#include <Wire.h>         //Includes the library for connections
#include <Servo.h>        //Includes the library for Servo Motor
```

```
}

 Servo  s; int e
 = 4; int t = 5;
 int r = 12; int
 b = 11; int g
 = 10; int sec
 = 0; int
 Sensor = 0;
 int soil = 0;
 int
 motorPin
 = 9;
 Adafruit_LiquidCrystal lcd(0);
void setup()
 {
    Wire.begin();
    pinMode(A0, INPUT);        // Temperature Sensor  pinMode(A1,
    INPUT);     // Soil Moisture Sensor  pinMode(t, OUTPUT);       //
    Ultra sonic Trigger  pinMode(e, INPUT);       // Ultra sonic Echo
    pinMode(b, OUTPUT);       // GREEN light for LED pinMode(g,
    OUTPUT);      // BLUE light for LED  pinMode(r, OUTPUT);
    // RED light for LED  pinMode(motorPin, OUTPUT);  // DC
    motor
     s.attach(3);                                  // Servo Motor
    lcd.begin(16, 2);         // LCD 16x2 Display
    lcd.setBacklight(0);  Serial.begin(9600);
 }
```

```
float readDistanceCM()
{
   digitalWrite(t, LOW);
   delayMicroseconds(2);
   digitalWrite(t, HIGH);
   delayMicroseconds(10)
   ;  digitalWrite(t,
   LOW); int duration =
   pulseIn(e,  HIGH);
   return duration * 0.034
   / 2;
}


void loop()
{


   // Soil Moisture:
   Sensor = analogRead(A1);
// Reads data from Soil Moisture            sensor
   soil = map(Sensor, 0, 1023, 0, 117);
// Low analog value indicates HIGH moisture level and High analog value
indicates LOW moisture level
   // data = map(analogValue,fromLOW,fromHIGH,toLOW,toHIGH)
   Serial.print("Soil Moisture value:");
   Serial.println(soil);
   //'data = 0' indicates total wetness and 'data = 100' indicates total dryness


   // Temperature:
   double a = analogRead(A0); // Reads data from Temperature sensor double
   t = (((a / 1024) * 5) - 0.5) * 100;
   Serial.print("Temperature value:"); //Temperature value in Celsius
   Serial.println(t);


   // Ultrasonic sensor: float distance = readDistanceCM(); //Reads data from
   Ultrasonic sensor
   Serial.print("Measured distance: ");
   Serial.println(readDistanceCM());


   // LCD Display:
   lcd.setBacklight(1);  //ON the background light in LCD lcd.clear();


   // Conditions:


   /*If the temperature is Greater than 20 and less than 35 and also the moisture
of soil is less than 60 then the GREEN light will be turned ON indicating the
Normal condition */ if (t >= 20 && t < 35 && soil >=
   40 && soil < 50)
   {
    digitalWrite(b,                 0);
    digitalWrite(g,                 1);
    digitalWrite(r, 0); s.write(90);
    digitalWrite(motorPin,
    HIGH); lcd.setCursor(3, 0); lcd.print("ON
    MOTOR");
    delay(1000); lcd.clear();
    Serial.println("Water Partially Flows");
  }
```

/*If the temperature is Greater than 35 and less than 45, then the BLUE light will be turned ON indicating the Intermediate risk condition due to slightly warm weather */ **else if (t >= 35 && t < 45)**

```
  {

    digitalWrite(b,                 1);
    digitalWrite(g,                 0);
    digitalWrite(r, 0); s.write(90);
    digitalWrite(motorPin,
    HIGH); lcd.setCursor(3, 0); lcd.print("ON
    MOTOR"); delay(1000); lcd.clear();
    Serial.println("Water Partially Flows");
  }
```

/*If the temperature is Greater than 45 or the moisture of soil is less than 30, then the RED light will be turned ON indicating the Critical condition due to highly warm weather or the low moisture content in soil */ **else if (t >= 45 || soil < 30)**

```
{  digitalWrite(b,  0);
   digitalWrite(g, 0);
   digitalWrite(r, 1);
   s.write(180);
   digitalWrite(motorPin,  HIGH);
   Serial.println("Water  Fully Flows");
   lcd.setCursor(2,    0); lcd.print("ON
   MOTOR!!!");
   lcd.setCursor(3, 1);
   lcd.print("Low
   Water"); delay(1000);
   lcd.clear();
}
```

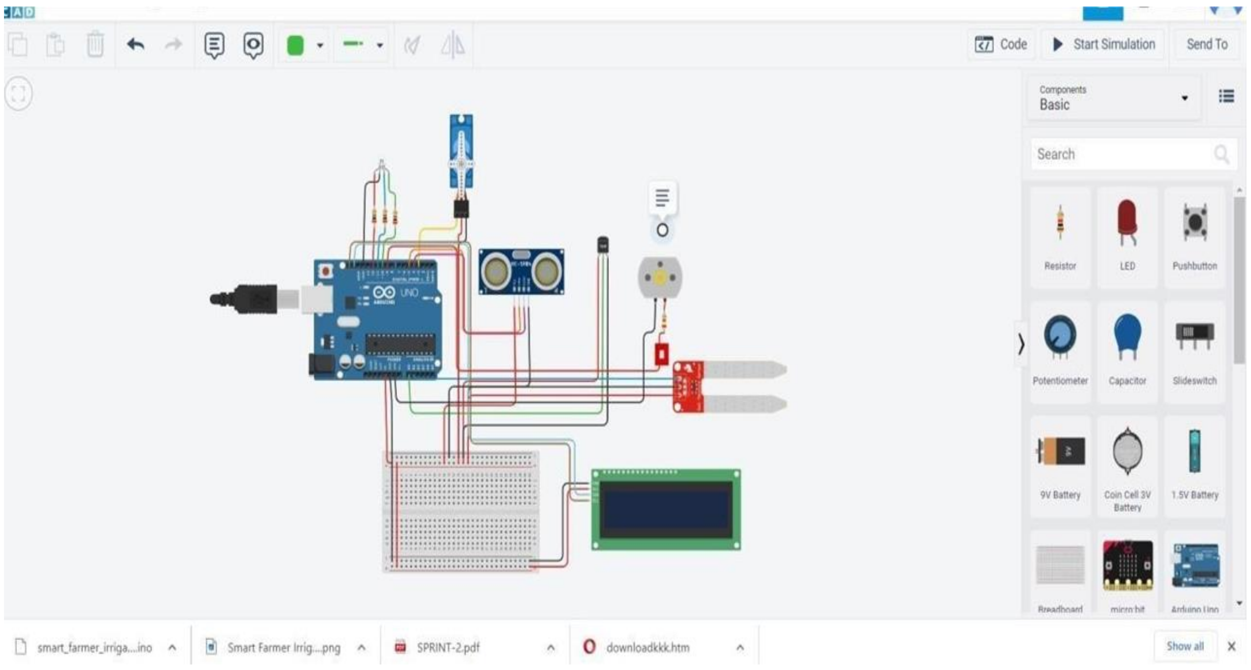
/*If the level of water is MORE in the field it will be indicated by distance sensor for less than 10cm and also the moisture of soil is greater than 80, then the YELLOW light will be turned ON indicating the high water level */ **else if (distance<10 && soil> 80)**

```
{  digitalWrite(b,  0);
   digitalWrite(g, 1);
   digitalWrite(r, 1);
   s.write(0);
   digitalWrite(motorPin,              LOW);
```

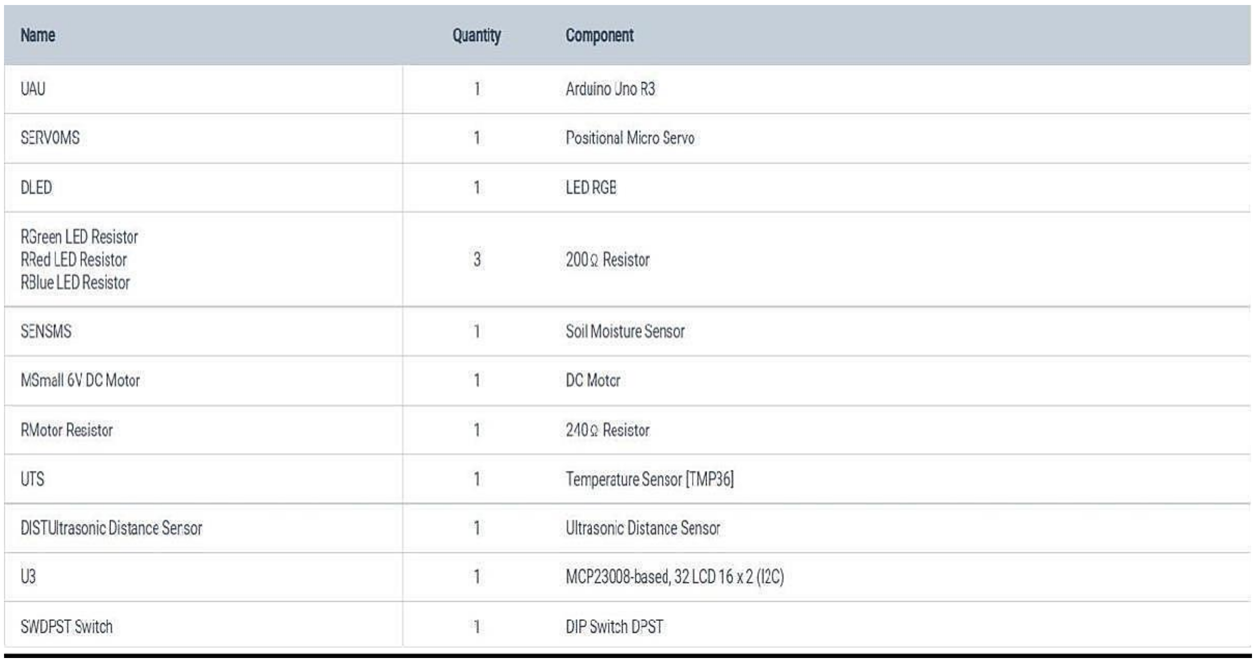
```
    Serial.println("Water Does Not Flow");
    lcd.clear();  lcd.setCursor(3,    0);
    lcd.print("OFF  MOTOR"); delay(1000);
    lcd.clear(); lcd.setCursor(1, 0);
    lcd.print("DRAIN WATER!!!");
    delay(1000); lcd.clear();
}
else
{

    digitalWrite(b,                     1);
    digitalWrite(g,                     1);
    digitalWrite(r, 0); s.write(0);
    digitalWrite(motorPin,
    LOW); lcd.setCursor(3, 0); lcd.print("OFF
    MOTOR");
    delay(1000); lcd.clear();
    Serial.println("Water Does Not Flow");
}



lcd.setCursor(0, 0);
lcd.print("Temp:");
lcd.print(t);
lcd.print("degree");
lcd.setCursor(0, 1);
lcd.print("SoilWetness:");
```

```
    lcd.print(soil);
    lcd.print("%");


    Serial.println("        ---------------------------------------------------     ");
    delay(1000);
}
```

CIRCUIT DIAGRAM:



COMPONENT USED:



Name	Quantity	Component
UAU	1	Arduino Uno R3
SERV0MS	1	Positional Micro Servo
DLED	1	LED RGB
RGreen LED Resistor RRed LED Resistor RBlue LED Resistor	3	200Ω Resistor
SENSMS	1	Soil Moisture Sensor
MSmall 6V DC Motor	1	DC Motor
RMotor Resistor	1	240Ω Resistor
UTS	1	Temperature Sensor [TMP36]
DISTUltrasonic Distance Sensor	1	Ultrasonic Distance Sensor
U3	1	MCP23008-based, 32 LCD 16 x 2 (I2C)
SWDPST Switch	1	DIP Switch DPST

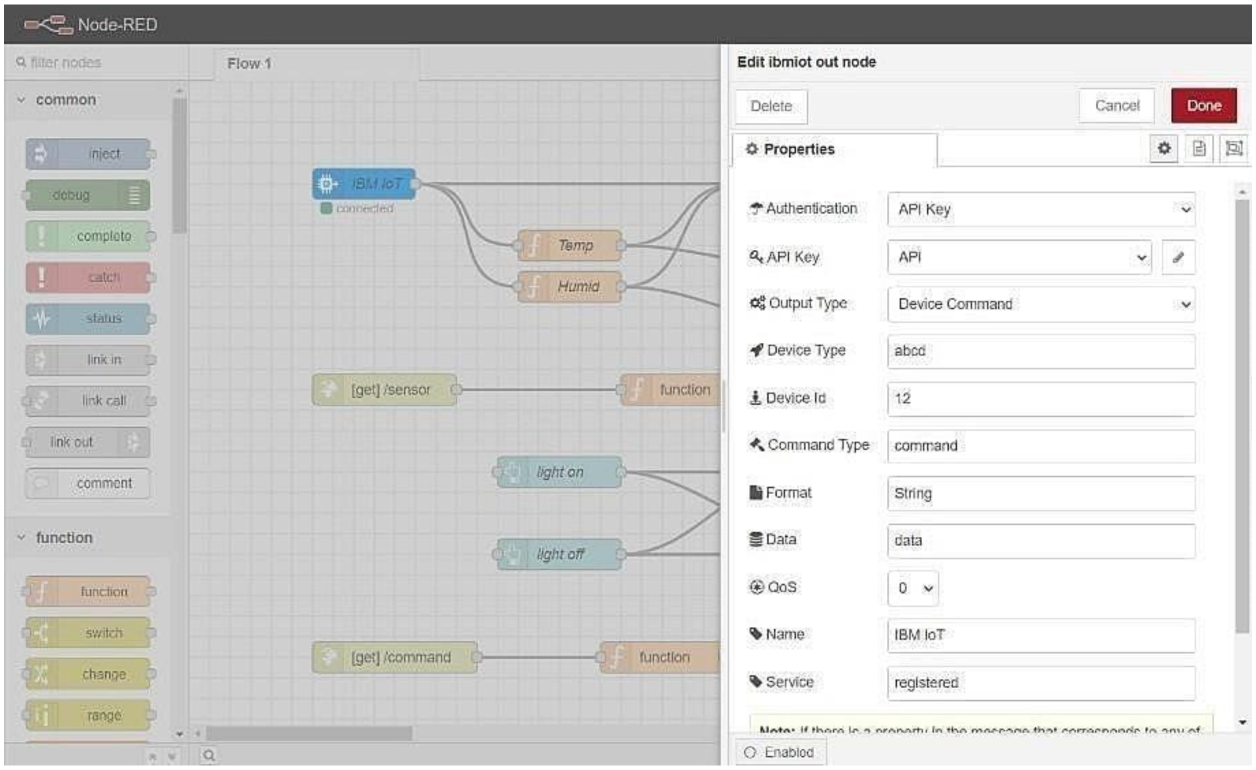
<u>SPRINT 3</u>

Configuration of Node-Red to send commands to IBM cloud

Ibm iot 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 Watson device.

Here we add two buttons in UI
1.Light On

2.Light Off



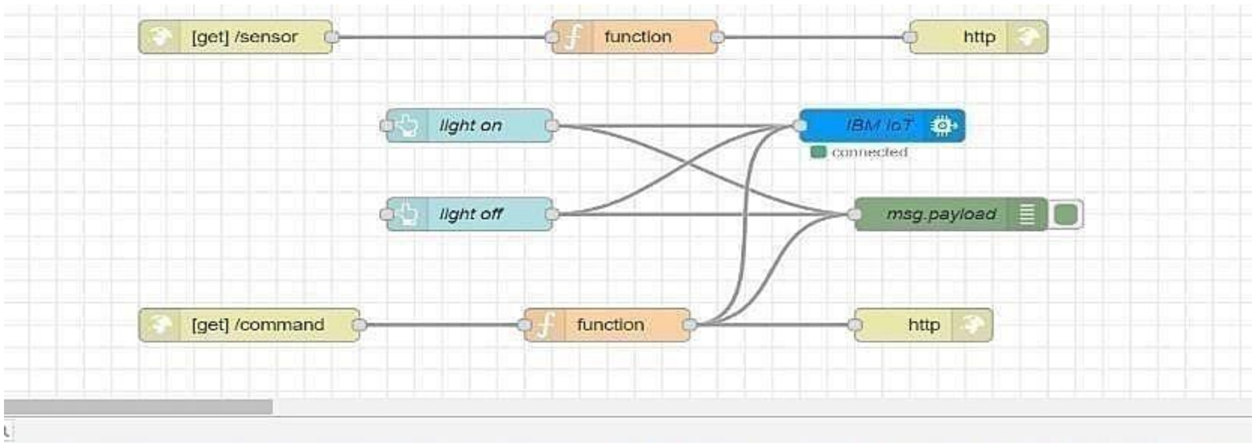
Edit ibmiot out node

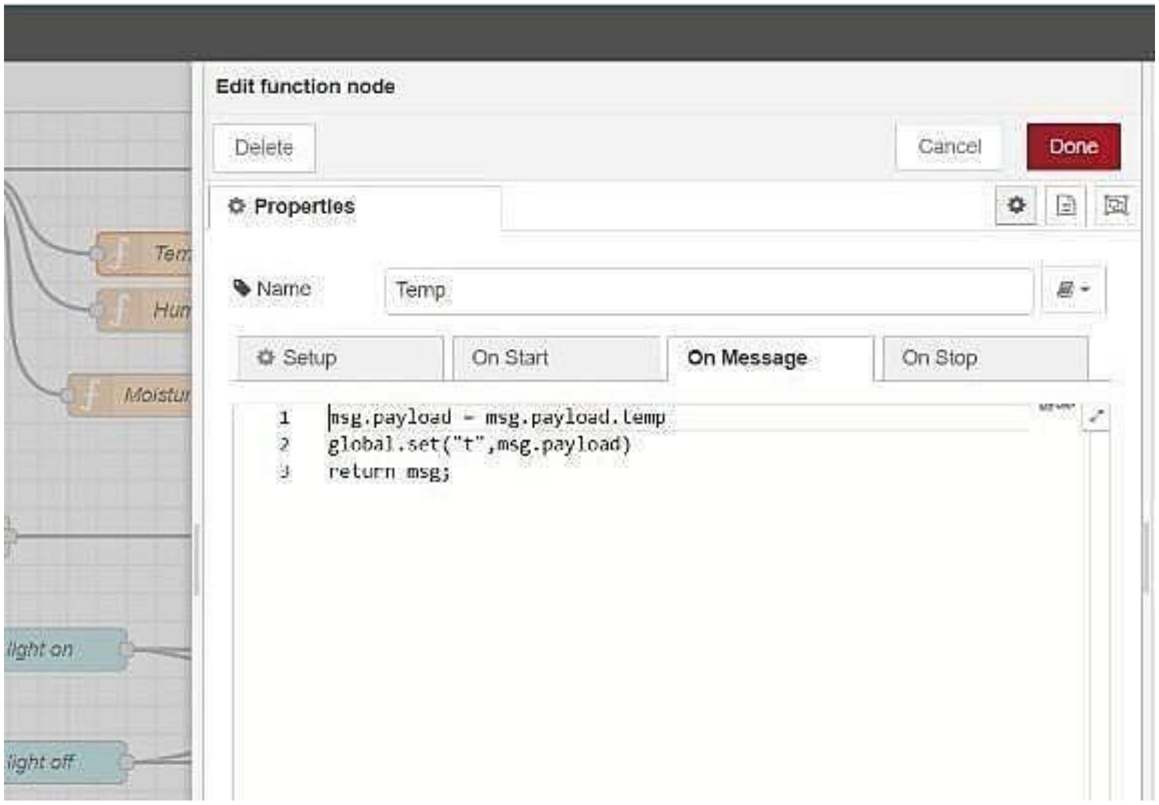
Authentication	API Key
API Key	API
Output Type	Device Command
Device Type	abcd
Device Id	12
Command Type	command
Format	String
Data	data
QoS	0
Name	IBM IoT
Service	registered

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"};
```

X

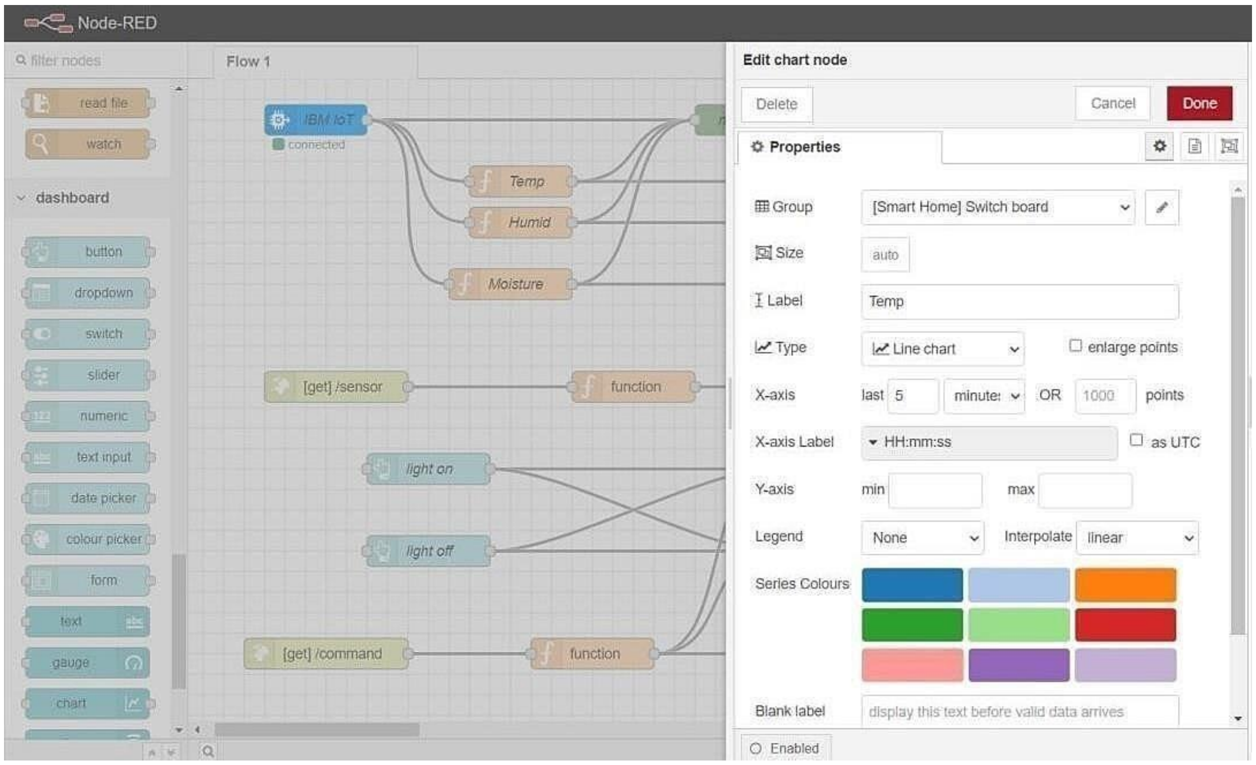




```
msg.payload = msg.payload.temp
global.set("t",msg.payload)
return msg;
```

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.



Edit chart node

Group: [Smart Home] Switch board

Size: auto

Label: Temp

Type: Line chart ☐ enlarge points

X-axis: last 5 minute: OR 1000 points

X-axis Label: HH:mm:ss ☐ as UTC

Y-axis: min max

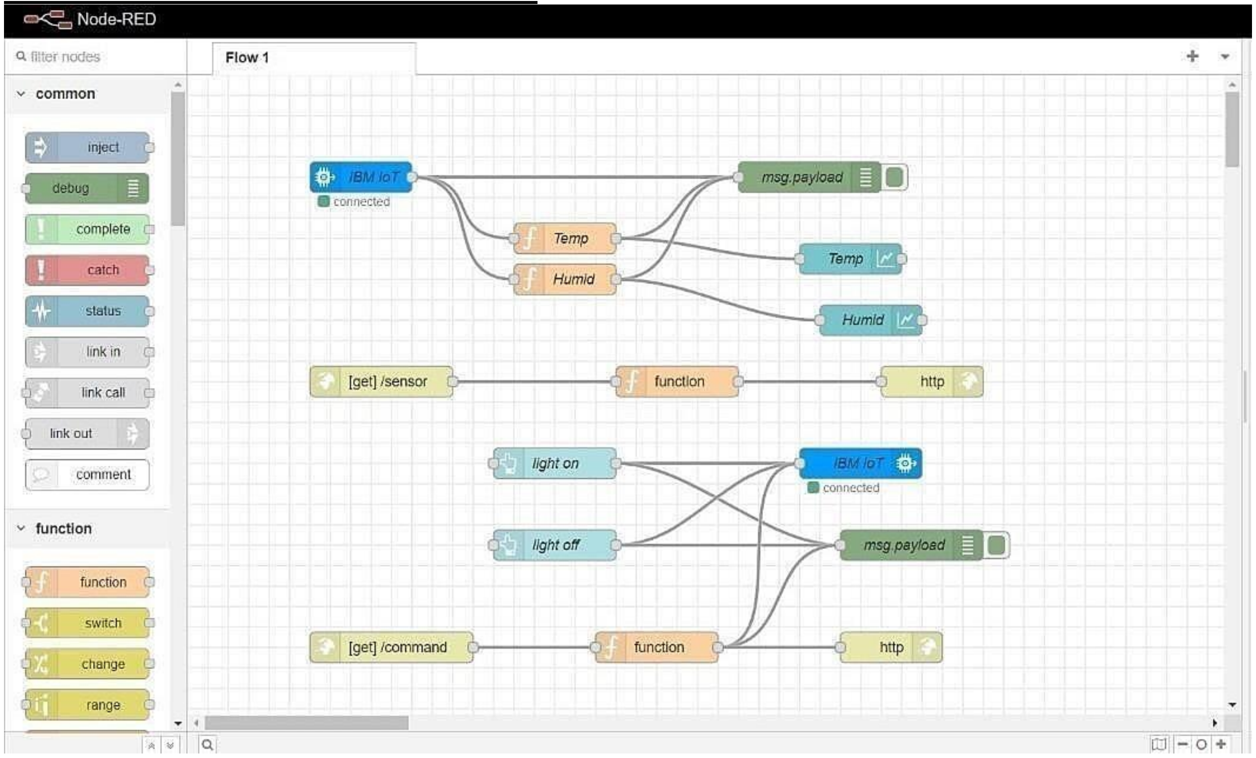
Legend: None Interpolate: linear

Series Colours

Blank label: display this text before valid data arrives

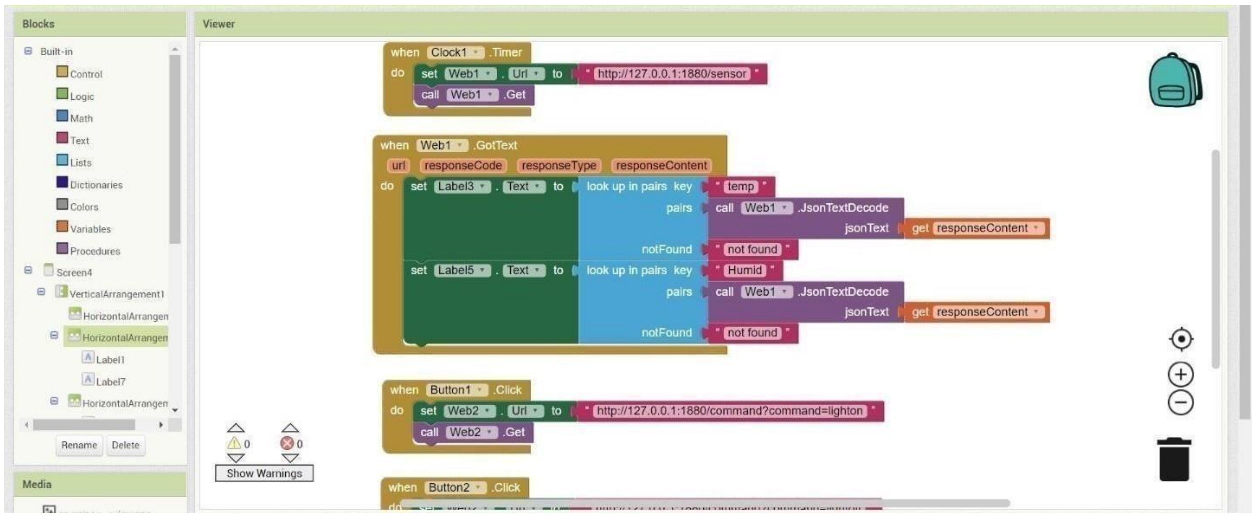
Enabled

COMPLETE PROGRAM FLOW



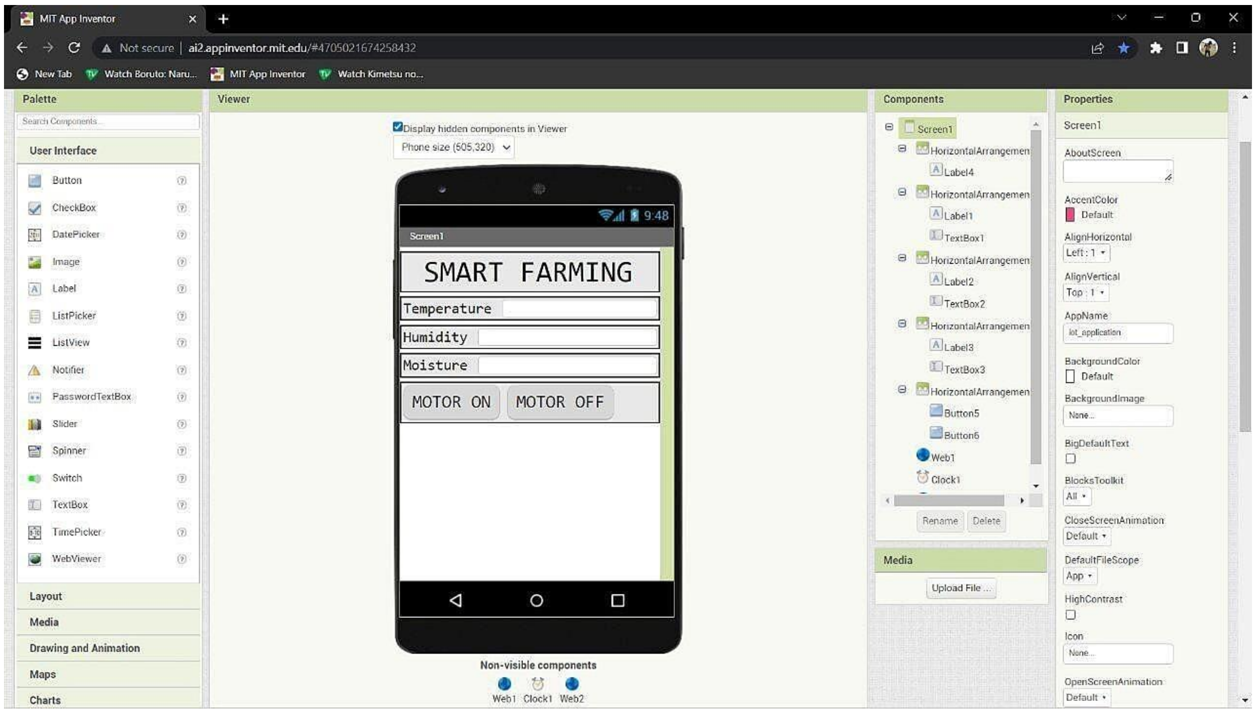
MOBILE APP WEB :

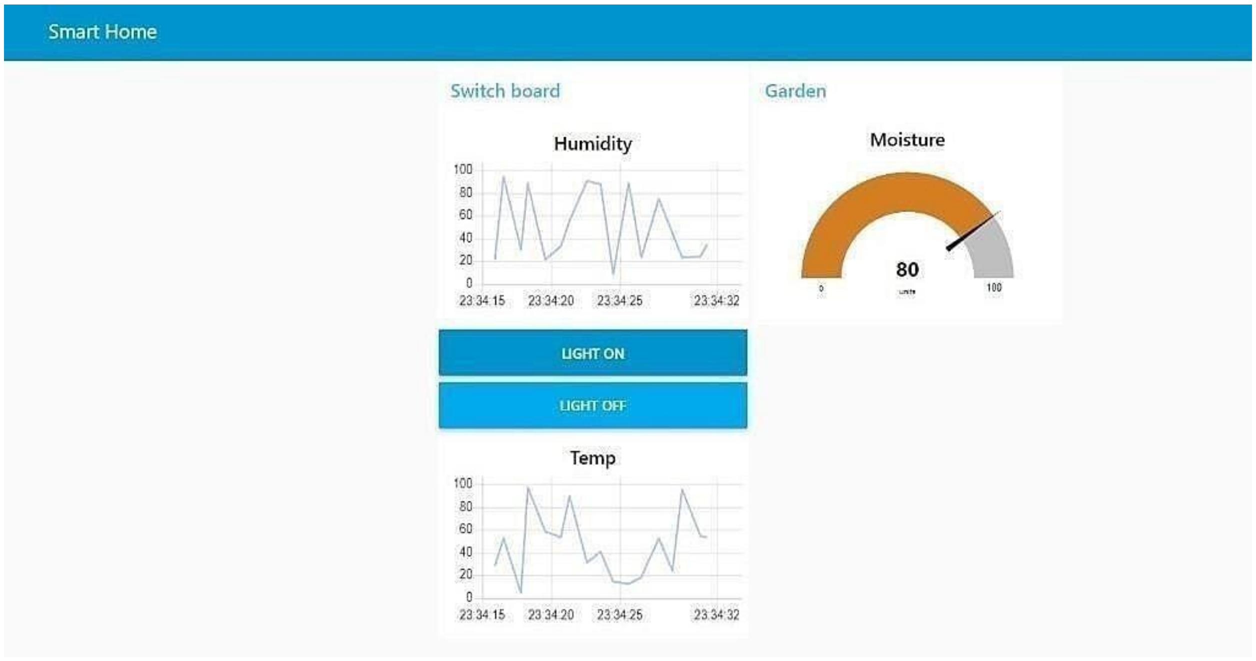
BLOCK DIAGRAM



```
when Clock1 .Timer
do  set Web1 . Url to "http://127.0.0.1:1880/sensor"
    call Web1 .Get

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set Label3 . Text to  look up in pairs  key "temp"
                                            pairs  call Web1 .JsonTextDecode
                                                        jsonText  get responseContent
                                            notFound "not found"
    set Label5 . Text to  look up in pairs  key "Humid"
                                            pairs  call Web1 .JsonTextDecode
                                                        jsonText  get responseContent
                                            notFound "not found"

when Button1 .Click
do  set Web2 . Url to "http://127.0.0.1:1880/command?command=lighton"
    call Web2 .Get

when Button2 .Click
```

SCREEN – 1





SPRINT 4

Receiving commands from IBM cloud using Python program

```
import time

importsys

import

ibmiotf.application

import ibmiotf.device

importrandom
```

#Provide your IBM Watson Device Credentials

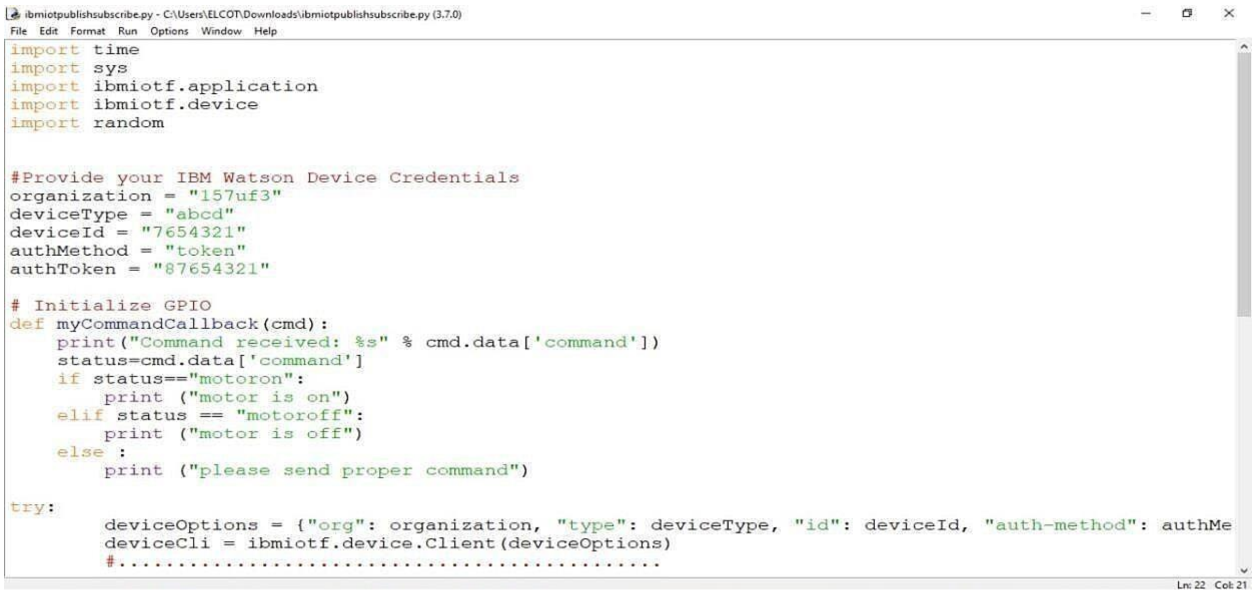
```
organization = "157uf3"

deviceType = "abcd"

deviceId = "7654321"
```

```
  authMethod = "token"

   authToken = "87654321"

# Initialize GPIO
  def   myCommandCallback(cmd):

   print("Command received: %s" %

  cmd.data['command'])

  status=cmd.data['command']

  if status=="motoron": print ("motor is on")

  elif

  status == "motoroff":    print("motor is off")

   else :
         print ("please send proper command")


  try:
              deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
  "auth-method": authMethod, "auth-token":

  authToken}deviceCli =

  ibmiotf.device.Client(deviceOptions)

            #..............................................



  except Exception as e:
        print("Caught exception connecting device: %s" % str(e))
sys.exit()
# Connect and send a datapoint "hello" with value "world" into the cloud
as anevent of type "greeting" 10 timesdeviceCli.connect()


while True:
      #Get Sensor Data from
DHT11
temp=random.randint(90,11
0)
Humid=random.randint(60,
100) Mois=random.
Randint(20,120)
   data = { 'temp' : temp,
'Humid':Humid ,'Mois': Mois}
      #print data
def
myOnPublishCall
back():
        print ("Published Temperature = %s C"% temp, "Humidity = %s
%%"%Humid, "Moisture=%s deg c" % Mois "to IBM Watson")
       success = deviceCli.publishEvent("IoTSensor", "json", data,
qos=0,on_publish=myOnPublishCallback)        if not success:
        print("Not connected to
IoTF")time.sleep(10)
```

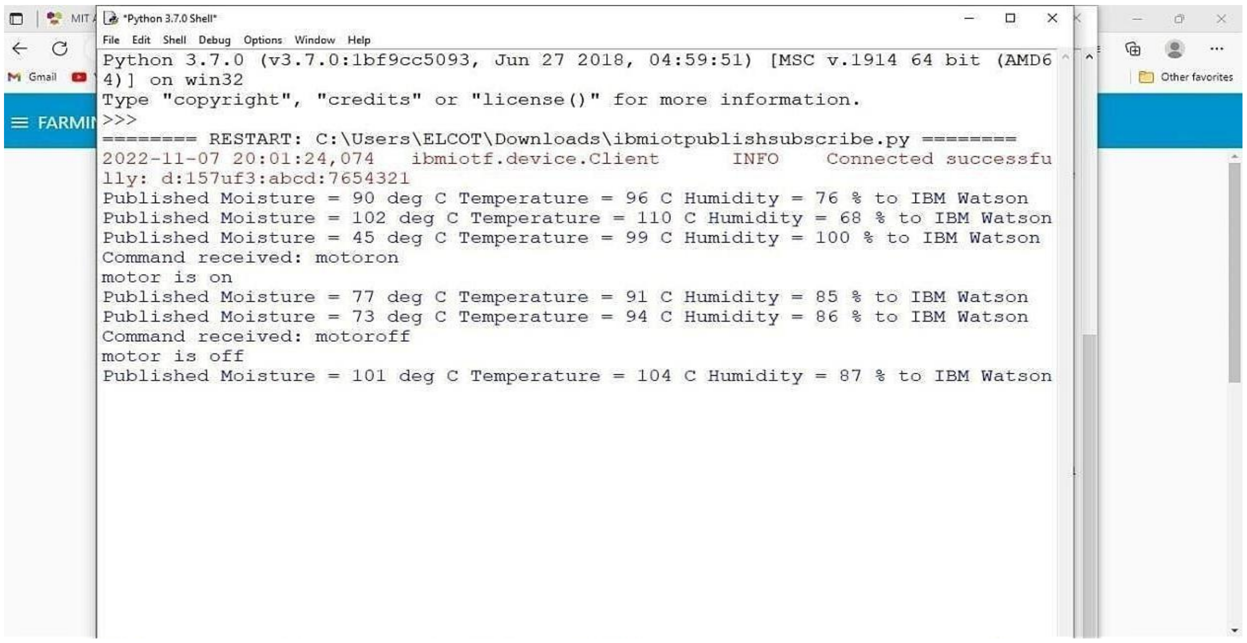
deviceCli.commandCallback =

myCommandCallback #Disconnect the device and

application from the cloud deviceCli.disconnect()

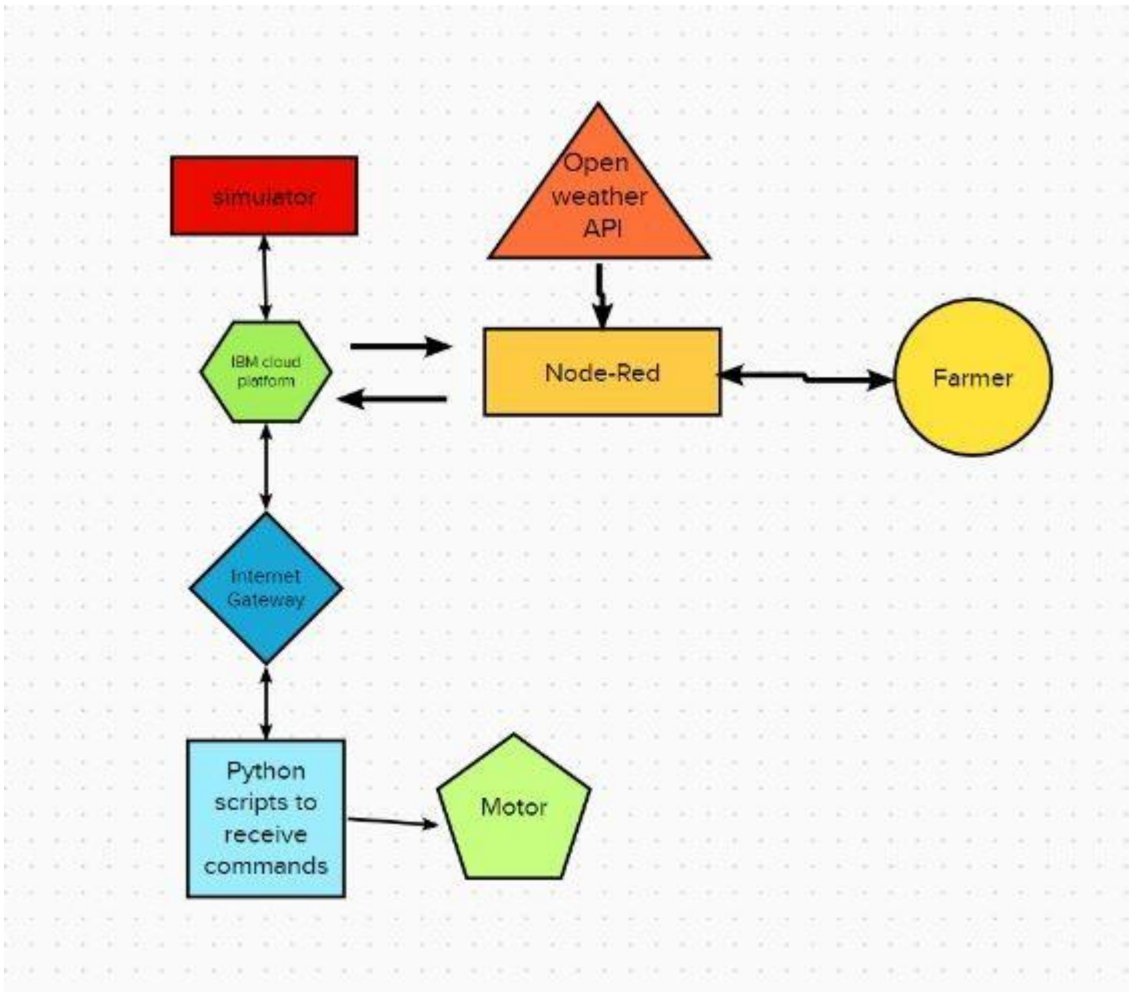


```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "157uf3"
deviceType = "abcd"
deviceId = "7654321"
authMethod = "token"
authToken = "87654321"

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

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

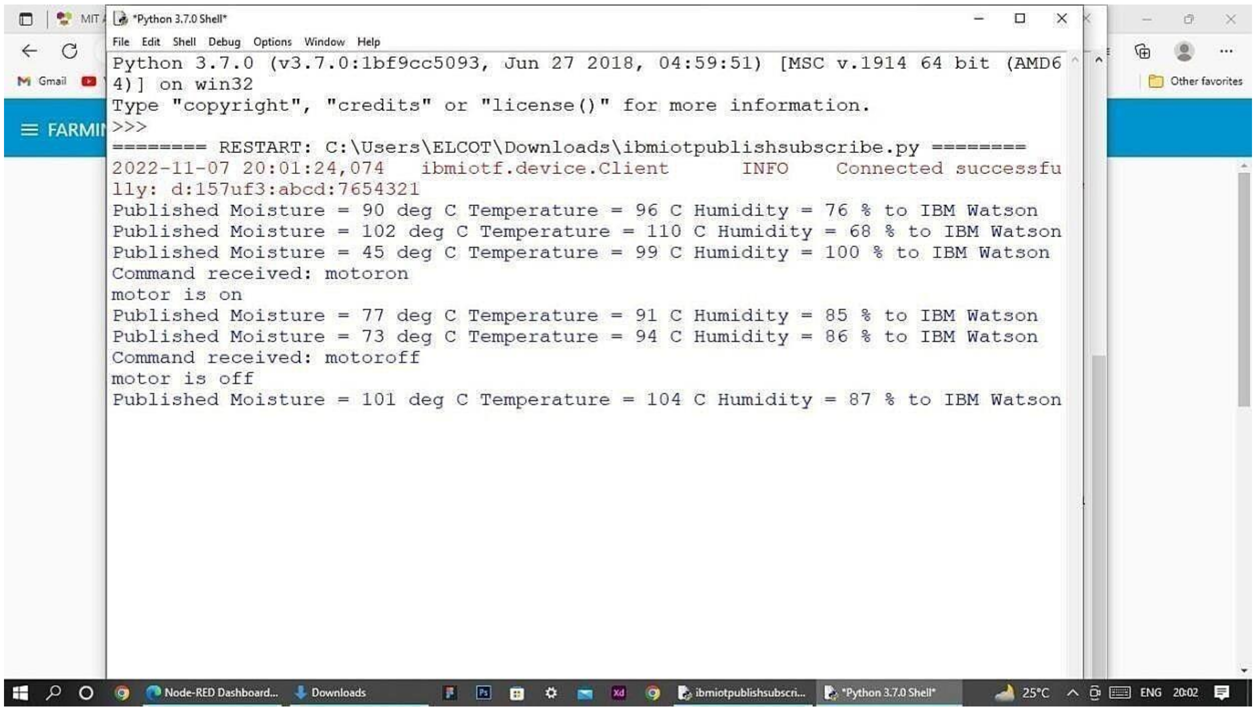


```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py ========
2022-11-07 20:01:24,074   ibmiotf.device.Client      INFO    Connected successfu
lly: d:157uf3:abcd:7654321
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 = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

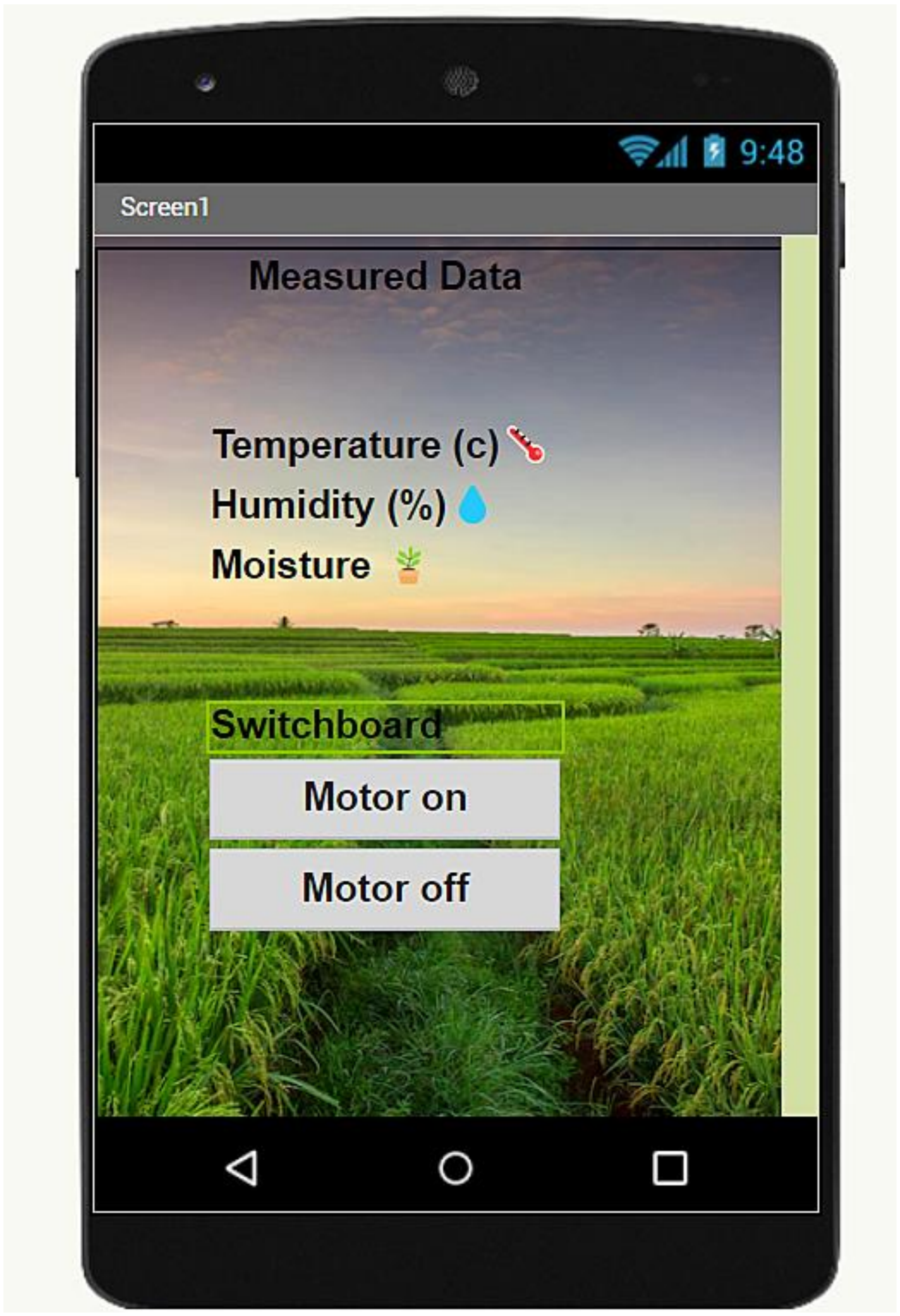
Flow Chart



Observations & Results



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py ========
2022-11-07 20:01:24,074   ibmiotf.device.Client      INFO    Connected successfu
lly: d:157uf3:abcd:7654321
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 = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```



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 Mobile App.

Conclusion

Therefore, the paper proposes a thought of consolidating the most recent innovation into the Zimbabwean farming system to turn the customary techniques for water system to current strategies in this way making simple profitable and temperate trimming. Some degree of mechanization is presented empowering the idea of observing the field and the product conditions inside some long-separate extents utilizing cloud administrations. This idea of modernization of farming is straightforward, reasonable and operable. As relying upon these parameter esteems rancher can without much of a stretch choose which fungicides and pesticides are utilized for enhancing crop creation. In Future, IoT based smart farming can also become an Intelligent IoT based smart farming system in monitoring and predicting the soil condition for irrigating the field, machine learning techniques can be employed towards crop yield and crop disease prediction and also for deciding on spraying appropriate chemicals for proper growth of crop. Lastly the data security and integrity of agricultural data can be secured while transmitting for analysis towards prediction and sending the control signal for actuation.

Thus, the objective of the project to implement an IOT system in order to help farmers to control and monitor their farms has been implemented successfully.