

# SMART FARMING APPLICATION

---

TEAM ID : PNT2022TMID29336

## Project Report

### **1.Introduction:**

#### **1.1 Project overview:**

Internet of Things Smart technology enables new digital agriculture. Today technology has become a necessity to meet current challenges and several sectors are using the latest technologies to automate their tasks. Advanced agriculture, based on [Internet of Things](#) technologies, is envisioned to enable producers and farmers to reduce waste and improve productivity by optimizing the usage of fertilizers to boost the efficiency of plants. It gives better control to the farmers for their livestock, growing crops, cutting costs, and resources.

#### **1.2 Purpose:**

We have tried to focus on different scientific applications which could be put together in the agricultural field for better accuracy with better productivity using less manpower. Moreover, we include a method for monitoring the agricultural fields from any remote location and assessing the basic condition of the field. This is the project from the motivation of the farmers working in the farmlands who are solely dependent on the rains and bore wells for irrigation of their land. In recent times, the farmers have been using irrigation techniques through manual control in which the farmers irrigate the land at regular intervals by turning the water-pump ON/OFF when required.

## **2.literature survey:**

### **2.1 Existing Problem:**

Agriculture is the foundation of our Nation. In the past, agriculturists used to figure out the ripeness of soil and influenced presumptions to develop which kind of product. They didn't think about the dampness, level of water and especially climate conditions which are more horrifying to an agriculturist. They utilize pesticides in view of a few suspicions which lead a genuine impact to the yield if the supposition isn't right. Profitability relies upon the last phase of the harvest on which agriculturists depend.

### **2.2 References:**

- [1] S. S. Gill, I. Chana, C. Science, and R. Buyya, "IoT Based Agriculture as aCloud and Big Data Service: The Beginning of Digital India," vol. 29, no. 4, pp. 1–23, 2017.
- [2] R. K. Kodali and A. Sahu, "An IoT based Weather Information Prototype Using WeMos," no. 1, pp. 612–616, 2016.
- [3] I. Mat, M. Rawidean, M. Kassim, A. N. Harun, I. M. Yusoff, and M. Berhad, "Smart Agriculture Using Internet of Things," 2018 IEEE Conf. Open Syst., pp. 54–59, 2018.
- [4] S. R. Prathibha, A. Hongal, and M. P. Jyothi, "IOT BASED MONITORING SYSTEM IN SMART AGRICULTURE," pp. 5–8, 2017.
- [5] Y. Bo and H. Wang, "The Application of Cloud Computing and The Internet of Things in Agriculture and Forestry," pp. 168–172, 2011.
- [6] Tien Cao Huang and Can Nguyen Duy, "Environment Monitoring for Agricultural Application Based on Wireless Sensor Network.," April 1617.
- [7] M. K. Gayatri, "Providing Smart Agricultural Solutions to Farmers for better yielding using IoT," no. Tiar, pp. 40–43, 2015.

- [8] S. V. Mukherji, R. Sinha, S. Basak, and S. P. Kar, "Smart Agriculture using Internet of Things and MQTT Protocol," 2019 Int. Co

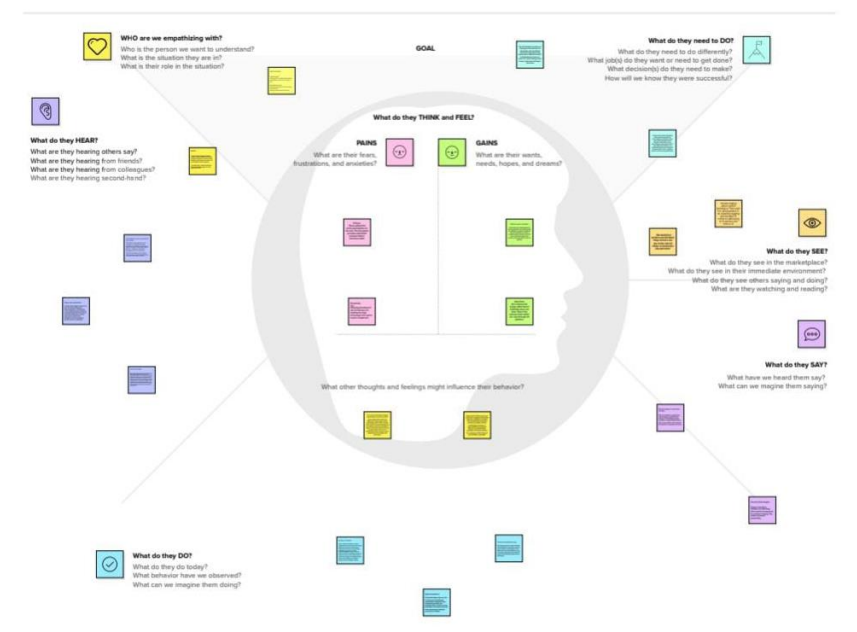
## 2.3 Problem Statement Definition:

The researcher is supposed to implement an "IoT Based Smart Farming System" with various sensors, which will help to collect the data and analyse it. The proposed system collects information about different agricultural parameters (temperature, humidity, moisture) using an IoT sensor. These values collected are then sent over the mobile via SMS. Farmers can view all the parameters required for a smart farming system through the webpage.

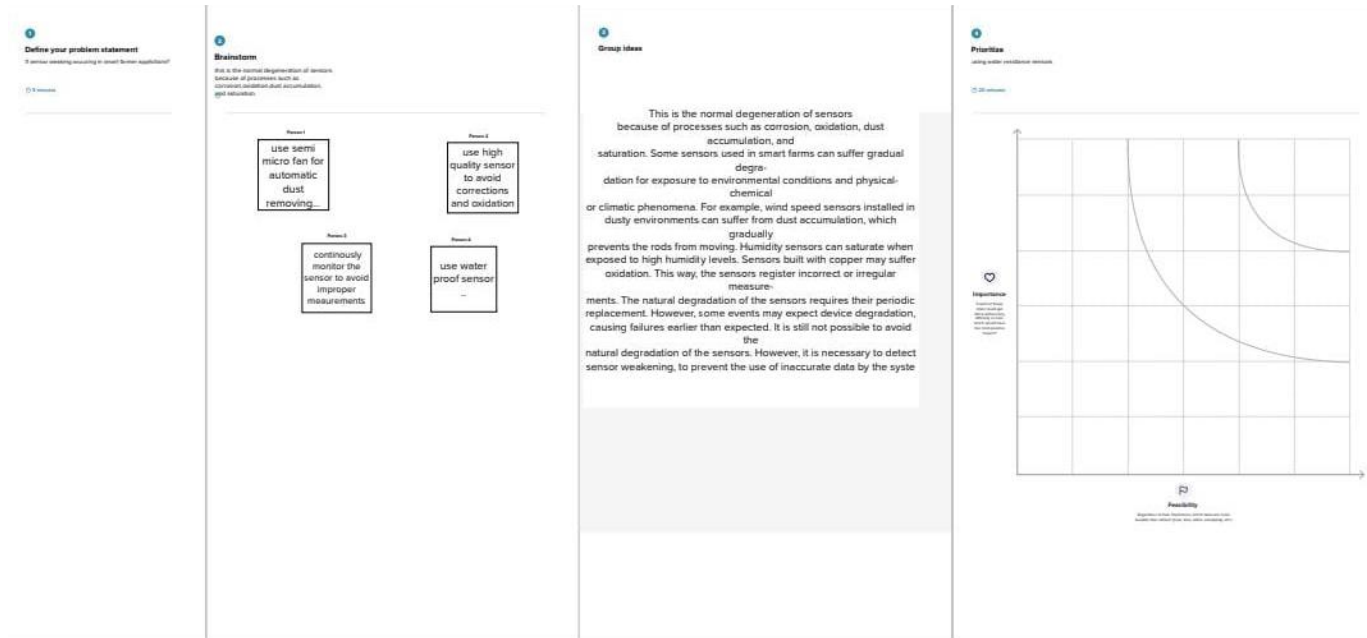
## 3. IDEATION AND PROPOSED SOLUTION:

### 3.1 Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.



## 3.2 Ideation & Brainstorming:



## 3.3 Proposed Solution:

To improve the efficiency of the product thereby supporting both rancher and country we need to utilize the innovation which appraises the nature of harvest and gives recommendations. The Internet of things (IOT) is revamping agribusiness, engaging farmers by a broad assortment of techniques, for instance, accuracy and conservative cultivation to go up against challenges in the field.

stored online using cloud computing and Internet of Things. By using wireless transmission, the sensed data is forwarded towards the web server database. If irrigation is automated, then that means if the moisture and temperature fields fall below the potential range. The user can monitor and control the system remotely with the help of an application which provides a web interface to the user.

### 3.4 Problem Solution fit:

Project Title: SMART FARMING		Project Design Phase-I - Solution Fit Template		Team ID: PNT2022TMD29336	
1. CUSTOMER SEGMENT(S) Farmers and Students are the customers.		6. CUSTOMER CONSTRAINTS High quality products on low budget ,easily affordable for everyone.		5. AVAILABLE SOLUTIONS Toll free 24/7 customer service for time saving .	
2. JOBS TO BE DONE/ PROBLEMS Sensor Weakening in smart farming applications.		6. PROBLEM ROOT CAUSE Improper maintenance of sensor and corrosion ,oxidation ,dust accumulation.		7. BEHAVIOUR Improper measurement and the speed of the device is slow.	
2. JOBS TO BE DONE/ PROBLEMS		6. PROBLEM ROOT CAUSE		7. BEHAVIOUR	
1. CUSTOMER SEGMENT(S)		6. CUSTOMER CONSTRAINTS		5. AVAILABLE SOLUTIONS	
3. TRIGGERS Early accessible and eco friendly, time saving and also AI/ARTIFICIAL INTELLIGENCE.		10. YOUR SOLUTION Use semi micro fan for automatic dust removing and use water proof sensors.		8. CHANNELS OF BEHAVIOUR ONLINE Report the problem to the company or dealers. OFFLINE Repair and cleaning the sensors.	
4. EMOTIONS: BEFORE / AFTER BEFORE: time is saved and making work smarter AFTER : Frustrated , can not make a proper monitoring.					

Identify strong TR & EM

Identify strong TR & CM

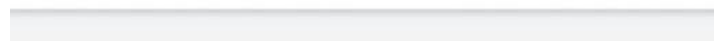
## 4. REQUIREMENT ANALYSIS

### 4.1 Functional requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User verification	Verifying and get into the application.
FR-4	User understanding	Understanding all the instruction and learning about that information. Follow the rules and instruction .
FR-5	User startup	Startup using instruction do the Process given in the application.

### 4.2 Non-Functional requirements:

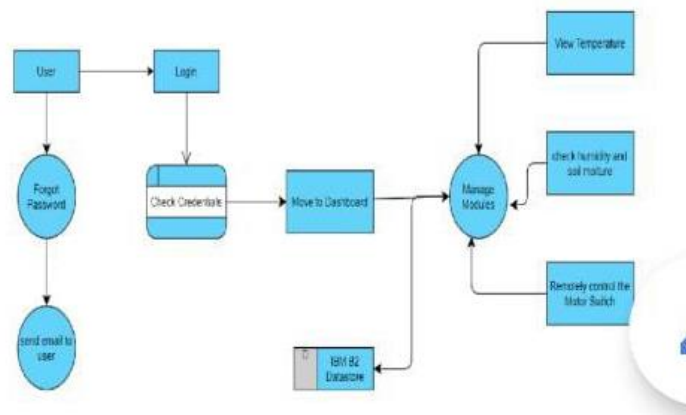
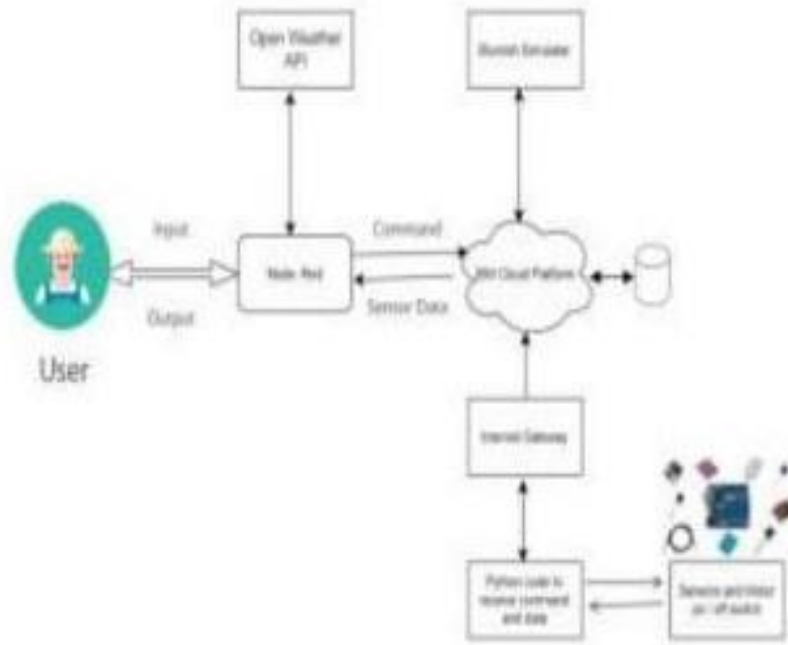
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Easy to perform service on the component Or whole system needed. It ensure technical and support staff are able to monitor the operation
NFR-2	Security	Smart devices are widely used by a range of people from farmers to entrepreneurs. These technologies are used in a variety of ways, from finding real-time status of crops and soil moisture content to deploying drones to assist with tasks such as applying pesticide spray. Multi-layered architecture relevant to the precision agriculture domain and discusses the security and privacy issues in this dynamic and distributed cyber physical environment.



NFR-3	Reliability	Reliability is essential in Smart Farming. Smart Farms' un protection may cause significant economic losses and low yields of production. This paper introduces an optimization model for providing reliability and, consequently, service continuity to the smart form
NFR-4	Performance	Performance are make or break the success of a software system or a product. User will not hesitate outright reject a system even if its meet all expected functional requirement. Availability are those that are not able to perform its regular function because of impaired and damage part.
NFR-5	Availability	There are so many application in using different tools and sensor ect....
NFR-6	Scalability	Scalability is the ability of the application to handle an increase in workload without performance , or its ability to quickly enlarge.

## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams:



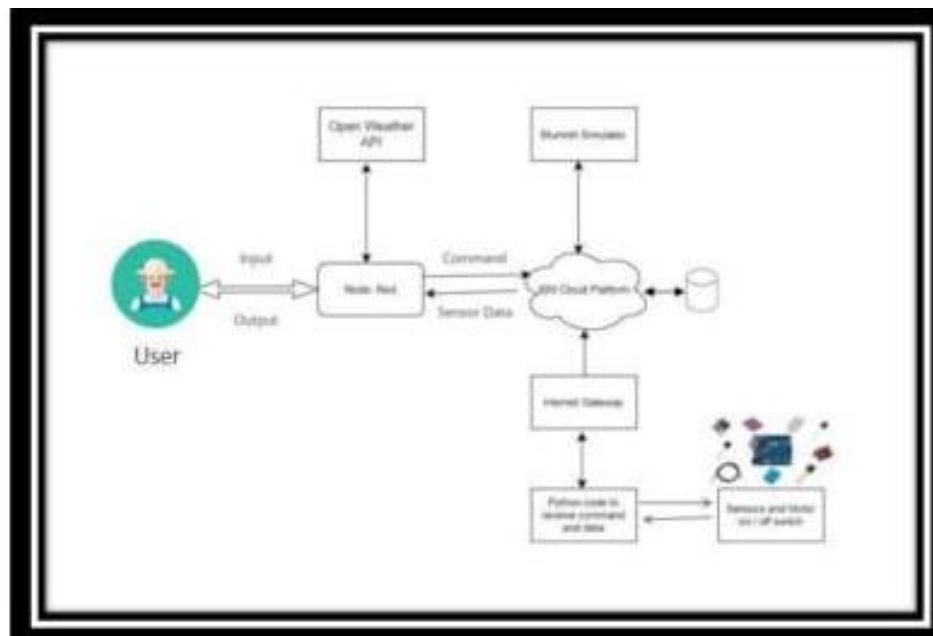
### 5.2 Solution & Technical Architecture:

- IoT-based agriculture systems help 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 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

### 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 must complete all the activities and tasks listed below:
- Create and configure IBM Cloud Services
- 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

## 5.3 User Stories:

List of all the user stories for the Smart farming application are as follows:

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		Medium	Sprint-1

			register for the application through Gmail			
	Login	USN-4	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Web user)	Dashboard	USN-5	As a User can view the dashboard, and this dashboard include the check roles of access and then move to the manage modules.	I can view the dashboard in this smart farming application system.	High	Sprint 2
		USN-6	User can remotely access the motor switch	In the smart farming app	High	Sprint 3
Administrator			As a user once view the manage modules this describes the Manage system Admins and Manage Roles of User and etc.			Sprint 2

## 6. PROJECT PLANNING & SCHEDULING6.1 Sprint

### Planning & Estimation:

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

	Functional Requirement (Epic)	User Story Number	DESIGN	Points	QUALITY	Team Members
Sprint-1	Simulation Creation	USN-1	Connect Sensors and Arduino with python code	2	HIGH	PRIYANKA.K
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	HIGH	SARUMATHI
Sprint-3	MIT APP INVENTOR	USN-3	Develop an application for the Smart farmer project using MIT App Inventor	2	HIGH	PRIYANGA.M

SPRINT-4	Web UI	USN-4	To make the user to interact with software	2	HIGH	PRIYANKA.K
----------	--------	-------	--	---	------	------------

## 6.2 Sprint Delivery Schedule:

SPRINT	TOTAL STORY POINT	NO.OF DAYS	DATE	SPRINT END DATE	STORY POINT COMPLETED	SPRINT RELEASE DATE
SPRINT 1	20	7 DAYS	30 Oct 2022	06 Nov 2022	20	29 OCT 2022
SPRINT 2	20	9DAYS	31 Oct 2022	09 Nov 2022		05 OCT 2022
SPRINT 3	20	6 DAYS	06 NOV 2022	13 NOV 2022		12 OCT 2022
SPRINT 4	20	6 DAYS	11 NOV 2022	17 NOV 2022		15 OCT 2022

### Start 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)

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

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

### PYTHON CODE:

To develop the python code to publish and subscribe to the commands from the IBM cloud.

### PROGRAM:

```
import wiotp.sdk.device
```

```
import time
```

```
import datetime
```

```
import random
```

```
myConfig={
```

```
"identity":{
```

```
"typeId": "Device1",
```

```
"deviceId": "67890"
```

```
"auth": {
```

```
ENG 12:28 26-02-2021
```

```
O HI
```

```
Lv 1 Cel 7
```

```
"orgId": "hj5fmy",
```

```
Ele Edit Format Bun Options Window Help
```

```
import os
```

```
"token": "87654321"
```

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

```

print ("Message received from IBM IoT Platform: %s" %
cmd.data['command']) m=cmd.data['command']

def myCommandCallback (cmd):

if (m=="motoron"):

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

print ("Motor is switched OFF")

print("")

while True:

soil=random.randint (0,1100) temp=random.randint (-20, 125) hum-
random.randint (0, 100)

myData={'soil moisture': soil, 'temperature':temp, humidity:hum)
client.publishEvent (event Id="status", msgFormat="json",
data=myData, qos=0, onPi print ("Published data Successfully: %s",
myData)

client.commandCallback myCommandCallback client.disconnect ()

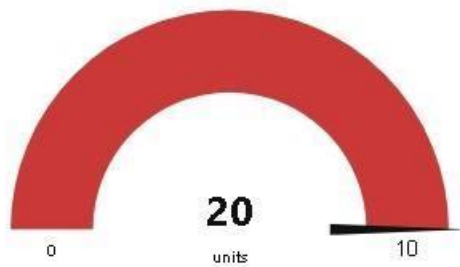
```

## 7.1 Feature 1 - SOIL MOISTURE DETECTION

IoT UI

MOTOR ON

Moisture



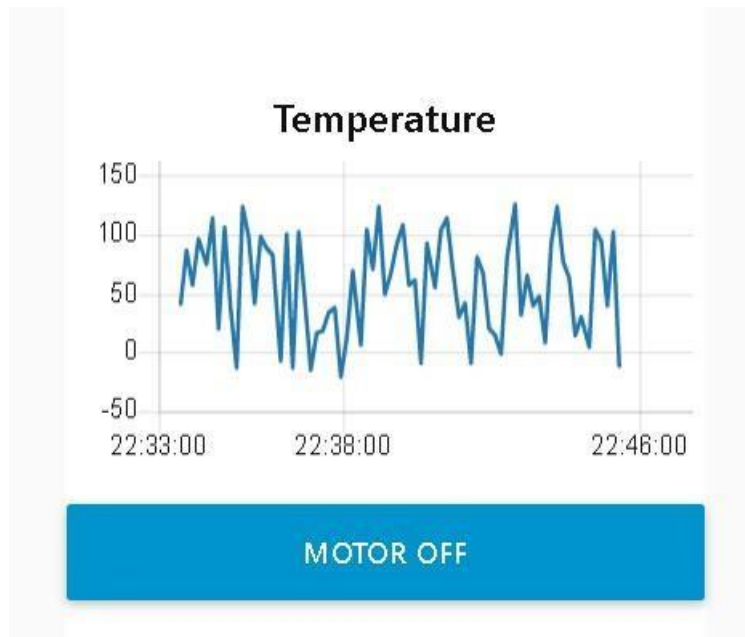
## 7.2 Feature 2 - HUMIDITY DETECTION

MOTOR OFF

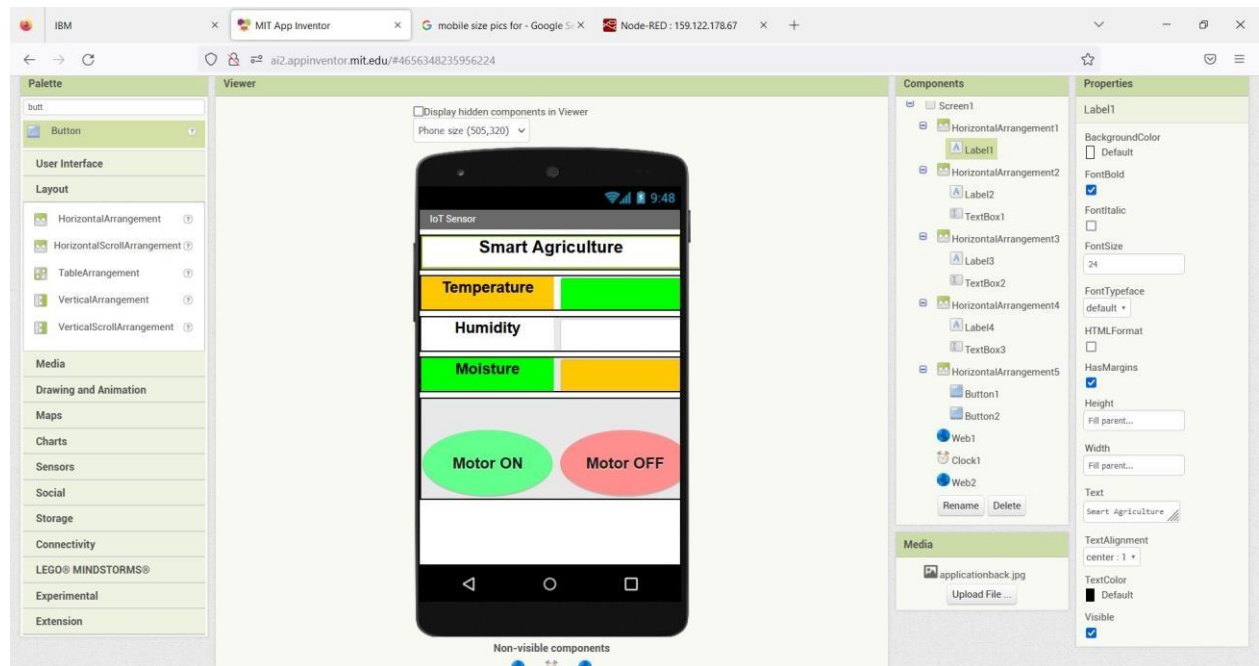
Humidity



## 7.3 Feature 3 - TEMPERATURE DETECTION



## 7.4 Feature 4 - MOTOR ON AND OFF



## 8. TESTING



## **8.1 Test Cases:**

### **1. IBM Watson IOT service:**

To create an IBM Watson IOT service Steps to create an IBM Watson IOT service: and create a device using it.

- Click on catalog in IBM cloud account.
- Click on services.
- Enter as an Internet of things platform.
- Enter region and pricing plan.
- Enter service name and click create.
- Click on launch.
- Then IBM Watson OT platform opens.
- Click on sign in.
- Enter IBM Id.
- Enter Password.
- Then you can access IBM Watson IOT platform.

### **Steps to create a device:**

- Click on devices in IBM Watson IOT platform.
- Choose to create a device● Enter the device type as abcd.
- Enter the device ID as 12345.
- Click next.

- Enter device credentials (optional).
- Click next.
- Enter the authentication token as 12345678.
- Click on continue.
- Click on next.
- Click finish Device is created successfully, and we can see device credential

## **2. Creation of Node – Red Service:**

To create a Node Red service.

Steps to create a Node-Red service:

- Click on catalog in IBM cloud account.
- Click on services.
- Enter the Node red service.
- Node red app opens click on get started.
- Enter app name as default.
- Enter the region as London.
- Choose a pricing plan as lite.
- Click create.

- You will be redirected to a new page.
- Click on deploy your app.
- Choose cloud foundry.
- Enter IBM API key (by clicking new+).
- Choose memory size as default.
- Enter the region as London.
- Click next.
- Click create.
- Status will be updated after creation.
- Click on App URL.
- Click next.
- Choose not recommended.
- Click next.
- You will see the Node red page.
- Go to your node red flow editor. ● In the left panel choose nodes

## **8.2 User Acceptance Testing:**



## 9. RESULTS:

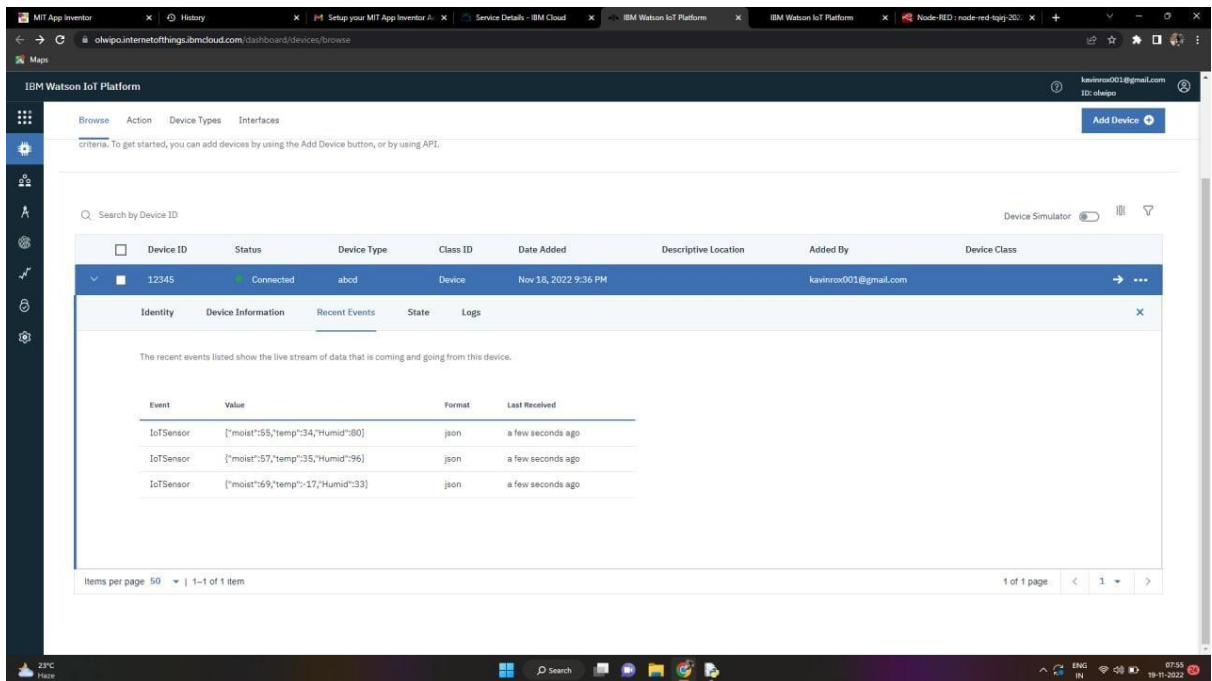
The yield appearing beneath signifies the temperature, soil moisture and humidity data received from the IoT simulator sensor and open weather API. The web app displays all this data for the past one hour. There are a set of buttons on the web application that can be used to control the motor and light on the farm to turn them ON/OFF remotely.

### 9.1 Performance Metrics:

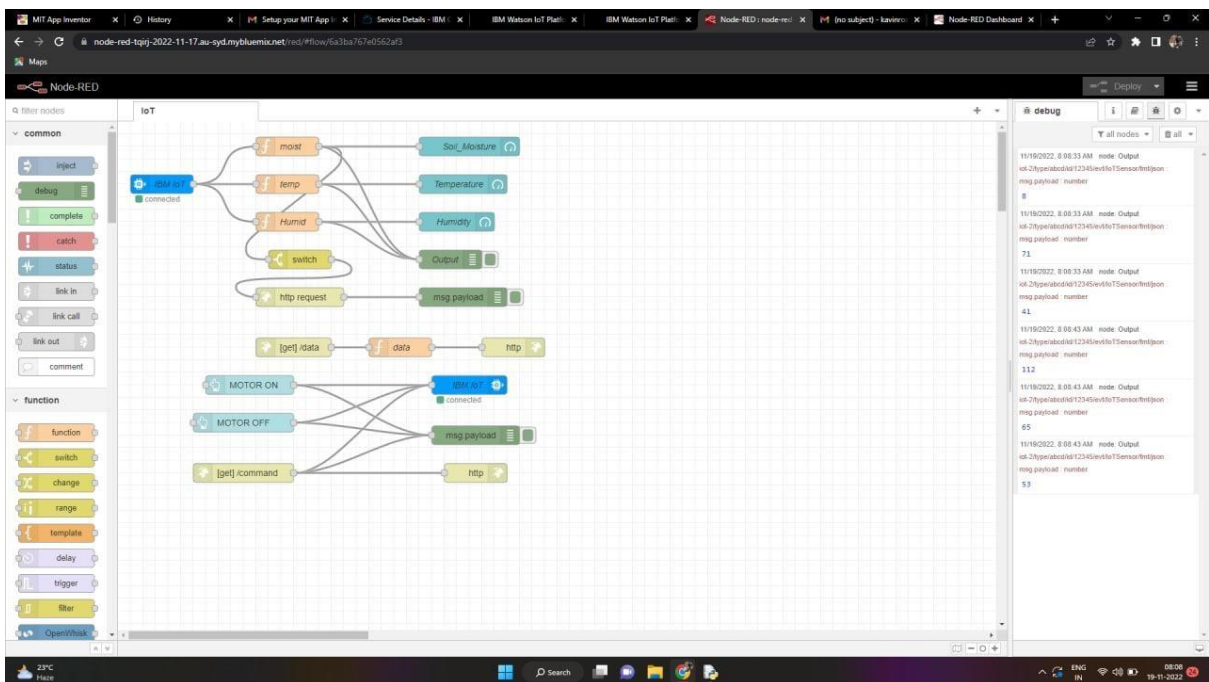
#### Python code Results:

```
Python 3.6.6 Shell
File Edit Shell Debug Options Window Help
Python 3.6.6 (v3.6.6:1f5546b7, Jun 27 2018, 03:37:03) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
----- RESTART: C:\Users\ank\Downloads\Maincodeforproject2.py -----
2022-11-19 07:12:24.087 IBMiotDeviceClient INFO Connected successfully! drcwiporahodil2345
Published moist = 34 C temp= 19 % Humid = 7 % to IBM Watson
Published moist = 28 C temp= 37 % Humid = 49 % to IBM Watson
Published moist = 62 C temp= 19 % Humid = 27 % to IBM Watson
Published moist = 98 C temp= 110 % Humid = 84 % to IBM Watson
Published moist = 96 C temp= 106 % Humid = 39 % to IBM Watson
Published moist = 50 C temp= -5 % Humid = 11 % to IBM Watson
Published moist = 54 C temp= -10 % Humid = 74 % to IBM Watson
Published moist = 24 C temp= 36 % Humid = 26 % to IBM Watson
Published moist = 30 C temp= -9 % Humid = 4 % to IBM Watson
Published moist = 70 C temp= 49 % Humid = 69 % to IBM Watson
Published moist = 2 C temp= 36 % Humid = 78 % to IBM Watson
Published moist = 35 C temp= 71 % Humid = 39 % to IBM Watson
Published moist = 58 C temp= -8 % Humid = 5 % to IBM Watson
Published moist = 36 C temp= 7 % Humid = 64 % to IBM Watson
Published moist = 32 C temp= 69 % Humid = 23 % to IBM Watson
Published moist = 16 C temp= 55 % Humid = 73 % to IBM Watson
Published moist = 91 C temp= 76 % Humid = 88 % to IBM Watson
Published moist = 19 C temp= 5 % Humid = 50 % to IBM Watson
Published moist = 53 C temp= -14 % Humid = 24 % to IBM Watson
Published moist = 77 C temp= -4 % Humid = 53 % to IBM Watson
Published moist = 93 C temp= 107 % Humid = 56 % to IBM Watson
Published moist = 65 C temp= 11 % Humid = 38 % to IBM Watson
Published moist = 77 C temp= 76 % Humid = 95 % to IBM Watson
Published moist = 59 C temp= 54 % Humid = 2 % to IBM Watson
Published moist = 3 C temp= 11 % Humid = 90 % to IBM Watson
Published moist = 29 C temp= 72 % Humid = 72 % to IBM Watson
Published moist = 6 C temp= 75 % Humid = 69 % to IBM Watson
Published moist = 89 C temp= 44 % Humid = 22 % to IBM Watson
Published moist = 61 C temp= 23 % Humid = 97 % to IBM Watson
Published moist = 15 C temp= 16 % Humid = 80 % to IBM Watson
Published moist = 22 C temp= 74 % Humid = 52 % to IBM Watson
Published moist = 26 C temp= 46 % Humid = 52 % to IBM Watson
Published moist = 49 C temp= 2 % Humid = 48 % to IBM Watson
Published moist = 4 C temp= 92 % Humid = 39 % to IBM Watson
Published moist = 62 C temp= -19 % Humid = 91 % to IBM Watson
Published moist = 25 C temp= 37 % Humid = 12 % to IBM Watson
Published moist = 64 C temp= 75 % Humid = 99 % to IBM Watson
Published moist = 34 C temp= 30 % Humid = 31 % to IBM Watson
Published moist = 79 C temp= 23 % Humid = 2 % to IBM Watson
Published moist = 37 C temp= 19 % Humid = 60 % to IBM Watson
Published moist = 41 C temp= -18 % Humid = 72 % to IBM Watson
Published moist = 47 C temp= 68 % Humid = 69 % to IBM Watson
Published moist = 53 C temp= 125 % Humid = 84 % to IBM Watson
Published moist = 11 C temp= 128 % Humid = 67 % to IBM Watson
Published moist = 31 C temp= 64 % Humid = 48 % to IBM Watson
Published moist = 54 C temp= 92 % Humid = 45 % to IBM Watson
Published moist = 17 C temp= 59 % Humid = 51 % to IBM Watson
Published moist = 53 C temp= 44 % Humid = 37 % to IBM Watson
Published moist = 27 C temp= 53 % Humid = 13 % to IBM Watson
Published moist = 4 C temp= 20 % Humid = 61 % to IBM Watson
Published moist = 68 C temp= 11 % Humid = 26 % to IBM Watson
Published moist = 34 C temp= 111 % Humid = 75 % to IBM Watson
Published moist = 43 C temp= 82 % Humid = 10 % to IBM Watson
Published moist = 71 C temp= 111 % Humid = 56 % to IBM Watson
Published moist = 66 C temp= 5 % Humid = 69 % to IBM Watson
Published moist = 25 C temp= 3 % Humid = 55 % to IBM Watson
```

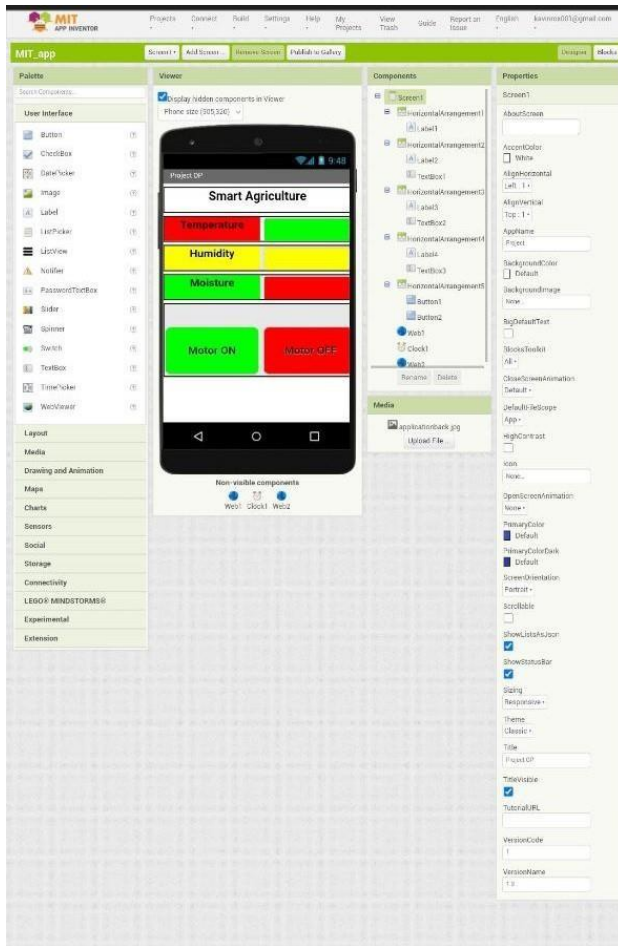
# IBM Watson-IOT Platform:



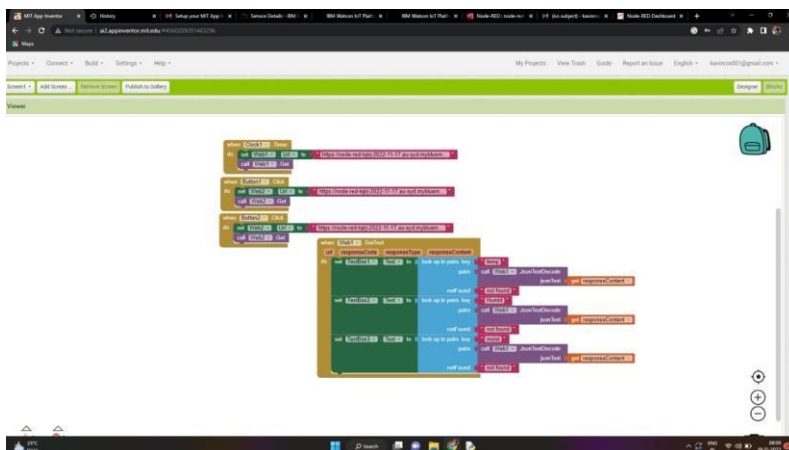
## NODE-RED Flow:



## MIT App Inventor:



## Blocks:



## **10. ADVANTAGES & DISADVANTAGES:**

### **ADVANTAGES:**

1. Communicating the device at a larger distance through web application. It will play an important role in reducing the manpower and traveling expenses of a farmer.
2. Monitoring parameters like temperature, humidity etc. will play an important role in improving the growth of the plant.
3. Integrating the weather station to the web browser will provide the details of status of the cloud, wind speed etc. It will allow the farmer to protect their plants from natural calamities.

### **DISADVANTAGE:**

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



## **11. CONCLUSION:**

IoT based SMART AGRICULTURE SYSTEM for Live Monitoring of Temperature and Soil Moisture and to control motor and light remotely have been proposed using Node Red and IBM Cloud Platform. The System has high efficiency and accuracy in fetching the live data of temperature and soil moisture. The IoT based smart farming System being proposed via this project will assist farmers in increasing the agriculture yield and take efficient care of food production as the System will always provide a helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with more than 99% accurate results. Therefore, the project proposes a thought of consolidating the most recent innovation into the agrarian field to turn the customary techniques for water systems to current strategies in this way making simple profitable and temperate trimming.

## **12. FUTURE SCOPE:**

The project has vast scope in developing the system and making it more user friendly and the additional features of the system like:

- By installing a webcam in the system, photos of the crops can be captured and the data can be sent to a database.
- Speech based options can be implemented in the system for the people who are less literate.
- GPS (Global Positioning System) can be integrated to provide specific location of the farmer and more accurate weather reports of agriculture fields and gardens.
- Regional language features can be implemented to make it easy for the farmers who are aware of only their regional language.

### 13. APPENDIX:

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

#Provide your IBM Watson Device Credentials
organization = " olwipo " deviceType = "abcd"
deviceId = "12345" authMethod = "token"
authToken =
"12345678"

# 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 an event of type "greeting" 10 times deviceCli.connect()
```

```
while True:
```

```
    #Get Sensor Data from DHT11
```

```
    temp=random.randint(90,110)
```

```
    Humid=random.randint(60,100)
```

```
    moist=random.randint(10,80)
```

```
    data = { 'temp' : temp, 'Humid': Humid, 'moist' : moist}
```

```
    #print      data def myOnPublishCallback():
```

```
    print ("Published Temperature = %s C" % temp, "Humidity = %s
%%"      % Humid, "moisture = %s %%" % moist, "to IBM
Watson")
```

```
    success = deviceCli.publishEvent("IoTSensor", "json", data,
qos=0, on_publish=myOnPublishCallback) if
not success:
```

```
        print("Not connected to IoT")
```

```
    time.sleep(10)    deviceCli.commandCallback    =
```

```
    myCommandCallback
```

```
#    Disconnect    the device    and application    from
the    cloud deviceCli.disconnect()
```

---

**GIT HUB:** <https://github.com/IBM-EPBL/IBM-Project-37567-1660312344>  
[1660312344](https://github.com/IBM-EPBL/IBM-Project-37567-1660312344)

**DEMO LINK:** <https://drive.google.com/file/d/10KIz4GyGA-IMK6YS2FN8aL1idZPWP3d6/view?usp=drivesdk>