# PROJECT REPORT

## SMART FARMER – IOT ENABLED SMART FARMING APPLICATION

## TEAM MEMBERS


**ANUKEERTHANA S – TEAM LEAD**

**ANANDHAN V**

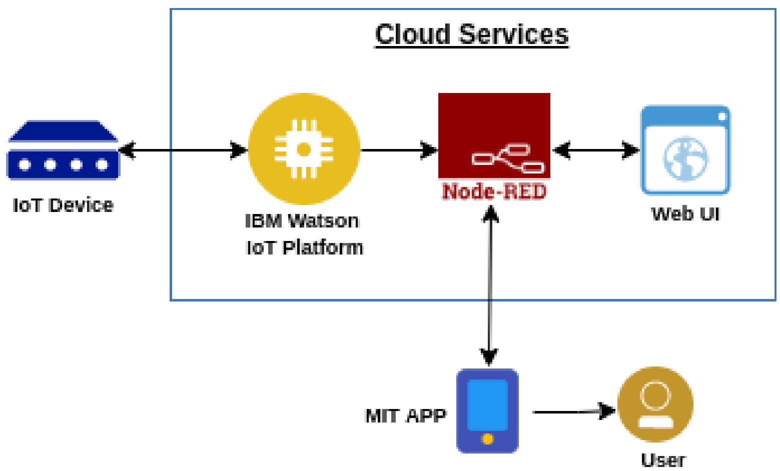
**DHANUSRIYA A S**

**MANI BHARATHI M**

## 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

The smart agriculture model main aim to avoid water wastage in the irrigation process. It is low cost and efficient system Is shown below. It includes NodeMCU, Arduino Nano, sensors like soil moisture and Dht11, solenoid valves, relays.

# 2.LITERATURE SURVEY

## 2.1 Existing problem

Smart agriculture farming system is a new idea of farming in agriculture, because which uses IoT technology to monitor the crop 24/7 and sends the information to the cloud. This emerging system increases the quality and quantity of agricultural products. IoT technology provides the information about farming fields and then takes action depending on the farmer input. In this project we can design an IoT based advanced solution for monitoring the soil conditions and atmosphere for efficient crop growth is presented. The developed system is capable of monitoring temperature, humidity, soil moisture type using Ardunio UNO R3 and different sensors connected to the microcontroller. Also, a notification is shown in farmer's phone using Wi-Fi about environmental condition and water levels of the crop field

- It is not a secure system.
- There is no motion detection for protection of agriculture field.
- Automation is not available.

## 2.2 References

**LITERATURE SURVEY**


[1] Gorli, Ravi & Yamini G. (2017). Future of Smart Farming with Internet of Things. Journal of Information technology and Its Applications. Volume 2, Issue 1, Page 27-38.

[2] M. Ryu, J. Yun, T. Miao, I. Y. Ahn, S. C. Choi and J. Kim, "Design and implementation of a connected farm for smart farming system," 2015 IEEE SENSORS,Busan, 2015, pp. 1-4.

[3] "Automated Water Irrigation System using Arduino Uno and Raspberry Piwith Android Interface", International Research Journal of Engineering and Technology (IRJET) 2018.

[4] Quy, V.K.; Nam, V.H.; Linh, D.M.; Ngoc, L.A.; Gwanggil, J. Wireless Communication Technologies for IoT in 5G: Vision, Applications, and Challenges. Wirel. Commun. Mob. Comput. 2022, 2022, 3229294.

[5] Agana, NA and Homaifar, A (2017). A deep learning based approach for longterm drought prediction. In SoutheastCon 2017 (pp. 1-8). IEEE.

[6] Jayaraman, P.P.; Yavari, A.; Georgakopoulos, D.; Morshed, A.; Zaslavsky, A. Internet of things platform for smart farming: Experiences and lessons learnt. Sensors 2016,16, 1884.

[7] C.B. Flora, Food security in the context of energy and resource depletion: sustainable agriculture in developing countries, Renew. Agric. Food Syst. 25 (2) (Jun.2010) 118–128.

[8] M. Pérez-Ortiz, P. A. Gutiérrez, J. M. Peña, J. Torres-Sánchez, F. LópezGranados and C. Hervás-Martınez, "Machine learning paradigms for weed mapping via ́ unmanned aerial vehicles," 2016 IEEE Symposium Series on Computational Intelligence (SSCI),Athens, 2016, pp. 1-8.

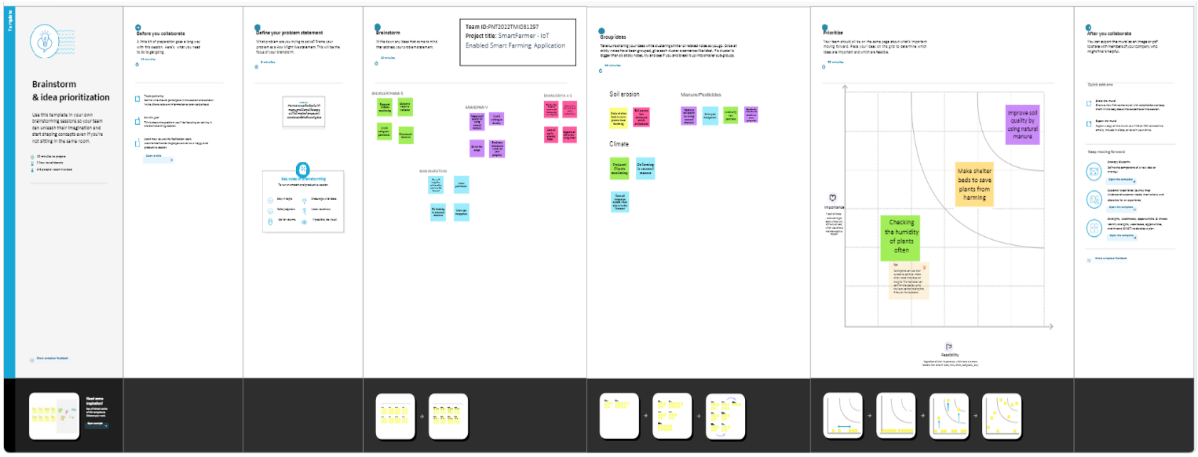
**2.3 Problem Statement Definition**

The soil moisture sensor measures wetness content in the soil. The Arduino UNO microcontroller used to receive input from a various sensor and it can be controlled automatically. When soil moisture sensor goes low the water pump will be on and it exceeds defined levels of the water motor will turn off automatically. We can constantly monitor the growth of a crop using ultrasonic sensor. PIR sensor detects the motion or unusual movement in the agricultural land. This device his very helpful to the former to monitor and control environmental parameters at their field. The farmers did not go to their field, they can remotely monitor and control using cloud

## 3.IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas

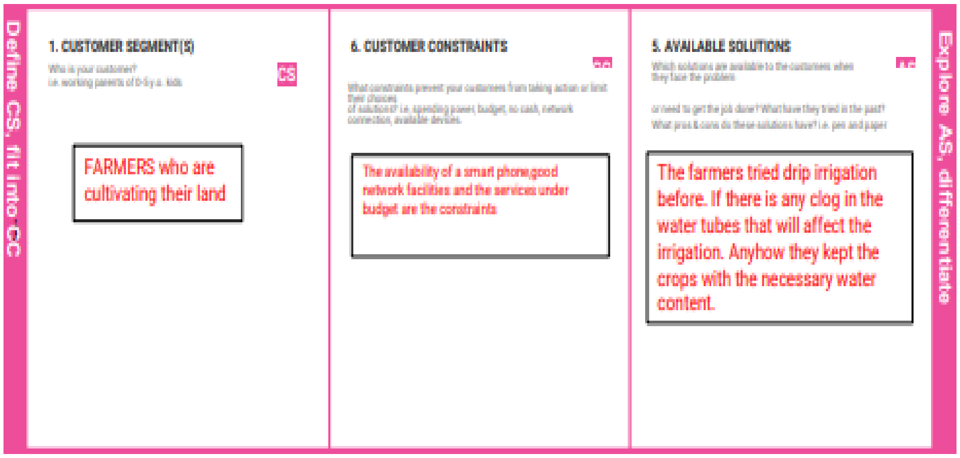
## 3.2Ideation and Brainstorming

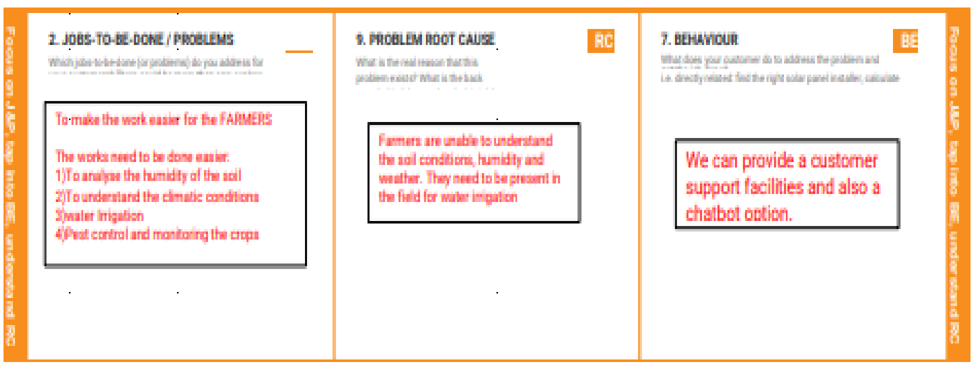


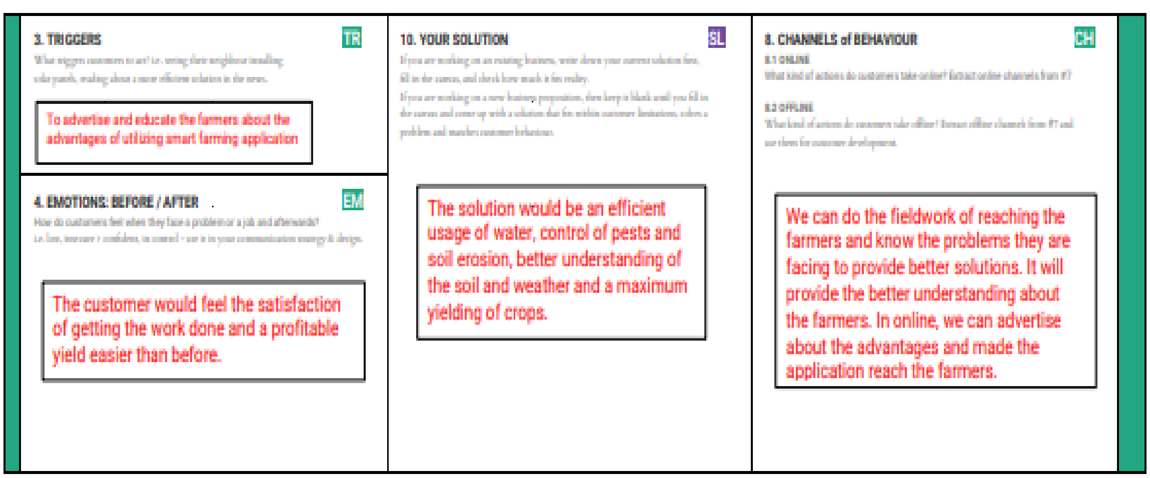
## 3.3 Proposed Solution

| S.No. | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | The better understanding of the soil, weather, humidity and water irrigation should be made easier. The automation would help farmers can do their farming from anywhere. The water level and pest control monitoring under application should be done. |
| 2. | Idea / Solution description | The application is a embedded network of different devices which make a self configuring network. The new developments of Smart Farming with use of IoT, by day turning the face of conventional agriculture methods by not only making it optimal but also making it cost efficient for farmers and reducing crop wastage. |
| 3. | Novelty / Uniqueness | Smart Farming application improves the entire Agriculture system by monitoring the field in real time using IOT devices. With the help of sensors and interconnectivity, the Internet of Things in Agriculture has not only saved the time of the farmers but has also reduced the vast use of resources such as Water and Electricity. |
| 4. | Social Impact / Customer Satisfaction | Smart farming, the dependency on manual labour has reduced significantly. The processes like pest control, fertilizing, and irrigation are increasingly becoming automated, and farmers can control them remotely. The use of smart IoT sensors can maintain these processes, increasing crop production. |
| 5. | Business Model (Revenue Model) | It is trying to execute this technique as we need to introduce an Arduino sensors which was modified with an Arduino that takes received signals from sensors. Easy operability and maintenance. Required low time for maintain. Cost is reasonable. |
| 6. | Scalability of the Solution | The adaptability of a system to increase the capacity of smart farming. The sensors and actuators used here enables the efficiency of the system. The system is scalable for the usage and the monitoring of the crops. |

### 3.4 Problem solution fit



# 4.REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENT

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Enter e-mail address and password<br>E-mail: Enter email address<br>Password: Enter password |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Log In | It will serve authentication for Logging into the system |
| FR-4 | Manage modules | It will manage system admins, roles of user and user permission |
| FR-5 | Check whether condition | It will perform Temperature monitoring status and humidity monitoring status |
| FR-6 | Log Out | Exit from the system |

## 4.2 NON FUNCTIONAL REQUIREMENT

**Non-functional Requirements:**

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

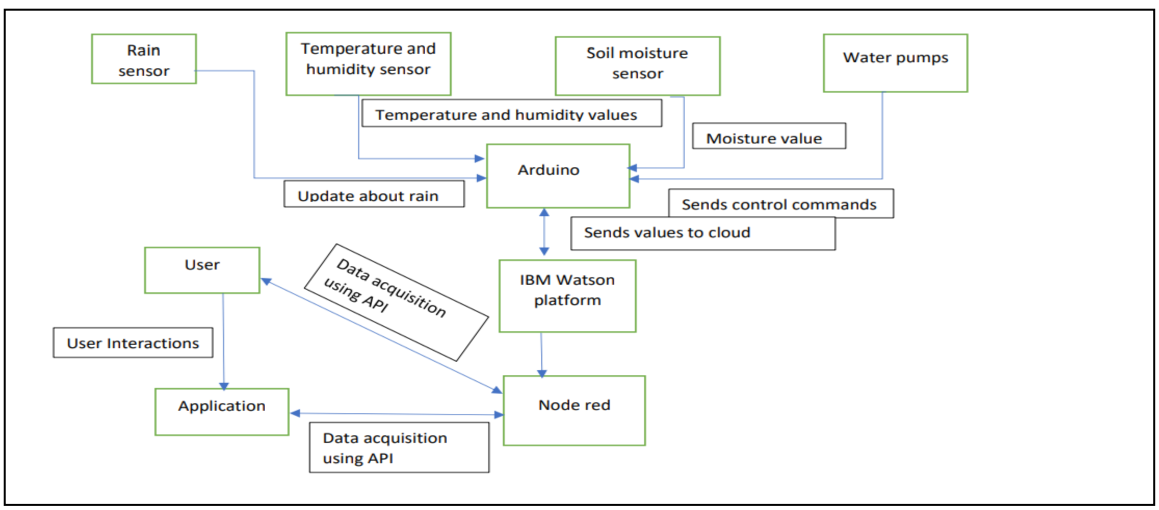
| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | Better understanding of the application, learning ability, usage efficiency, technology advancement and time saving. |
| NFR-2 | **Security** | Their preferences and decisions are kept secret. Their data would help them to achieve greater yield by analysing the records kept secretly. |
| NFR-3 | **Reliability** | The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection methodology to avoid farm service outages. |
| NFR-4 | **Performance** | The process of implementing integrated sensors with sensing soil and environmental parameters in farming will be more efficient. Since performance is a major concern for customers, it would attract them. |
| NFR-5 | **Availability** | Automatic adjustment of farming equipment made possible by linking information like crops or weather and equipment to auto-adjust temperature, humidity, etc. |
| NFR-6 | **Scalability** | Scalability is a major concern for IoT platforms. It has shown that different architectural choices of IoT platforms affect system scalability, real time decision making is feasible in an environment. |

## 5.PRODUCT DESIGN

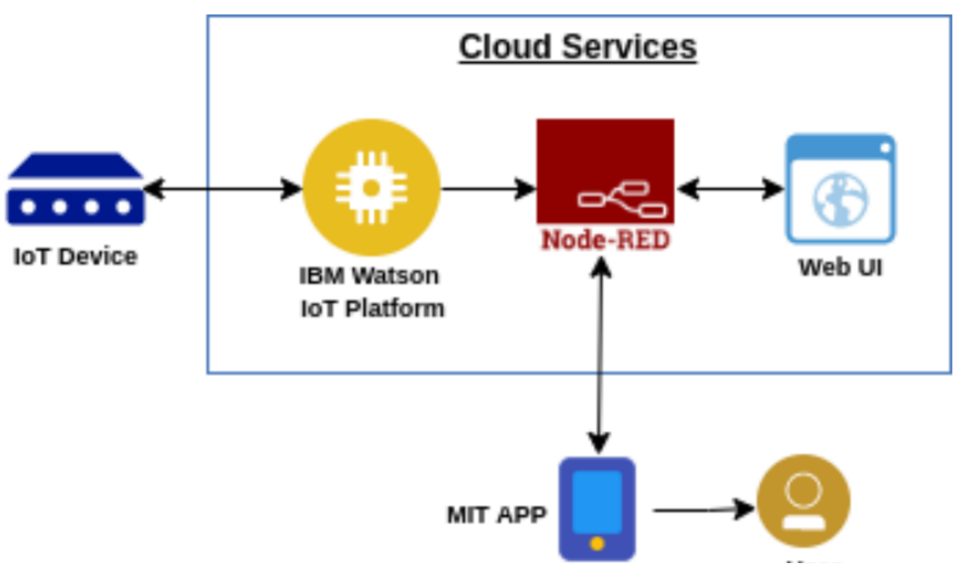
### 5.1Data flow diagrams

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 sensed using different sensors and obtained value is stored in the IBM cloud.
Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.
NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.
All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could plan through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch




### 5.2Solution and Technical Architecture



The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in IBM cloud.

Arduino Uno is used as a processing unit that process the data obtained from the sensors and weather data from the weather API
NODE-RED is used as a programming tool to write the hardware, software and APIs. The MQTT protocol is followed for the communication.
All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could decide through an app, whether to water the crop or not depending upon the sensor values, By using the app,they can remotely operate the motor switch(0n/off).

## 6.PROJECT PLANNING AND SCHEDULING

| Sprint | User Story Number | User Story/ Task | Story Points | Priority | Team Mambers |
|---|---|---|---|---|---|
| Sprint 1 | USN-1 | Sensors and wi-fi module with python code | 2 | High | Anukeerthana S<br>Anandhan V<br>Dhanusriya A S<br>Mani Bharathi M |
| Sprint 2 | USN-2 | IBM Watson IoT Platform, workflows for IoT Scenarios using Node-Red | 2 | High | Anukeerthana S<br>Anandhan V<br>Dhanusriya A S<br>Mani bharathi M |
| Sprint 3 | USN-3 | To Develop an Mobile application using MIT | 2 | High | Anukeerthana S<br>Anandhan V<br>Dhanusriya A S<br>Mani bharathi M |
| Sprint 4 | USN-4 | To make the user to interact with Software | 2 | High | Anukeerthana S<br>Anandhan V<br>Dhanusriya A S<br>Mani bharathi M |

## 7.CODING AND SOLUTIONING

### 7.1 Feature

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


#Provide your IBM Watson Device Credentials
organization = "1xIO8d"
deviceType = "abcd"
deviceId = 12
```

token = "12345678"
authMethod = "token"

**# Initialize GPIO**

def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
else:
print ("led is off")

**#print(cmd**

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-authMethod, "auth-token": Token}
deviceCli = ibmiotf.device.Client(deviceOptions)
#..............................................

except Exception as e:
print("Caught exception connecting device: %s" % str(e))
sys.exit()

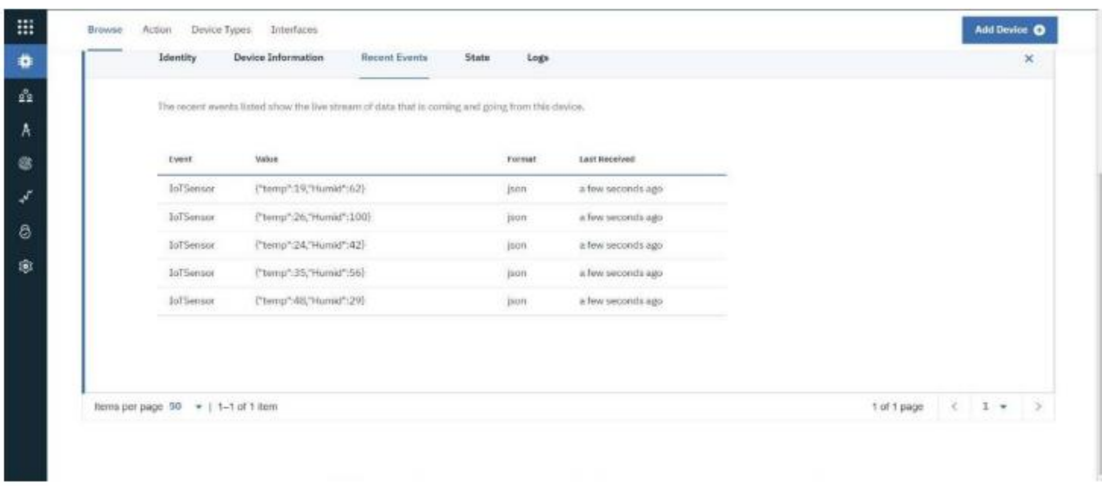
**# Connect and send a datapoint "hello" with value "world" into the cloud as an type "greeting" 10 times**
deviceCli.connect()

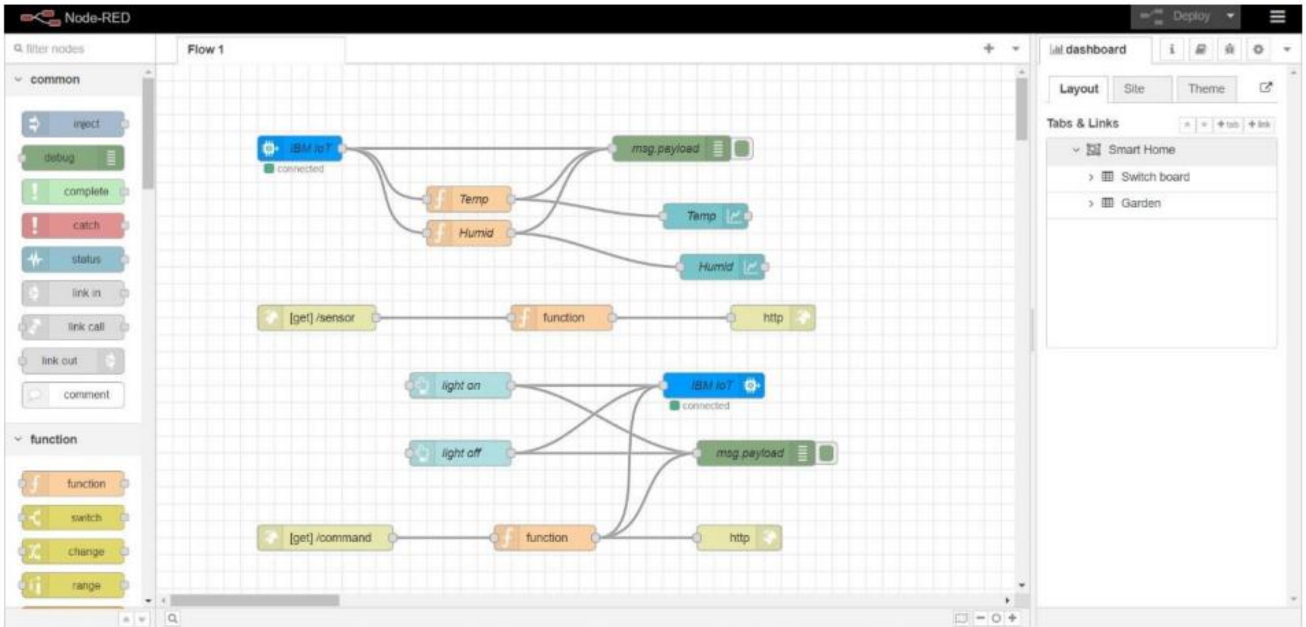
while True:
#Get Sensor Data from DHT11
temp=random.randint(90,110)
hum=random.randint(60,100
soil=random.randint(20,100
data = { 'temp' : temp, 'humid': humid , 'Mois': Mois}
**#print data def myOnPublishCallback( ):**
print (f"Published temp = {temp} C , humid = {humid} , Mois = {Mois} deg c to IBM
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**

# 8.Testing

## 8.1Testcase

Web application using node red.

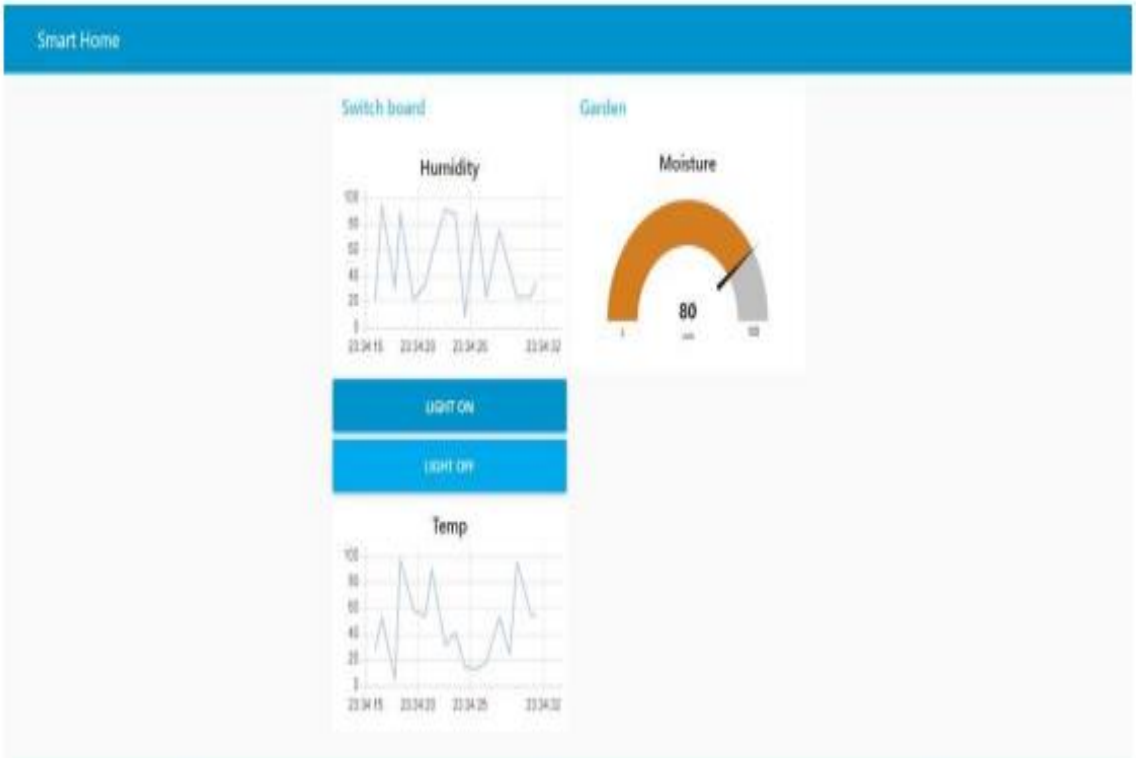





## 8.2 User Aceceptance Testing

# 9.Results

## 9.1 Performance Metrics




# 10.Advantages and disadvantages

## Advantages:

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

For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.

Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.

Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.

**Disadvantages:**

The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

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

## 11.Conclusion

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of 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 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 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 a entire.
We can update the this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. It reduces the power line cost. It will be a one time investment. We can add solar fencing technology to this project.
We can use GSM technology to this project so that the farmers can get the information directly to his home through SMS. This helps the farmer to get information if there is a internet issues.
We can add camera feature so that the farmer can monitor his field in real time. This helps in avoiding thefts

13.Appendix

**Source code**

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

**#Provide your IBM Watson Device**

```
organization = "1x108d"
deviceType = "abcd"
deviceId = “12”
authMethod = “token”
authToken = “12345678”
```

**# Initialize GPIO**

```
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="lighton":
print ("led is on")
else:
print ("motor is off")

#print(cmd)

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-
authMethod, "auth-token": Token}
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 type "greeting" 10**

```
deviceCli.connect()

while True:
#Get Sensor Data from DHT11
temp=random.randint(90,110)
```

hum=random.randint(60,100
soil=random.randint(20,100
data = { 'temp' : temp, 'hum': hum , 'soil': soil}
#print data def myOnPublishCallback( ):
print (f"Published temp = {temp} C , humid = {humid} , Mois = {Mois} deg c to IBM
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**
**deviceCli.disconnect**

Github Link : https://github.com/IBM-EPBL/IBM-Project-38542-1660382299