# PROJECT REPORT

**Project Name**: SmartFarmer –
IoT Enabled Smart Farming Application

**Team ID**: PNT2022TMID08036

**Submitted by**

**SUJITHA S**

**SRIMATHI G**

**SRINIDHI R**

**TAMIL SELVI D**

# CONTENTS

# 1. INTRODUCTION

## *1.1 Project Overview*

IoT based Smart Farming System which will enable farmers to have live data of soil moisture environment temperature at very low cost so that live monitoring can be done. Considering the standard farming procedures, farmers need to computing the agriculture plots frequently throughout the crop life to have a better idea about the crop conditions. For this, the need of smart agriculture arises, as 70% of farming time is spent monitoring and understanding the crop states instead of doing actual field work. Wireless of the sensors are facilitating the monitoring of crops constantly with higher accuracy and are able to, most importantly, detect early stages of unwanted state. Timely reporting using the value of sensors that makes the entire operation not only smart but also cost effective due to its precise monitoring capabilities. Sensors can be of the installed and start collecting data in a short time, which is then available online for further analyses nearly immediately. 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*

We have tried to focus on different scientific applications which could be put together in 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 assess the basic condition of the field. This is the project from the motivation of the farmers working in the farmlands are solely dependent on the rains and bore wells for irrigation of their land. In recent times, the farmers have been using irrigation technique through the 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*

- The existing model consist of IoT based system embedded together with Arduino and various sensors.
- The values of the sensors will be displayed in the Arduino LCD display.
- In this system the Arduino UNO is embedded with Arduino IDE of C/C++.

## *2.2 References*

1. Rajalakshmi.P, S.Devi Mahalakshmi, “IOT Based Crop-Field Monitoring and Irrigation Automation System”. International Conference on Intelligent Systems and Control (ISCO) 2016.

2. Baltej Kaur, Danish Inamdar, Vishal Raut, Akash Patil, Nayan Patil, “A Survey on Smart Drip Irrigation System”. International Research Journal of Engineering and Technology (IRJET) Volume: 03 Issue: 02, Feb 2016.

3. G.Parameswaran, K.Sivaprasath, “Arduino Based Smart Drip Irrigation System Using Internet of Things”. DOI:10.4010/2016.1348, ISSN 2321 3361©2016 IJESC.

4. Bouzekri Amel, Chabane Mohamed, Benahmed Tarek, “Smart Irrigation System using Internet of Things”. The Fourth International Conference on Future Generation Communication Technologies (FGCT 2015)

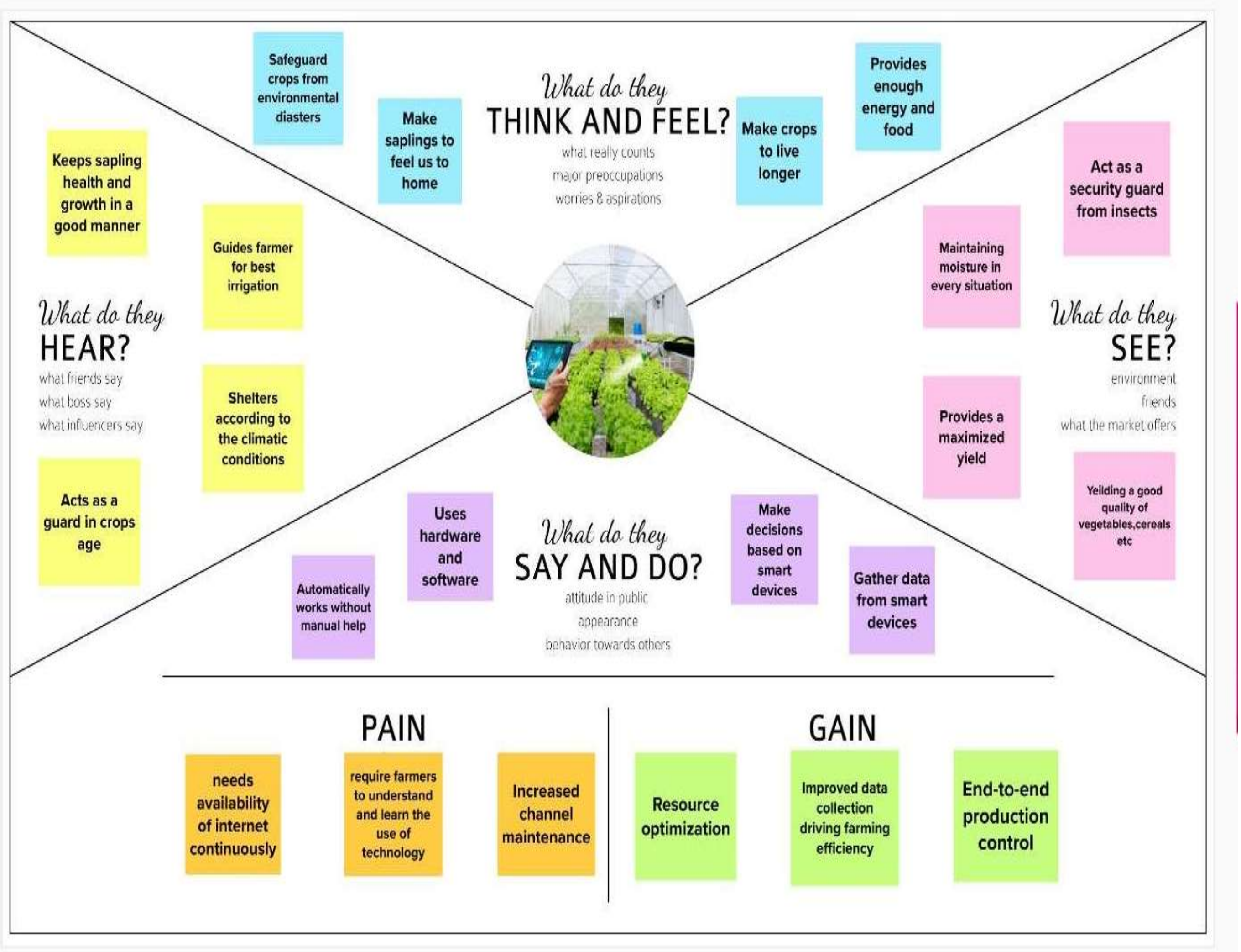
5. R.Hemalatha, G.Deepika, D.Dhanalakshmi, Dharanipriya, M.Divya, “Internet of Things (IOT) Based Smart Irrigation”. International Journal of Advanced Research in Biology Engineering Science and Technology (IJARBEST) Vol.2, Issue 2, Feb 2016.

## *2.3 Problem Statement Definition*

IoT plays a key role in smart agriculture. Internets of Things (IoT) sensors are used to provide necessary information about agriculture fields. The main advantage of IoT is to monitor the agriculture by using the sensor networks and collect the data from different sensors and send by wireless protocol. By using IoT system the smart agriculture is powered by NodeMCU. It includes the humidity sensor, temperature sensor, moisture sensor and motor. This system starts to check the humidity and moisture level. The sensors are used to sense the level of water and if the level is below the range then the system automatically starts watering. According to the change in temperature level the sensor does its job. IoT also shows the information of humidity, moisture level by including date and time.

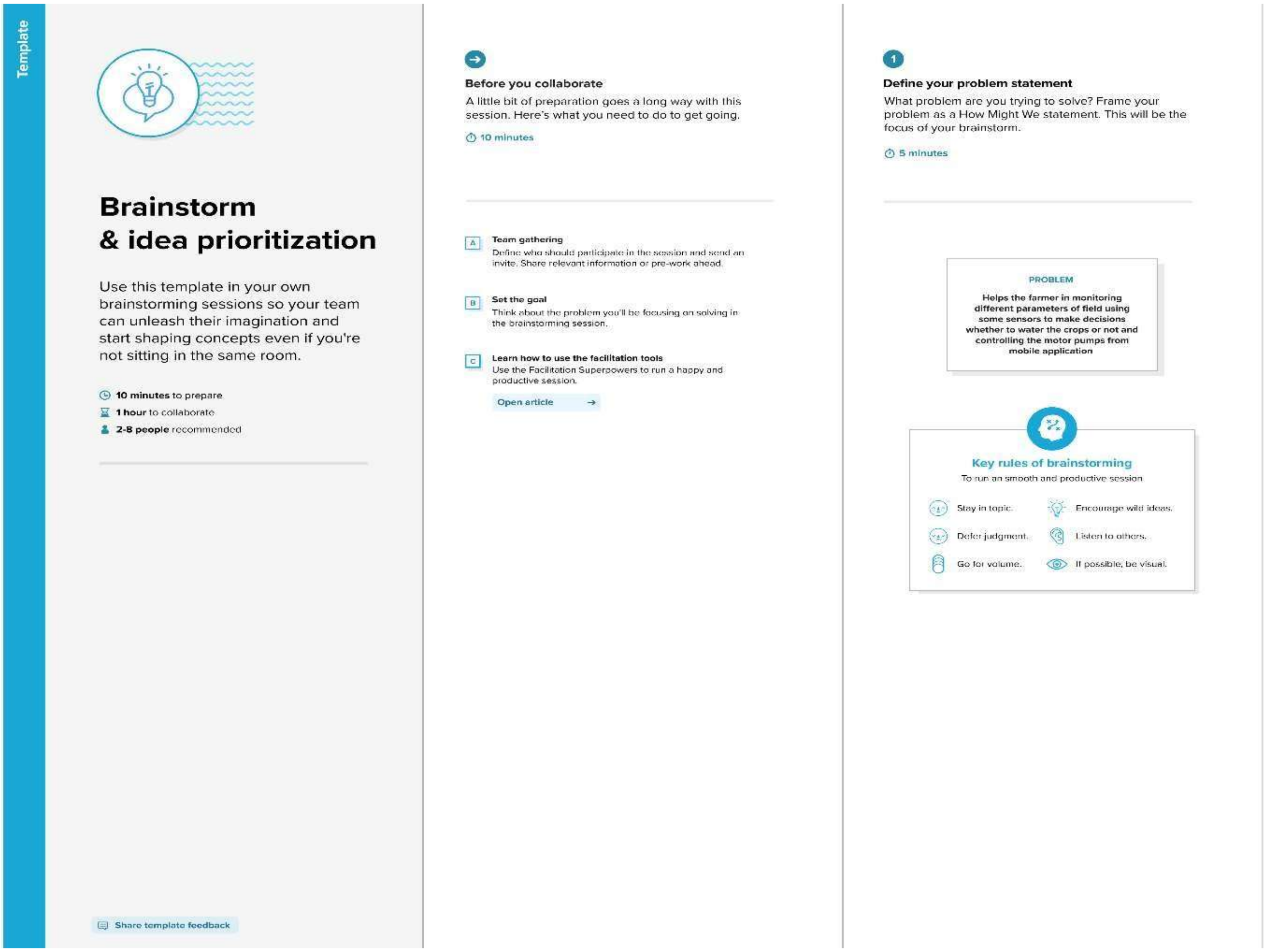
# 3. IDEATION & PROPOSED SOLUTION

## *3.1 Empathy Map Canvas*



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Safeguard crops from environmental diasters
Make saplings to feel us to home
Make crops to live longer
Provides enough energy and food
What do they HEAR?
what friends say
what boss say
what influencers say
Keeps sapling health and growth in a good manner
Guides farmer for best irrigation
Shelters according to the climatic conditions
Acts as a guard in crops age
What do they SEE?
environment
friends
what the market offers
Act as a security guard from insects
Maintaining moisture in every situation
Provides a maximized yield
Yeilding a good quality of vegetables,cereals etc
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Automatically works without manual help
Uses hardware and software
Make decisions based on smart devices
Gather data from smart devices
PAIN
needs availability of internet continuously
require farmers to understand and learn the use of technology
Increased channel maintenance
GAIN
Resource optimization
Improved data collection driving farming efficiency
End-to-end production control

## *3.2 Ideation & Brainstorming*



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

Share template feedback

**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

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **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

PROBLEM

**Helps the farmer in monitoring different parameters of field using some sensors to make decisions whether to water the crops or not and controlling the motor pumps from mobile application**

**Key rules of brainstorming**

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

2

**Brainstorm**

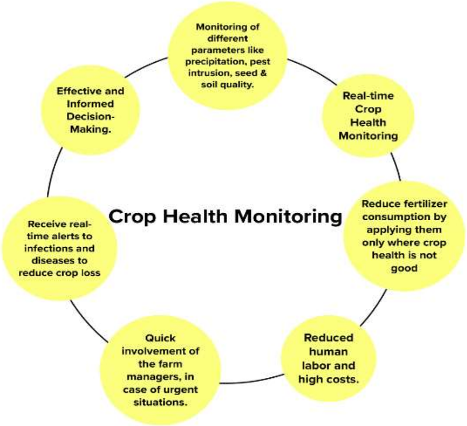
Write down any ideas that come to mind that address your problem statement.

⏱ 10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

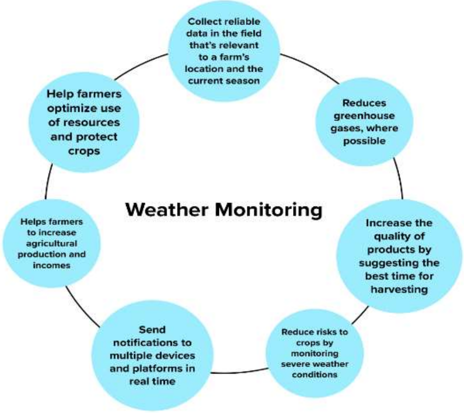
## Sujitha S



Crop Health Monitoring

- Monitoring of different parameters like precipitation, pest intrusion, seed & soil quality.
- Real-time Crop Health Monitoring
- Reduce fertilizer consumption by applying them only where crop health is not good
- Reduced human labor and high costs.
- Quick involvement of the farm managers, in case of urgent situations.
- Receive real-time alerts to infections and diseases to reduce crop loss
- Effective and Informed Decision-Making.

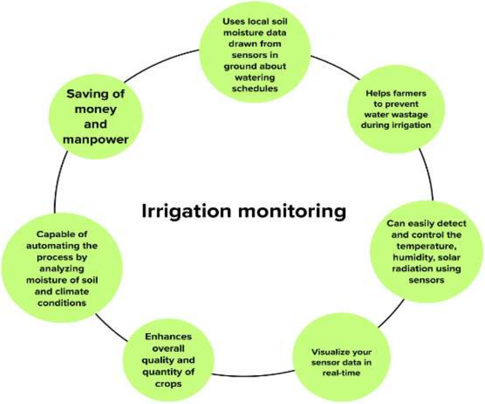
## Srimathi G



Weather Monitoring

- Collect reliable data in the field that's relevant to a farm's location and the current season
- Reduces greenhouse gases, where possible
- Increase the quality of products by suggesting the best time for harvesting
- Reduce risks to crops by monitoring severe weather conditions
- Send notifications to multiple devices and platforms in real time
- Helps farmers to increase agricultural production and incomes
- Help farmers optimize use of resources and protect crops

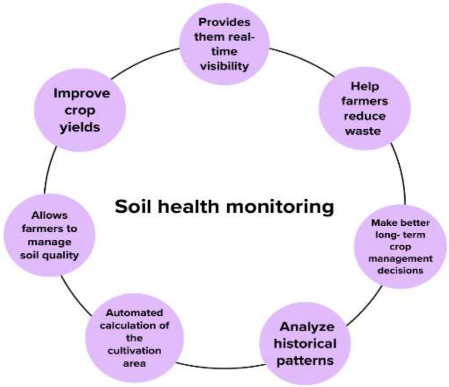
## Srinidhi R



Irrigation monitoring

- Uses local soil moisture data drawn from sensors in ground about watering schedules
- Helps farmers to prevent water wastage during irrigation
- Can easily detect and control the temperature, humidity, solar radiation using sensors
- Visualize your sensor data in real-time
- Enhances overall quality and quantity of crops
- Capable of automating the process by analyzing moisture of soil and climate conditions
- Saving of money and manpower

## Tamil Selvi D



Soil health monitoring

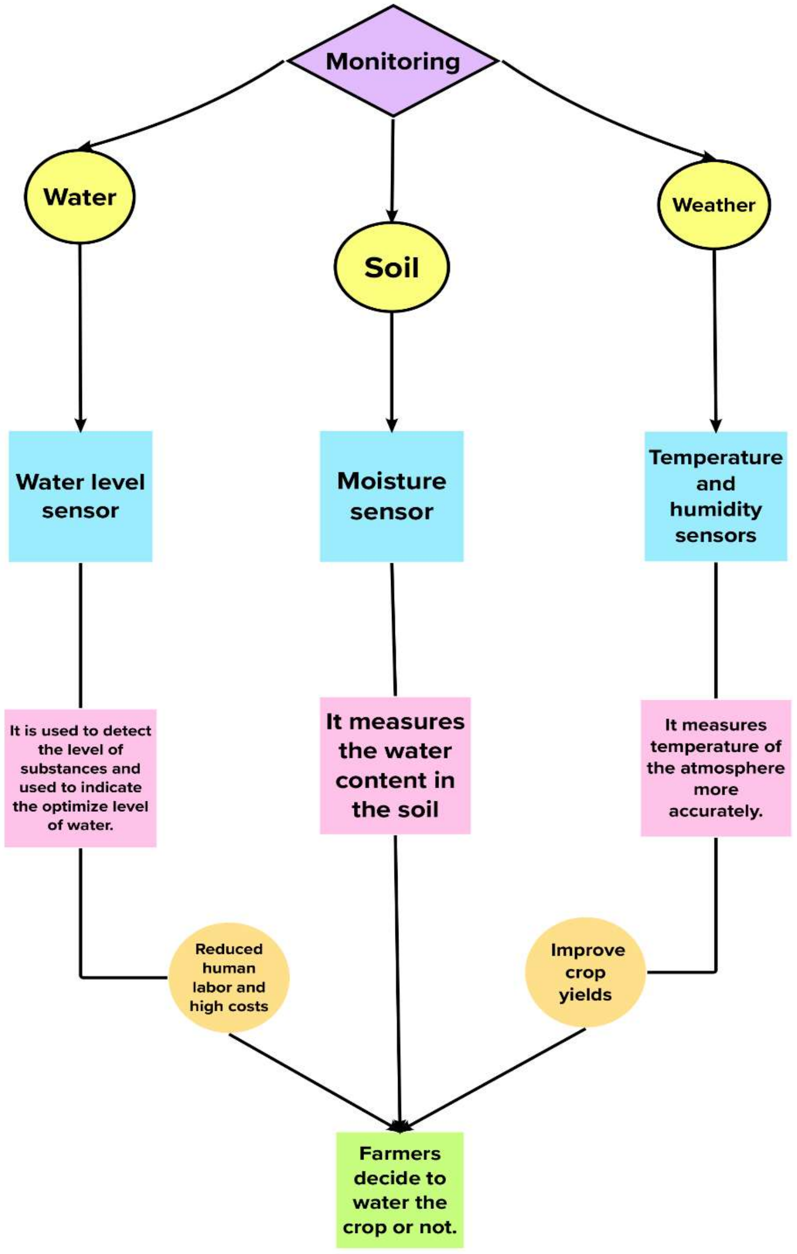
- Provides them real-time visibility
- Help farmers reduce waste
- Make better long- term crop management decisions
- Analyze historical patterns
- Automated calculation of the cultivation area
- Allows farmers to manage soil quality
- Improve crop yields

3

### Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes



Monitoring

Water

Soil

Weather

Water level sensor

Moisture sensor

Temperature and humidity sensors

It is used to detect the level of substances and used to indicate the optimize level of water.

It measures the water content in the soil

It measures temperature of the atmosphere more accurately.

Reduced human labor and high costs

Improve crop yields

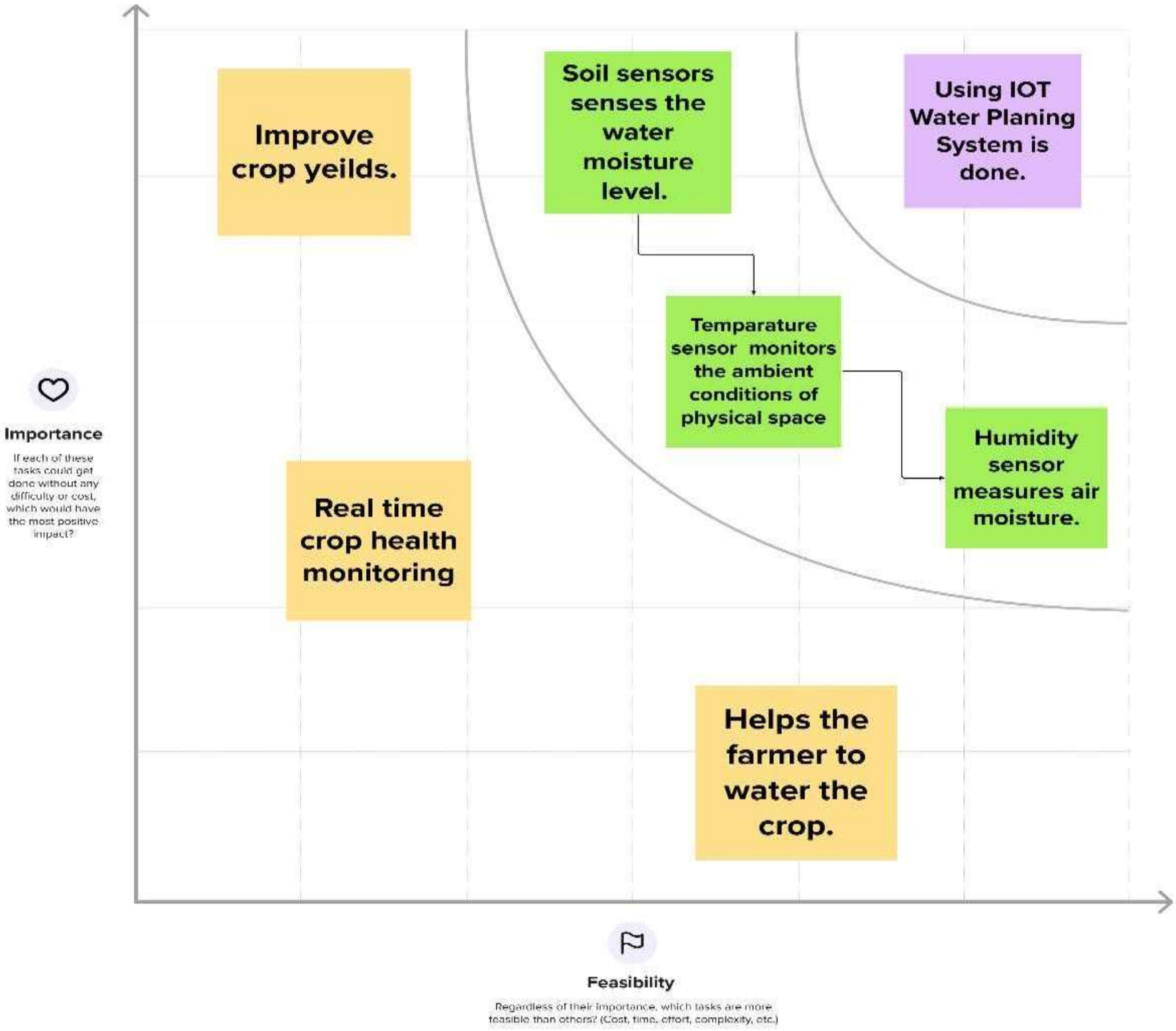
Farmers decide to water the crop or not.

4

**Prioritize**

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



Improve crop yeilds.

Soil sensors senses the water moisture level.

Using IOT Water Planing System is done.

Temparature sensor monitors the ambient conditions of physical space

Humidity sensor measures air moisture.

Real time crop health monitoring

Helps the farmer to water the crop.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

### *3.3 Proposed Solution*

- **Problem Statement:**

  Helps the farmer in maintaining different parameters of field using some sensors to make decisions whether to water the crops or not and controlling the motor pumps from mobile application.

- **Idea / Solution description:**

  Sensors are used to collect information about soil, temperature, humidity and are sent to IoT based cloud platform to make farmer relevant decision for better crop yield.

- **Novelty / Uniqueness:**

  Enable farmers to monitor crops from anywhere.

- **Social impact / Customer Satisfaction**

  Smart farming increases the quality of crops and maintains irrigation by decreasing the water wastage.

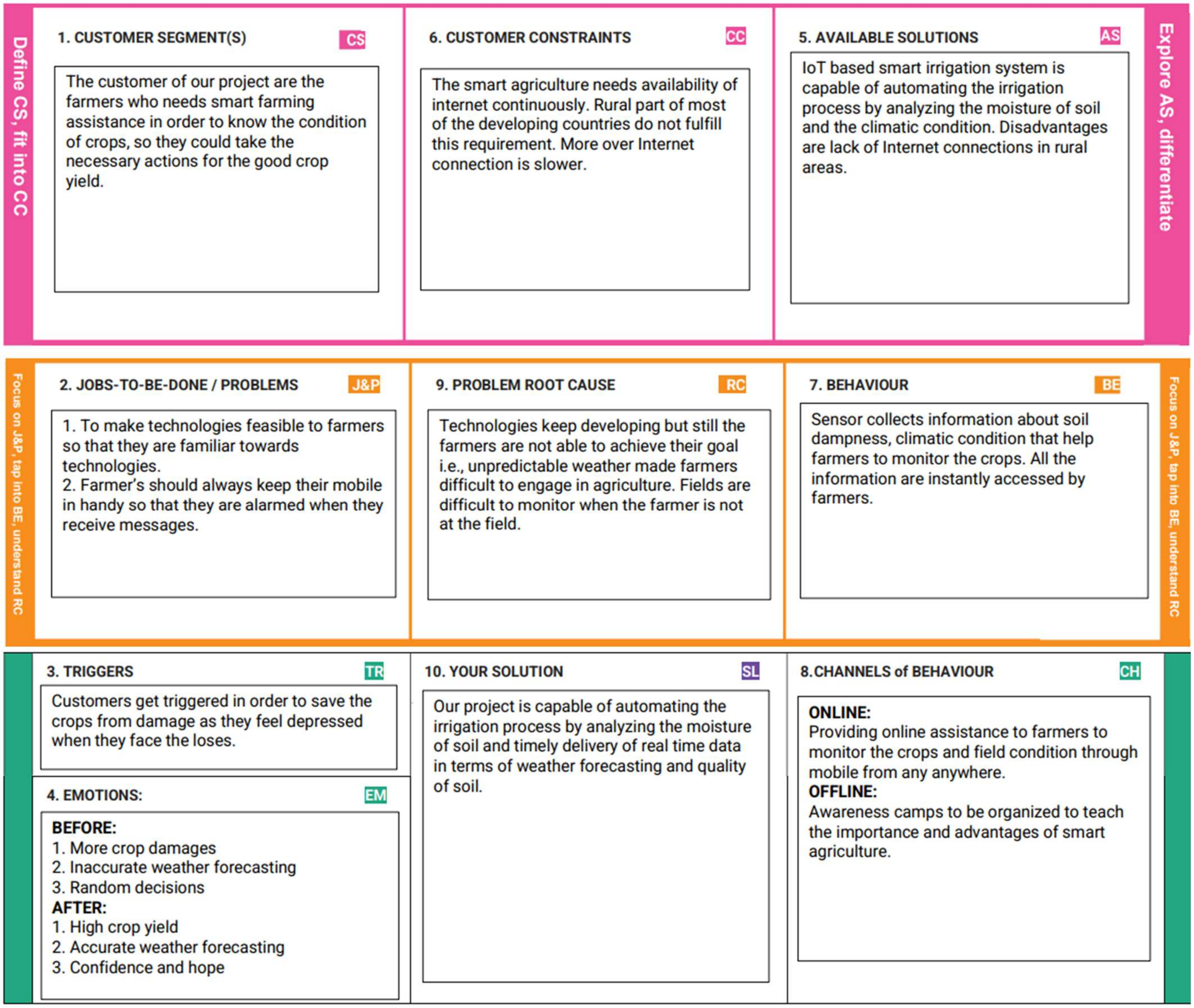
- **Business Model (Revenue Model)**

  This mobile application uses real time monitoring of crops and remotely to accelerate decision making process.

- **Scalability of the Solution**

  Maximizing operational efficiency and minimizing labor costs to simplify their workload through automated processing.

## *3.4 Problem Solution Fit*



Define CS, fit into CC

**1. CUSTOMER SEGMENT(S)** CS

The customer of our project are the farmers who needs smart farming assistance in order to know the condition of crops, so they could take the necessary actions for the good crop yield.

**6. CUSTOMER CONSTRAINTS** CC

The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfill this requirement. More over Internet connection is slower.

**5. AVAILABLE SOLUTIONS** AS

IoT based smart irrigation system is capable of automating the irrigation process by analyzing the moisture of soil and the climatic condition. Disadvantages are lack of Internet connections in rural areas.

Explore AS, differentiate

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

**2. JOBS-TO-BE-DONE / PROBLEMS** J&P

1. To make technologies feasible to farmers so that they are familiar towards technologies.
2. Farmer's should always keep their mobile in handy so that they are alarmed when they receive messages.

**9. PROBLEM ROOT CAUSE** RC

Technologies keep developing but still the farmers are not able to achieve their goal i.e., unpredictable weather made farmers difficult to engage in agriculture. Fields are difficult to monitor when the farmer is not at the field.

**7. BEHAVIOUR** BE

Sensor collects information about soil dampness, climatic condition that help farmers to monitor the crops. All the information are instantly accessed by farmers.

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

**3. TRIGGERS** TR

Customers get triggered in order to save the crops from damage as they feel depressed when they face the loses.

**4. EMOTIONS:** EM

**BEFORE:**
1. More crop damages
2. Inaccurate weather forecasting
3. Random decisions

**AFTER:**
1. High crop yield
2. Accurate weather forecasting
3. Confidence and hope

**10. YOUR SOLUTION** SL

Our project is capable of automating the irrigation process by analyzing the moisture of soil and timely delivery of real time data in terms of weather forecasting and quality of soil.

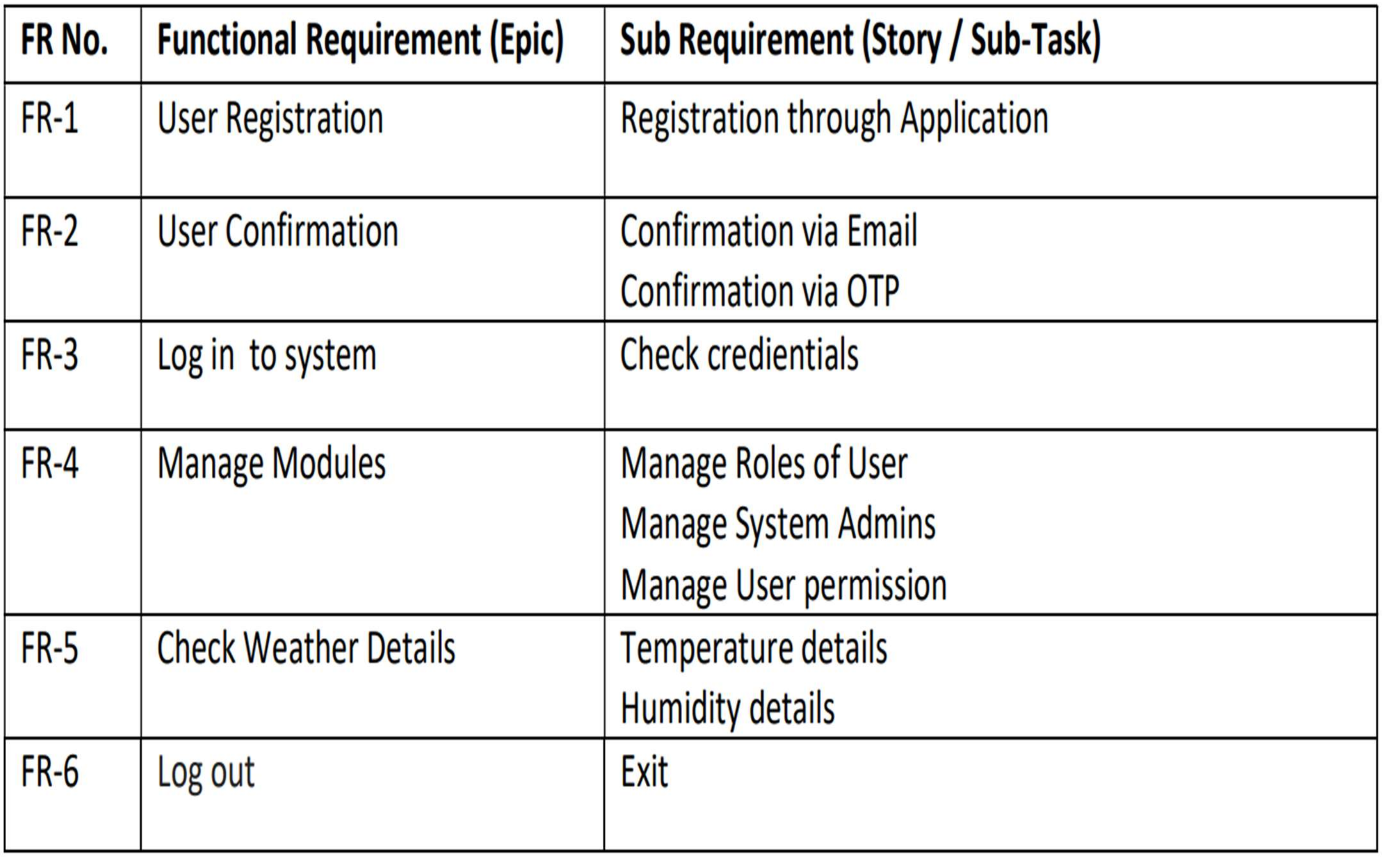
**8.CHANNELS of BEHAVIOUR** CH

**ONLINE:**
Providing online assistance to farmers to monitor the crops and field condition through mobile from any anywhere.

**OFFLINE:**
Awareness camps to be organized to teach the importance and advantages of smart agriculture.

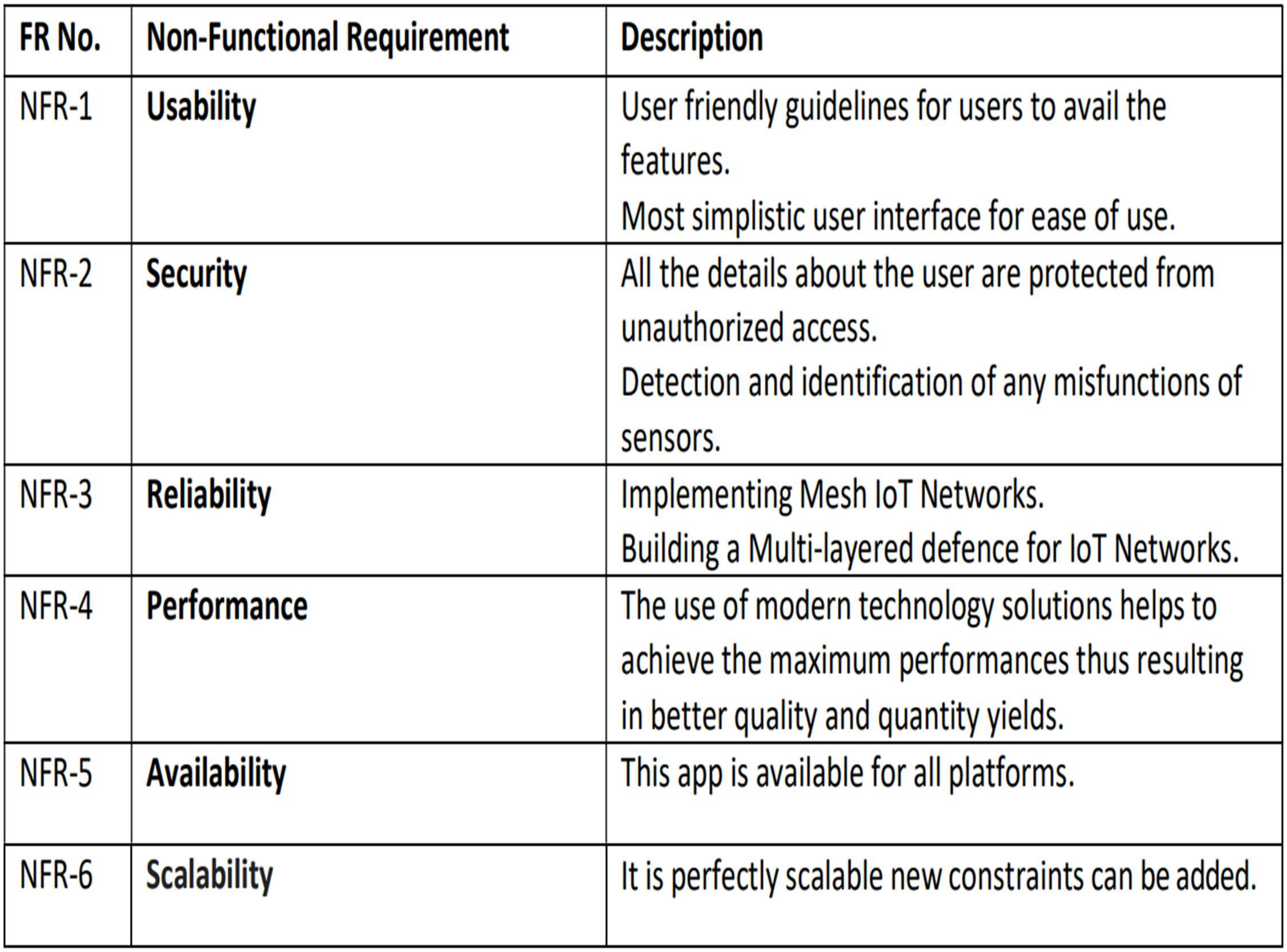
# 4. REQUIREMENT ANALYSIS

## *4.1 Functional Requirements*



| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Application |
| FR-2 | User Confirmation | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Log in  to system | Check credientials |
| FR-4 | Manage Modules | Manage Roles of User<br>Manage System Admins<br>Manage User permission |
| FR-5 | Check Weather Details | Temperature details<br>Humidity details |
| FR-6 | Log out | Exit |

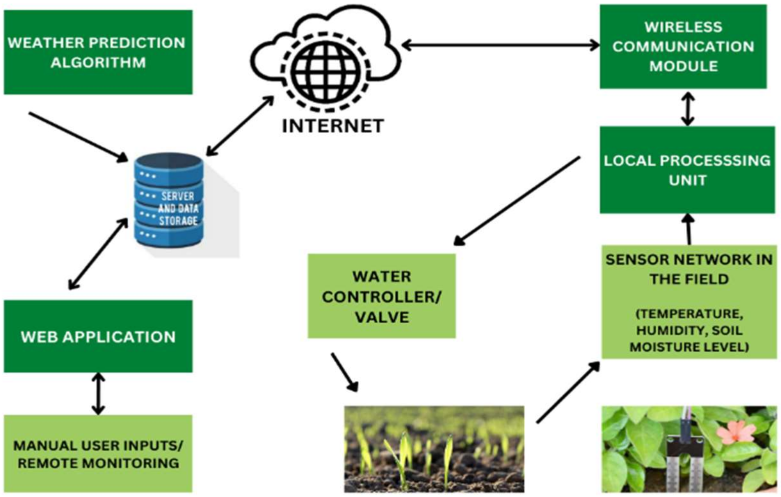
## *4.2 Non-Functional Requirements*

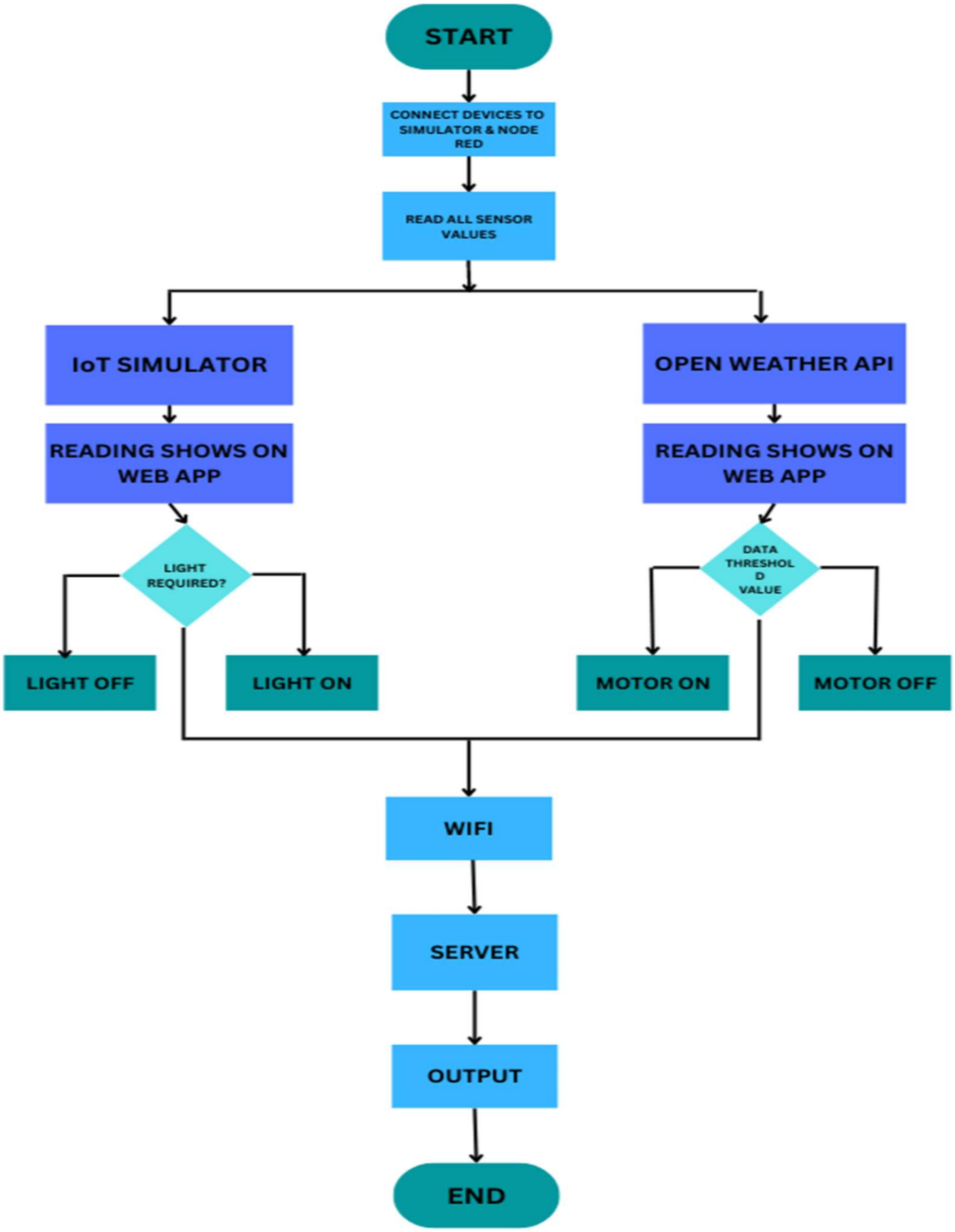


| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | **Usability** | User friendly guidelines for users to avail the features.<br>Most simplistic user interface for ease of use. |
| NFR-2 | **Security** | All the details about the user are protected from unauthorized access.<br>Detection and identification of any misfunctions of sensors. |
| NFR-3 | **Reliability** | Implementing Mesh IoT Networks.<br>Building a Multi-layered defence for IoT Networks. |
| NFR-4 | **Performance** | The use of modern technology solutions helps to achieve the maximum performances thus resulting in better quality and quantity yields. |
| NFR-5 | **Availability** | This app is available for all platforms. |
| NFR-6 | **Scalability** | It is perfectly scalable new constraints can be added. |

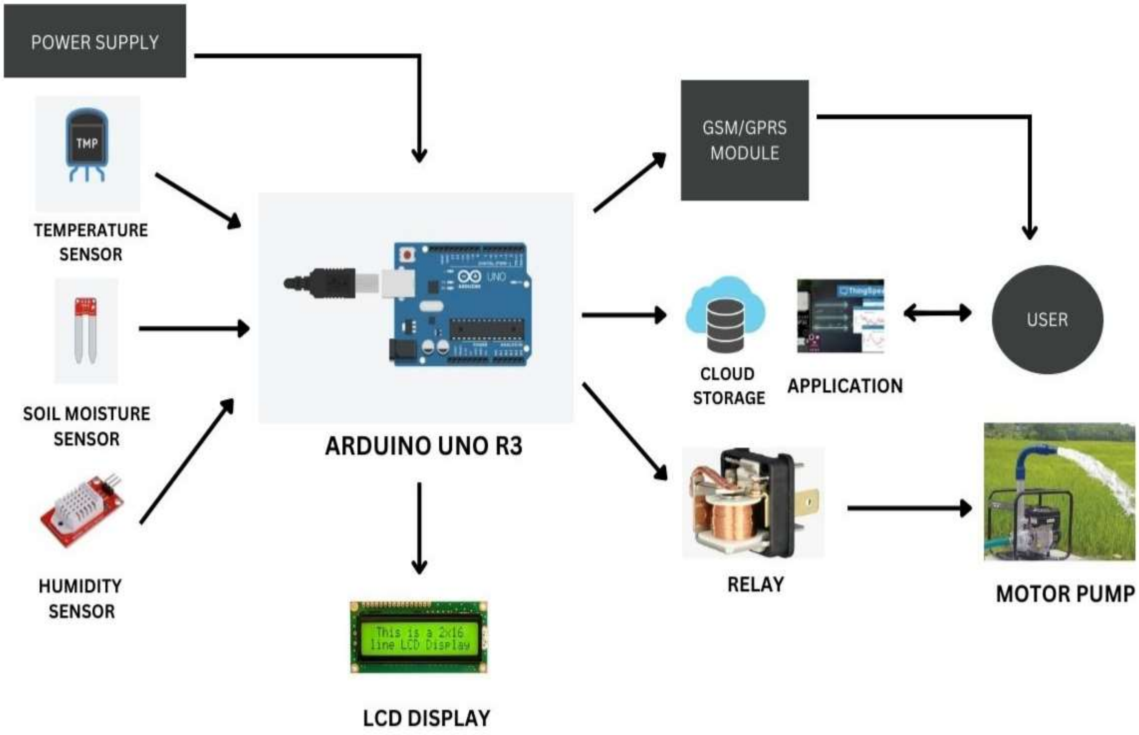
# 5. PROJECT DESIGN

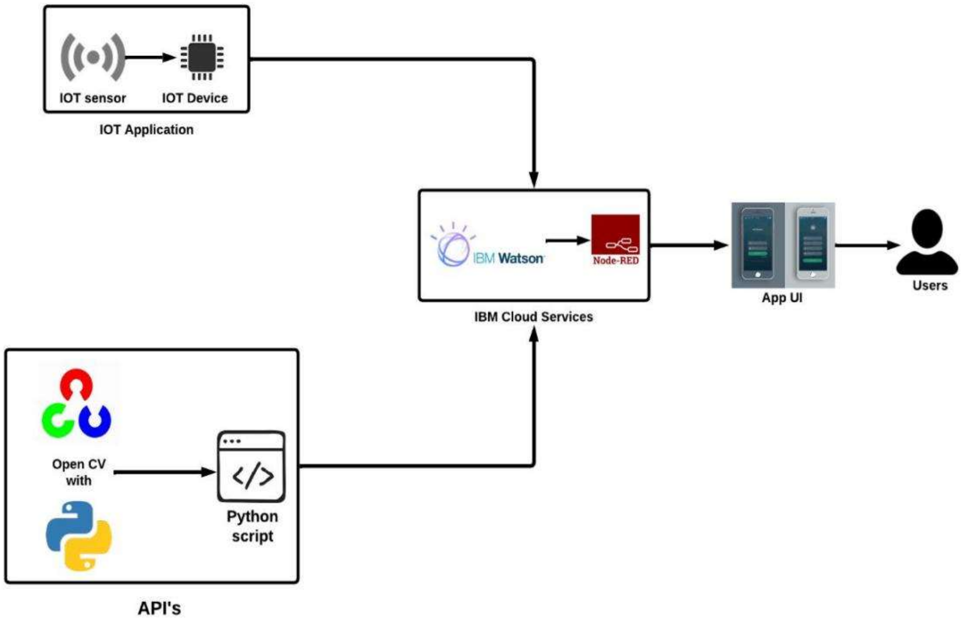
## *5.1 Data Flow Diagrams*


WEATHER PREDICTION ALGORITHM
INTERNET
WIRELESS COMMUNICATION MODULE
SERVER AND DATA STORAGE
LOCAL PROCESSSING UNIT
WATER CONTROLLER/ VALVE
SENSOR NETWORK IN THE FIELD
(TEMPERATURE, HUMIDITY, SOIL MOISTURE LEVEL)
WEB APPLICATION
MANUAL USER INPUTS/ REMOTE MONITORING

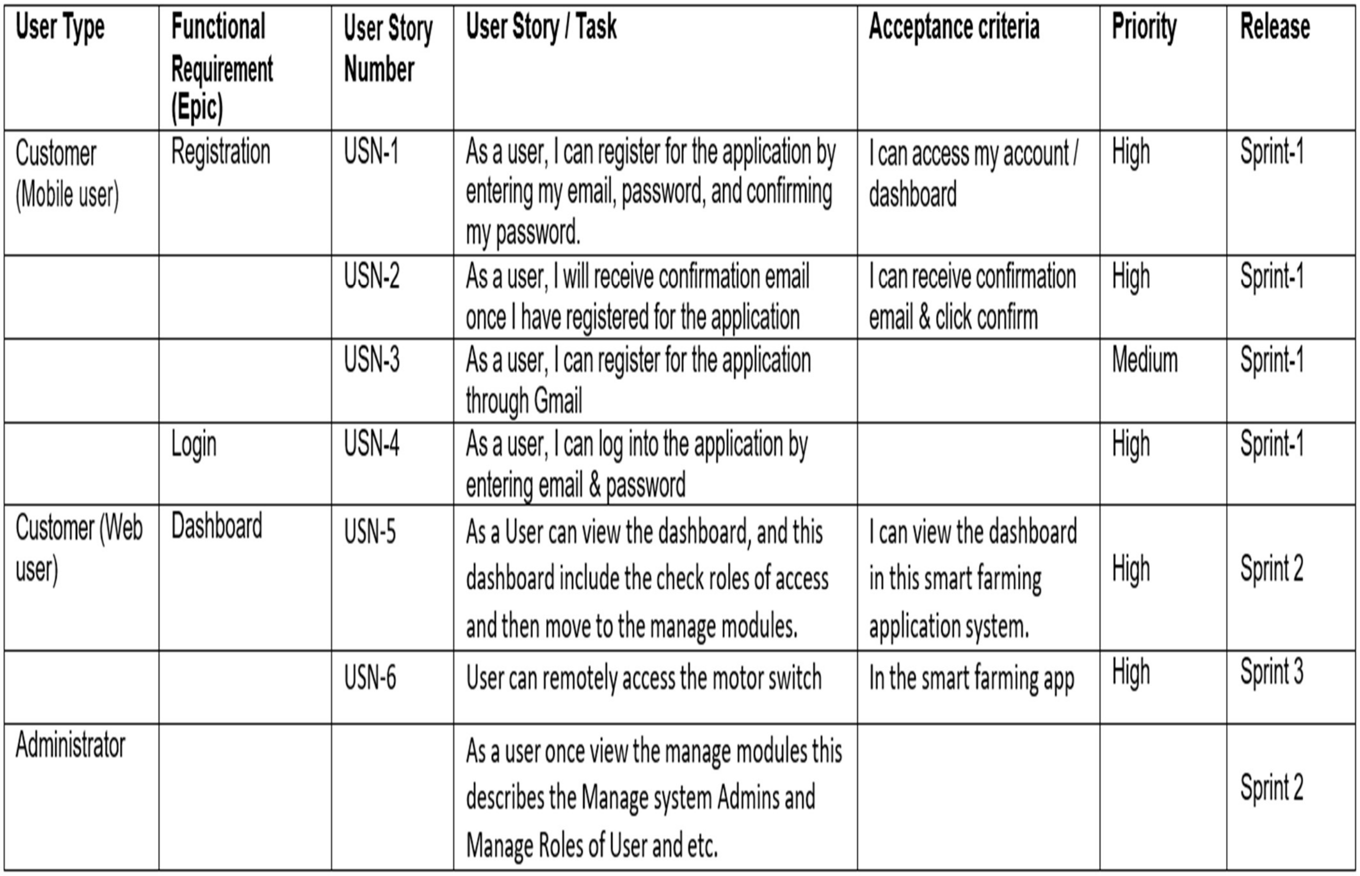

START
CONNECT DEVICES TO SIMULATOR & NODE RED
READ ALL SENSOR VALUES
IoT SIMULATOR
OPEN WEATHER API
READING SHOWS ON WEB APP
READING SHOWS ON WEB APP
LIGHT REQUIRED?
DATA THRESHOL D VALUE
LIGHT OFF
LIGHT ON
MOTOR ON
MOTOR OFF
WIFI
SERVER
OUTPUT
END

## 5.2 Solution & Technical Architecture





## *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 Gmail | | Medium | Sprint-1 |
| | 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 & SCHEDULING

## *6.1 Sprint Planning & Estimation*

| Milestone name | Milestone number | Description | Mandatory |
|---|---|---|---|
| Prerequisties | M-01 | We will to leam about cloud services and we willinstall python and required libraries | Yes |
| Project objectives | M-02 | Gain knowledge of Watson Platform, IBM Cloudant DB, web application development andGenerating QR codes with the required data | Yes |
| Create and configure IBMcloud services | M-03 | We will create and configure the IBM Cloud services which are being used in completing thisproject | Yes |
| Develop a python script Topublish And Subscribe To IBM IOT platform | M-04 | We will develop the python script to publish thedata and subscribe the data from the IBM Watson IOT Platform | Yes |

| | | | |
|---|---|---|---|
| Develop A Web Application Using Node-RED Service | M-05 | A Web UI will be created in Node-RED usingdashboard nodes available in it | Yes |
| Ideation phase | M-06 | Prepare Literature Survey on the selected Project and Information Gathering, empathy mapand ideation | Yes |
| Project design phase-I | M-07 | Prepare Proposed solution, problem-solution fitand Solution Architecture | Yes |
| Project Design Phase-II | M-08 | Prepare Customer journey functional requirements, Dataflow diagram and TechnologyArchitecture | Yes |
| Project Planning Phase | M-09 | Prepare Milestone list, Activity list and SprintDelivery Plan | Yes |
| Project Development Phase | M-10 | Project Development delivery of Sprint 1, Sprint2, Sprint 3, Sprint 4 | Yes |

## *6.2 Sprint Delivery Schedule*

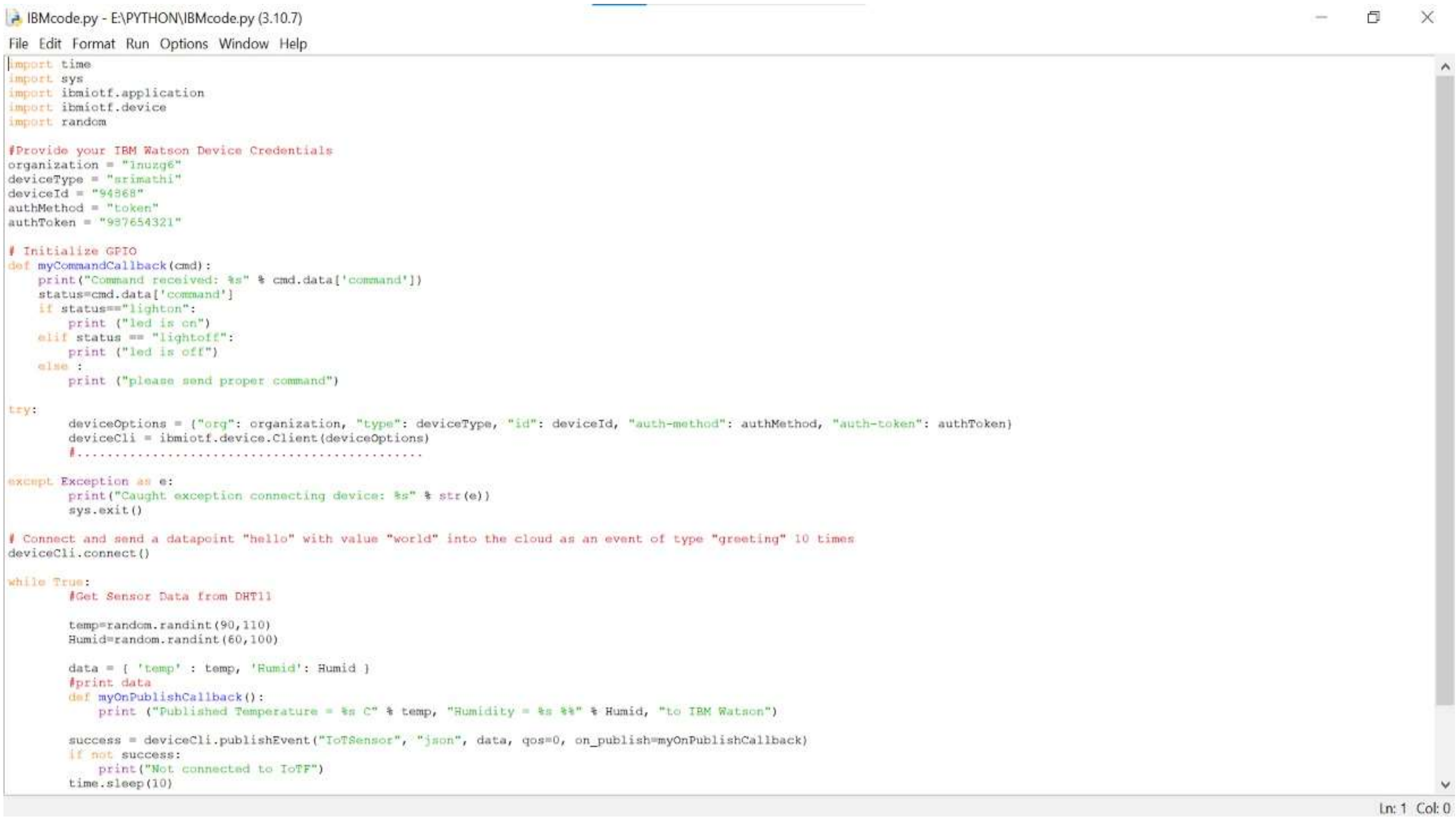
| Activity number | Activity | Sub activity | Assignedmemb er | Status |
|---|---|---|---|---|
| 1. | Prerequisites | IBM cloud services | Sujitha S | Completed |
| | | software | Srinidhi R | Completed |
| 2. | Project objectives | | | Completed |

| | | | | |
|---|---|---|---|---|
| 3. | Create and configure IBMcloud services | Create And Configure IBMcloudservices | Sujitha S | Completed |
| | | Create Node-RED service | Srinidhi R | Completed |
| 4. | Develop a python script To publish And Subscribe To IBM IOT platform | Develop the Python code | Tamil Selvi D | Completed |
| | | Publish data to the IBM cloud | Sujitha S | In-Progress |
| 5. | Develop A Web Application Using Node-RED Service | Develop The Web ApplicationUsingNode-RED | Sujitha S | In-Progress |
| | | Use Dashboard Nodes ForCreatingUI(Web App) | Sujitha S | In-Progress |

| | | | | |
|---|---|---|---|---|
| 6. | Ideation phase | 10.1. Literature Survey. | Srinidhi R | Completed |
| | | 10.2. Empathy map. | Tamil Selvi D | Completed |
| | | 10.3. Ideation. | Sujitha S | Completed |
| 7. | Project design phase-I | 10.1 Proposed Solution | Srimathi G | Completed |
| | | 10.2 Problem solution fit. | Srinidhi R | Completed |
| | | 10.3 Solution Architecture. | Tamil Selvi D | Completed |
| 08. | Project Design | 11.1 Customer journey. | Sujitha S | Completed |

## 7. Coding and Solutioning

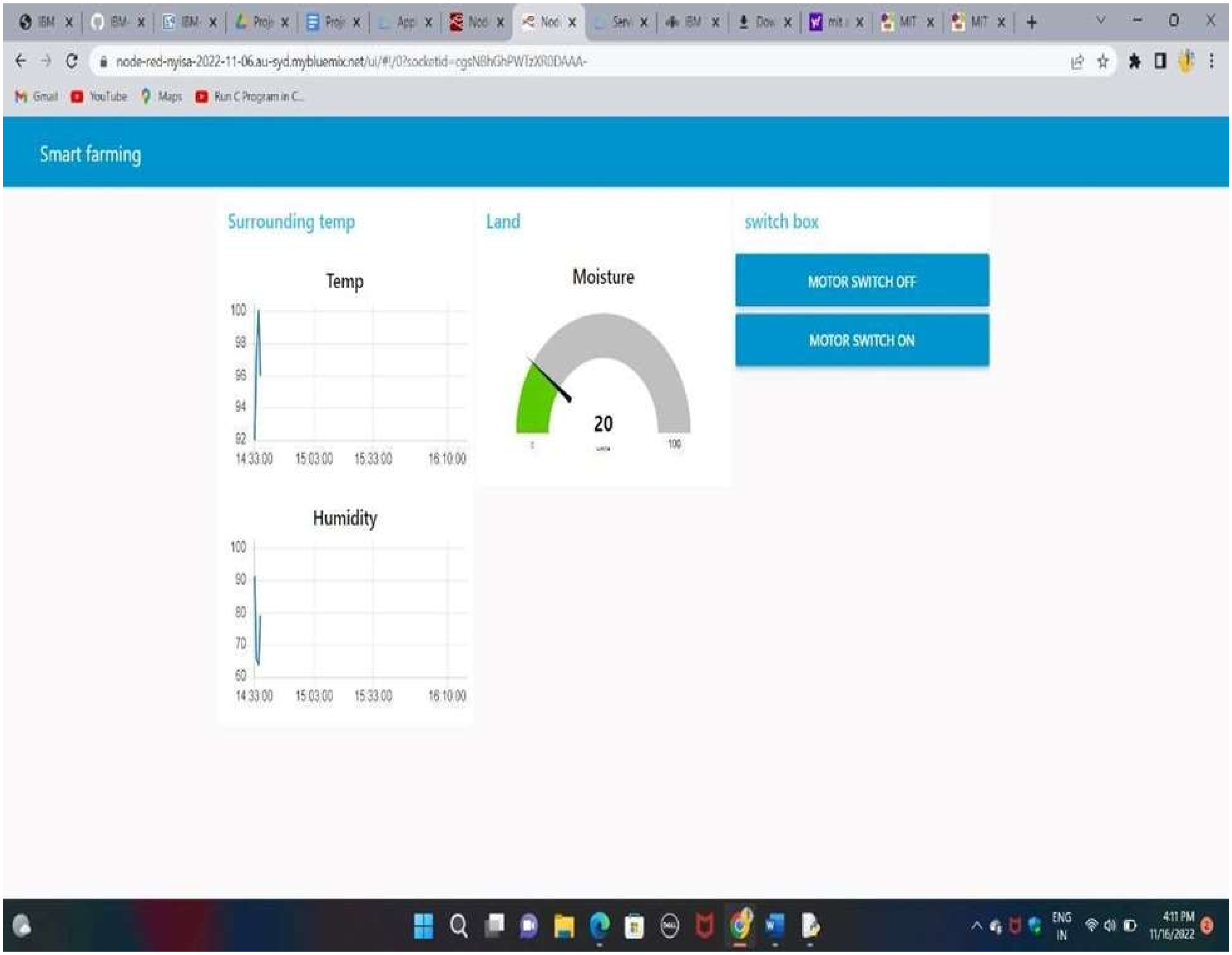
### 7.1 Feature

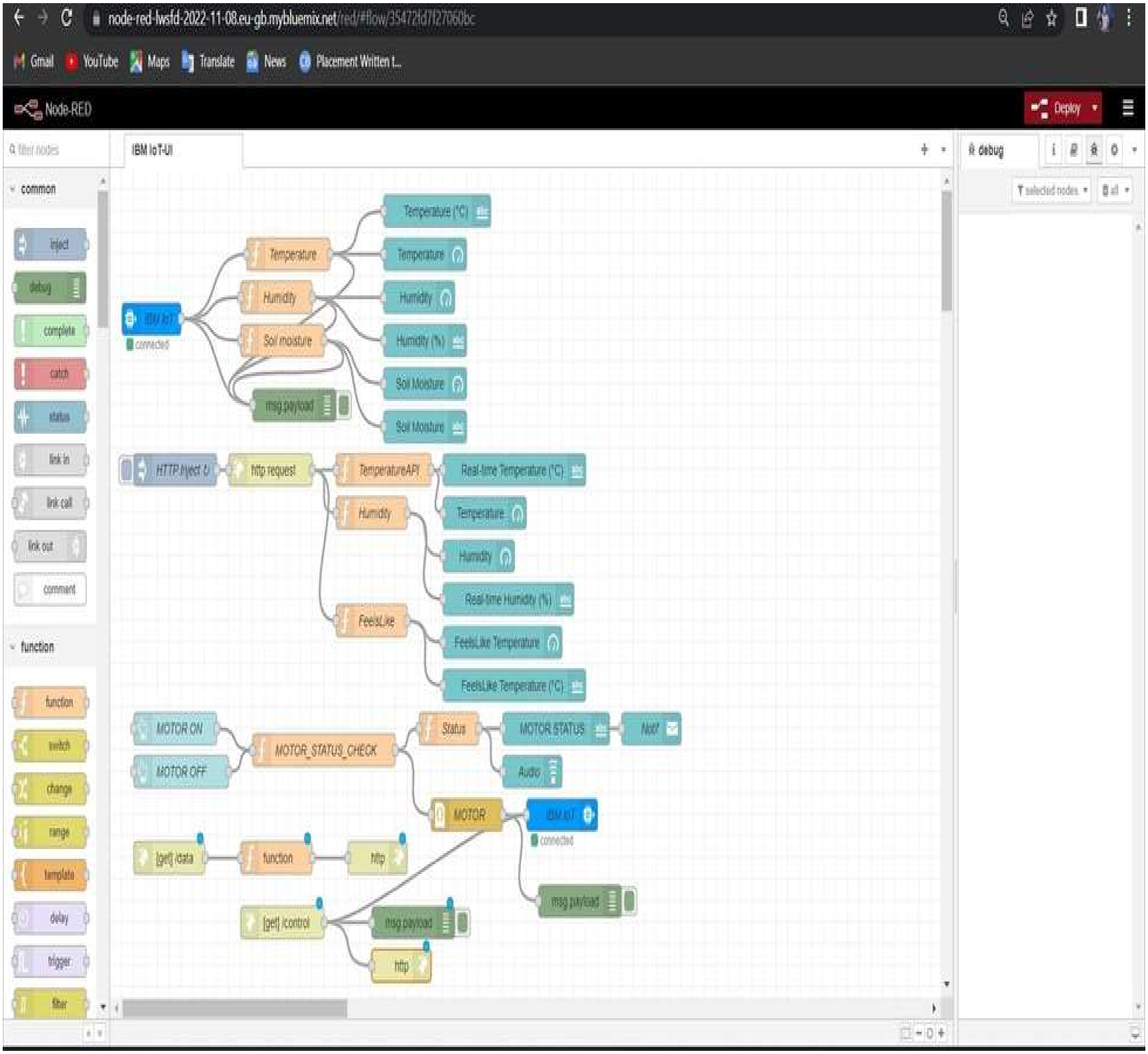

IBMcode.py - E:\PYTHON\IBMcode.py (3.10.7)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

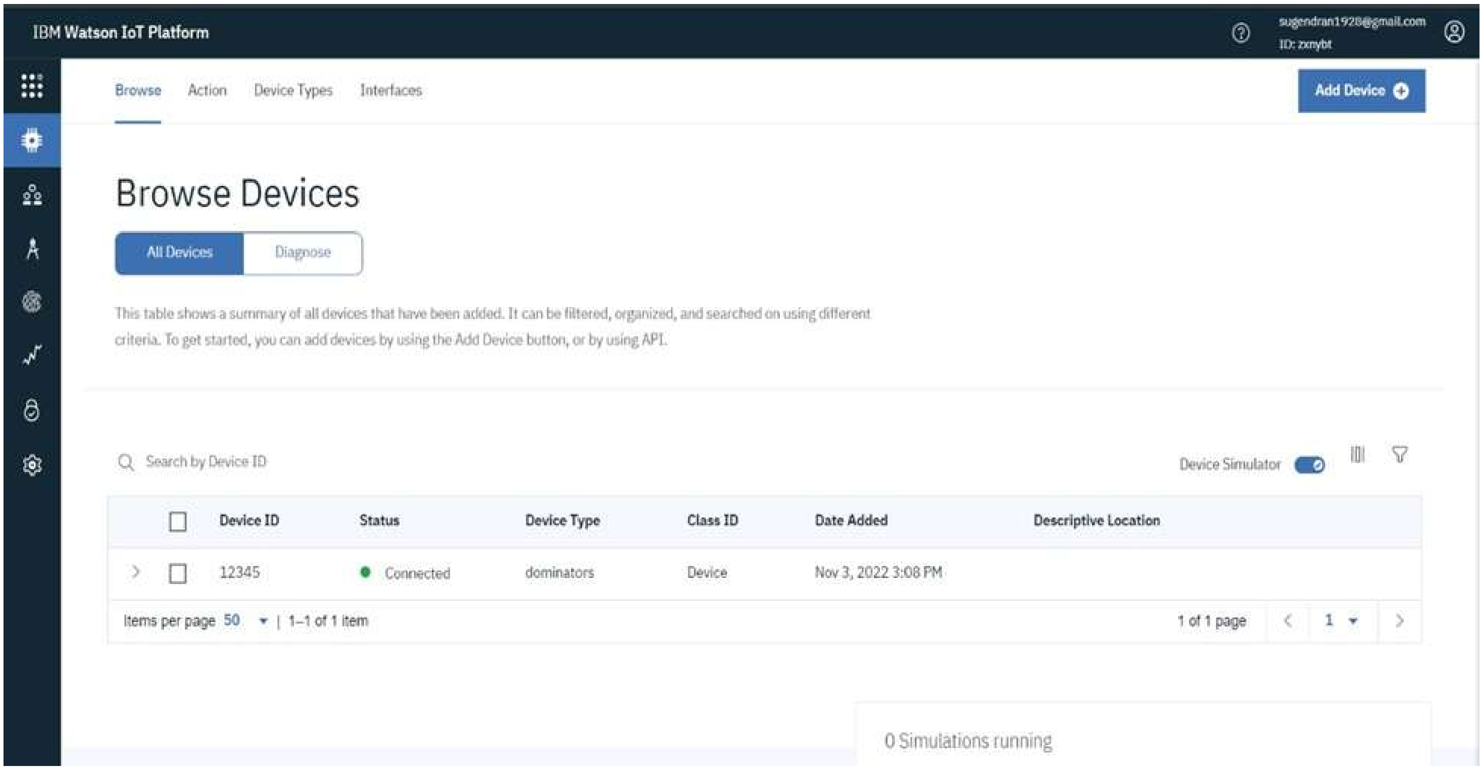
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "1nuzq6"
deviceType = "srimathi"
deviceId = "94868"
authMethod = "token"
authToken = "987654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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)

        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "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)
```

## *8. Testing*

### *8.1 Test cases*




node-red-lwsfd-2022-11-08.eu-gb.mybluemix.net/red/#flow/35472fd7f27060bc
Gmail
YouTube
Maps
Translate
News
Placement Written t...
Node-RED
Deploy
filter nodes
IBM IoT-UI
debug
selected nodes
all
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
template
delay
trigger
filter
IBM IoT
connected
Temperature
Humidity
Soil moisture
msg.payload
Temperature (°C)
Temperature
Humidity
Humidity (%)
Soil Moisture
Soil Moisture
HTTP Inject
http request
TemperatureAPI
Humidity
FeelsLike
Real-time Temperature (°C)
Temperature
Humidity
Real-time Humidity (%)
FeelsLike Temperature
FeelsLike Temperature (°C)
MOTOR ON
MOTOR OFF
MOTOR_STATUS_CHECK
Status
MOTOR STATUS
Notif
Audio
MOTOR
IBM IoT
connected
[get] /data
function
http
msg.payload
[get] /control
msg.payload
http

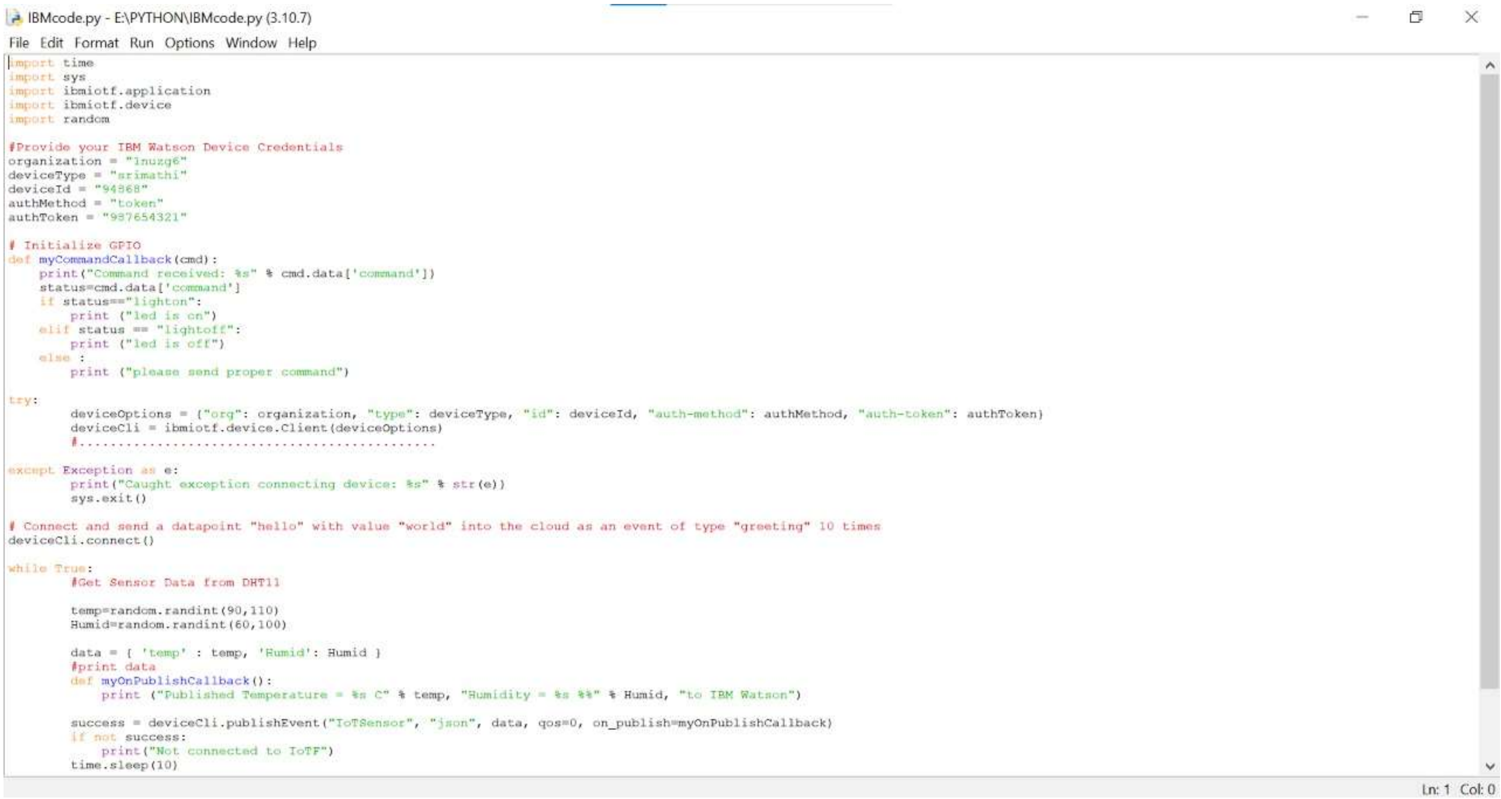


Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

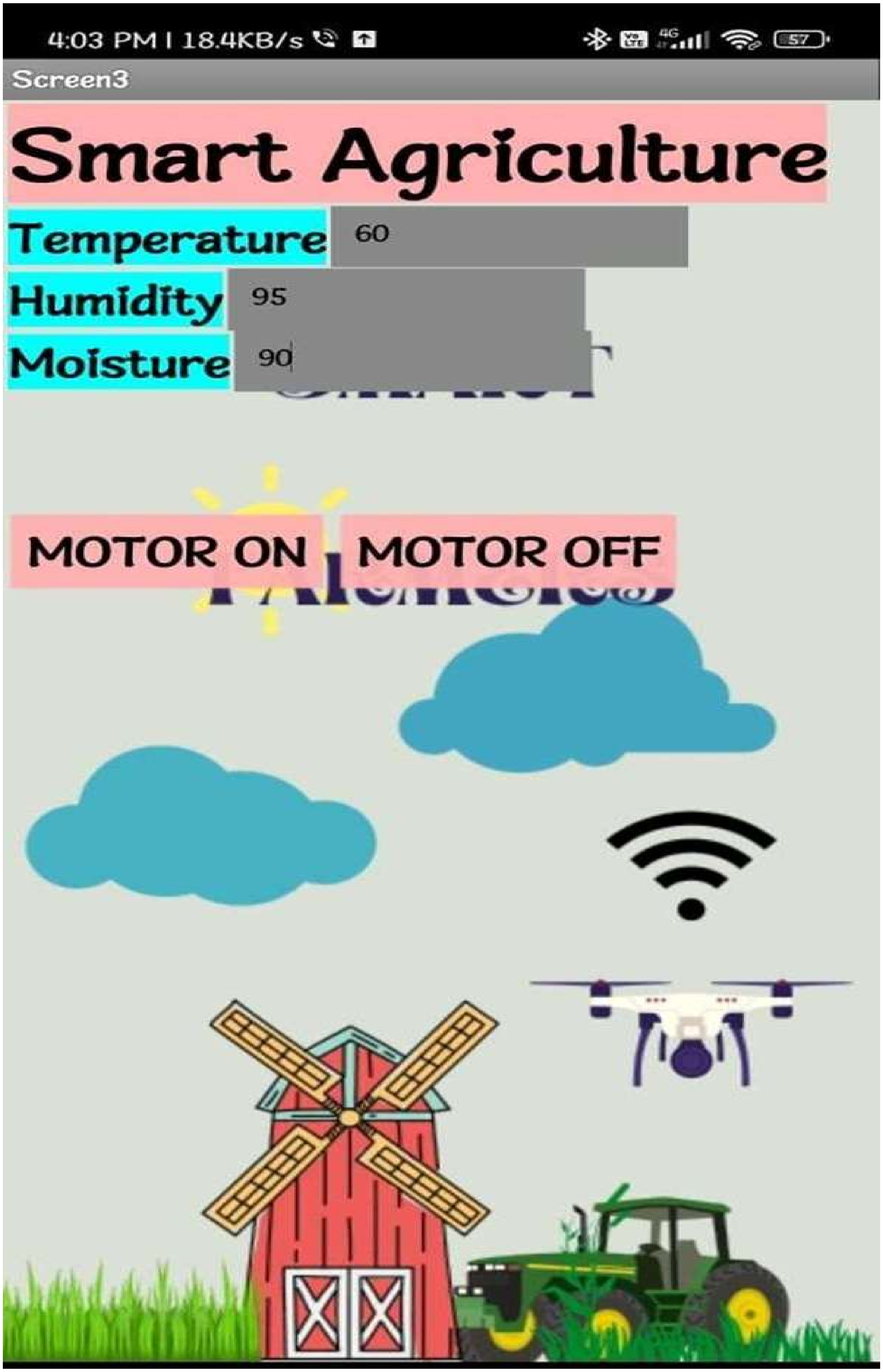
| | Device ID | Status | Device Type | Class ID | Date Added | Descriptive Location |
|---|---|---|---|---|---|---|
| □ | 12345 | ● Connected | dominators | Device | Nov 3, 2022 3:08 PM | |

0 Simulations running


IBMcode.py - E:\PYTHON\IBMcode.py (3.10.7)
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials
organization = "1nuzg6"
deviceType = "srimathi"
deviceId = "94868"
authMethod = "token"
authToken = "987654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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)

        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "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)
```

### *8.2 User Acceptance Testing*



Smart Agriculture

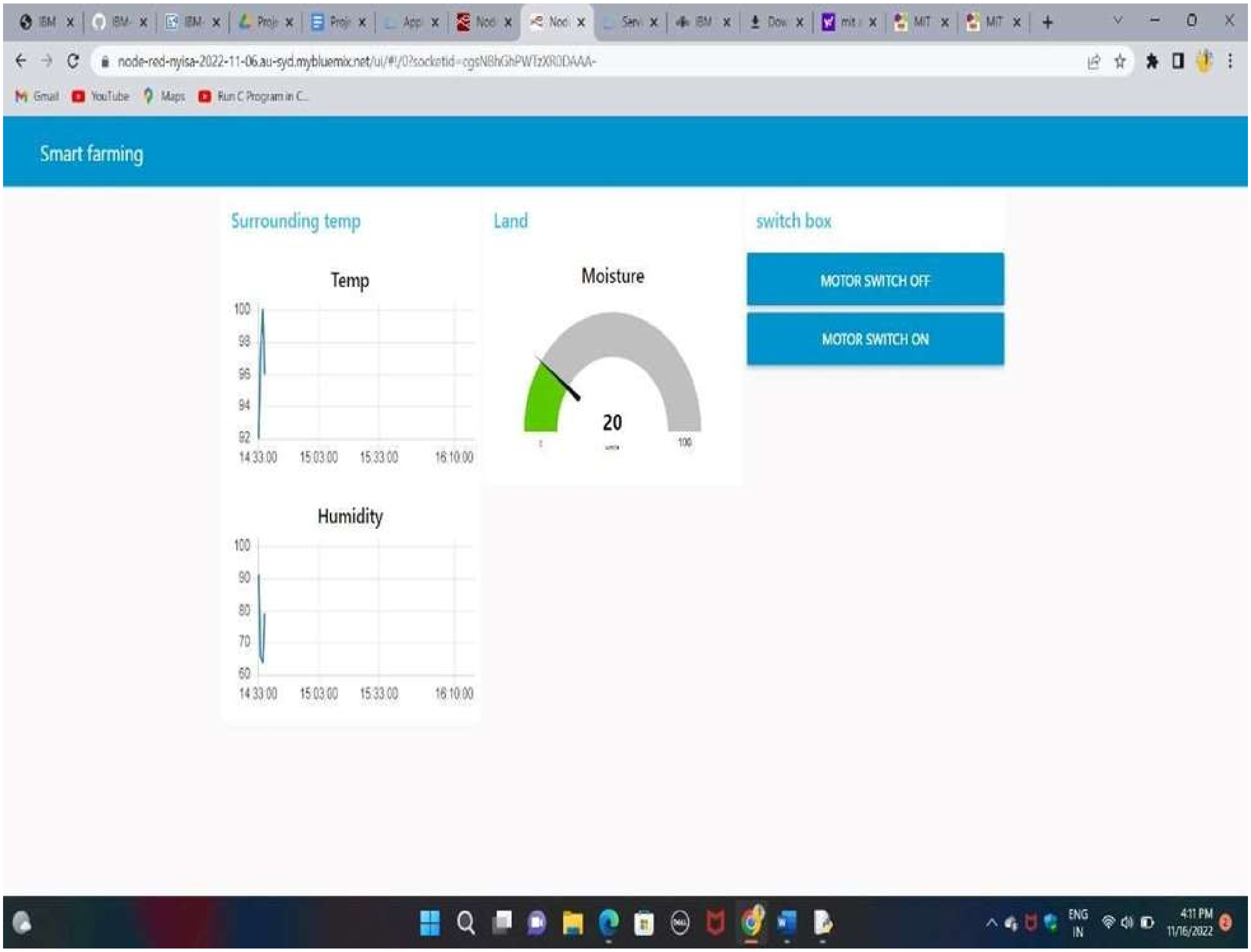
Temperature 60

Humidity 95

Moisture 90

# 9. Results

## 9.1 Performance Metrics



## *10.Advantages and Disadvantages*

### *Advantages*

- It is cost effective method.
- It delivers high quality crop production mobile operated pumps save cost of electricity.
- It allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc.

### *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 equipments require farmers to understand and learn the use of technology. This is major challange in adopting smart agriculture farming at large scale across the countries

## *11. Conclusion*

IoT based smart agriculture system for live monitoring of temperature and soil moisture and to control of motor has been implemented using NodeRed 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 agricultural yield and take efficient care of food production as the system will always provide helping hand to farmers for getting accurate live feed environmental temperature and soil moisture with more than 99% accurate results.

## *12. Future Scope*

Future work would be focused more on increasing sensors on this system to fetch more data specifically with regard to pest control and by also integrating GPS  module in this system to enhance this agriculture IoT technology to full-fledged agriculture precision ready product.

## *13. Appendix*

### *Source Code*

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

#Provide your IBM Watson Device Credentials
organization = "1nuzg6"
deviceType = "srimathi"
deviceId = "94868"
authMethod = "token"
authToken = "987654321"
```

```
# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led 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)

    data = { 'temp' : temp, 'Humid': Humid }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "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()
```

Github link: https://github.com/IBM-EPBL/IBM-Project-13296-1668575129

Project demo link: https://drive.google.com/file/d/17UB5Djvchq5OI0H-Z6g5mj6S7PXhl8GQ/view?usp=drivesdk