PROJECT REPORT

Project Name: SMART FARMER- IOT ENABLED SMART FARMING APPLICATION.

Team ID: PNT2022TMID24443

Team: CHUNDURUREDDY VENKATA VYSHNAVI
MOHAMMAD NUHIYA
DUDI CHARANI SRI
LAVANYA.P
NANDHIKA.G

1. INTRODUCTION
- Project Overview
- Purpose

2. LITERATURE SURVEY
2.1 Existing problem
2.2 References
2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION
3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution fit

4 REQUIREMENT ANALYSIS
4.1 Functional requirement
4.2 Non-Functional requirements

5 PROJECT DESIGN
5.1 Data Flow Diagrams & User Stories
5.2 Solution & Technical Architecture

6 PROJECT PLANNING & SCHEDULING
6.1 Sprint Planning & Estimation
6.2 Sprint Delivery Schedule

7 CODING & SOLUTIONING (Explain the features added in the project along with code)
7.1 Feature
7.2 Database Schema (if Applicable)

8 TESTING
8.1 Test Cases
8.2 User Acceptance Testing

9 RESULTS
9.1 Performance Metrics

10 ADVANTAGES & DISADVANTAGES

11 CONCLUSION

12 FUTURE SCOPE

13 APPENDIX
- Source Code
- GitHub & Project Demo Link

SMART FARMING

1. INTRODUCTION:

PROJECT OVERVIEW:

This is system that enables framers to monitor and their forms with a webbased application build with Node-RED.

It uses the IBM IOT Watson cloud platform as its Backend.

PURPOSE:

Smart Farming reduce the ecological foodprint of farming. Minimized or site specific application of inputs, such as fertilizers and pesticides ,in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases.

2. LITERATURE SURVEY:

2.1 EXISTING PROBLEM:

The biggest challenges faced by IoT in the agricultural sector are lack of information, high adoption costs , and security concers , etc. Most of the farmers are not aware of the implementation of IoT in agriculture.

2.2 REFERENCES:

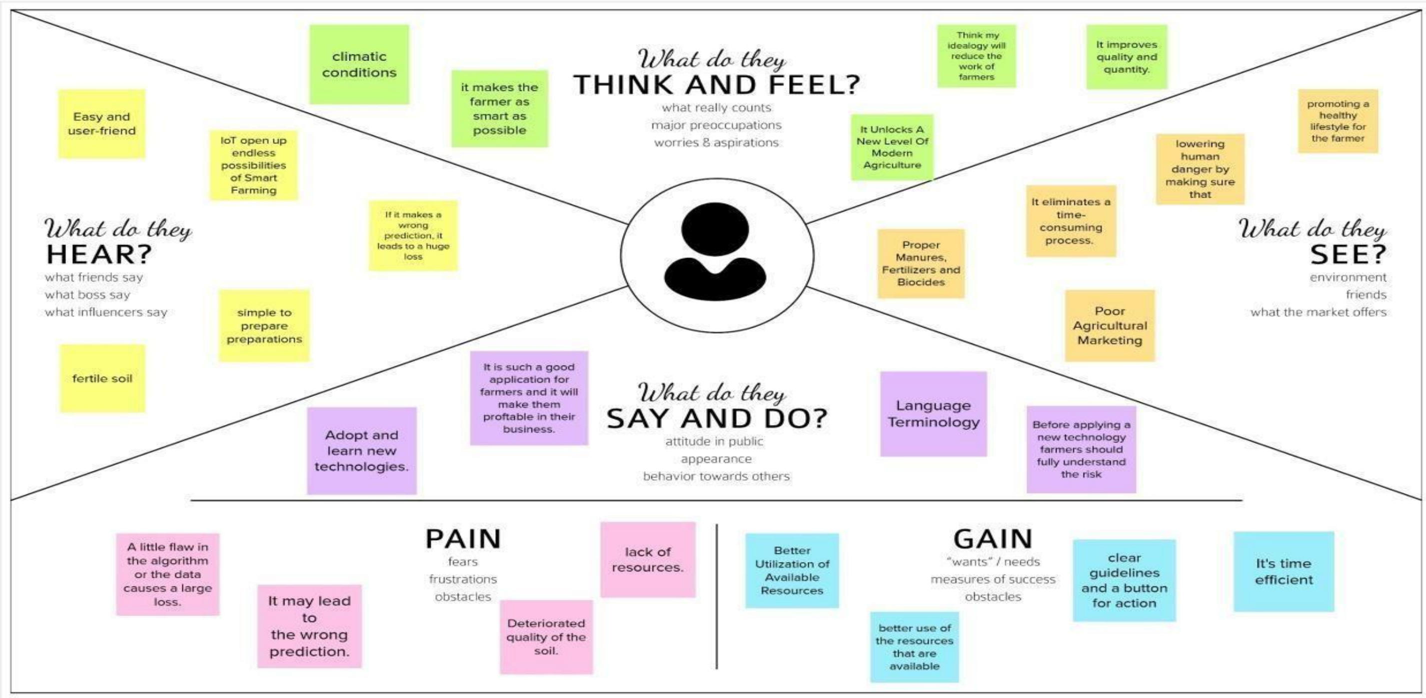
It is the application of modern ICT (Information and Communication Technologies) into agriculture. In IOT- based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.). The farmers can monitor the field conditions from anywhere.

2.3 PROBLEM STATEMENT DEFINITION:

Overuse of pesticides and fertilizer in agricultural fields leads to destruction of the crop as well as reduces the efficiency of the field increasing the soil vulnerability toward pest. IoT applications may be used to update the farmer/ user about type & quantity of pesticide required by the crop.

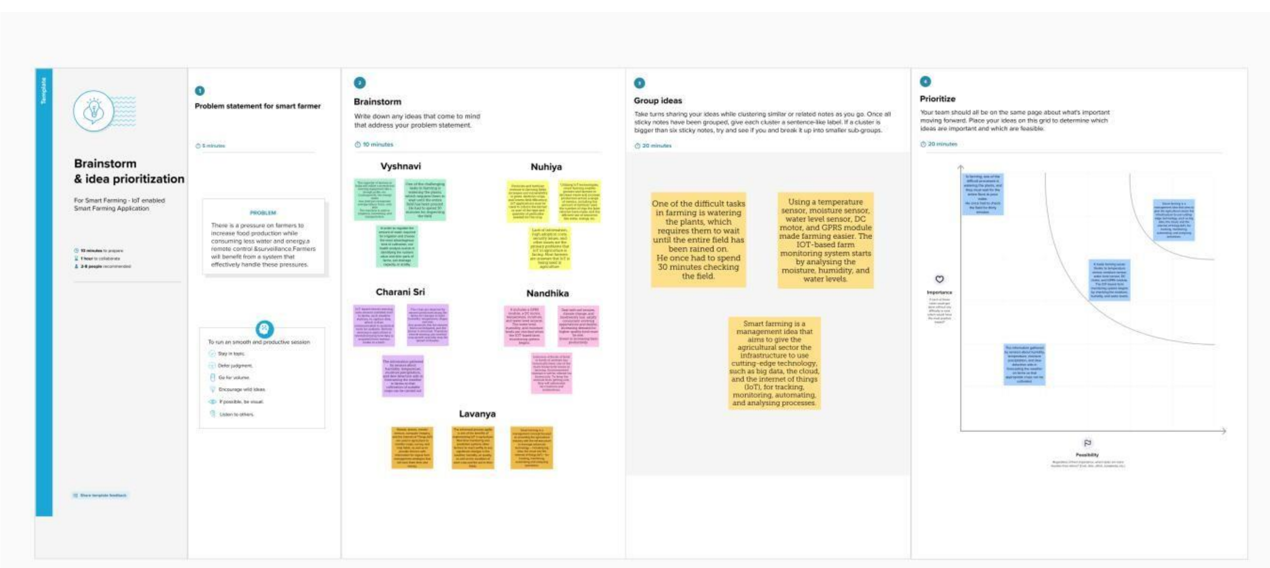
3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:



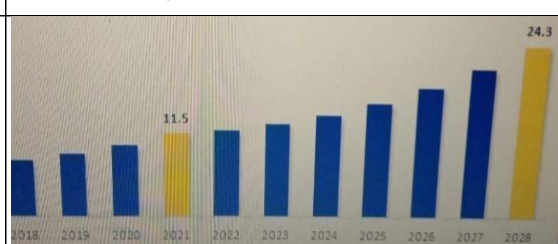
3.2 IDEATION & BRAINSTORMING:

Ideation is the create process of generating, developing, and communicating new ideas, where an
is idea understood as a basic element of thought that can beeither visual, concrete, or abstract

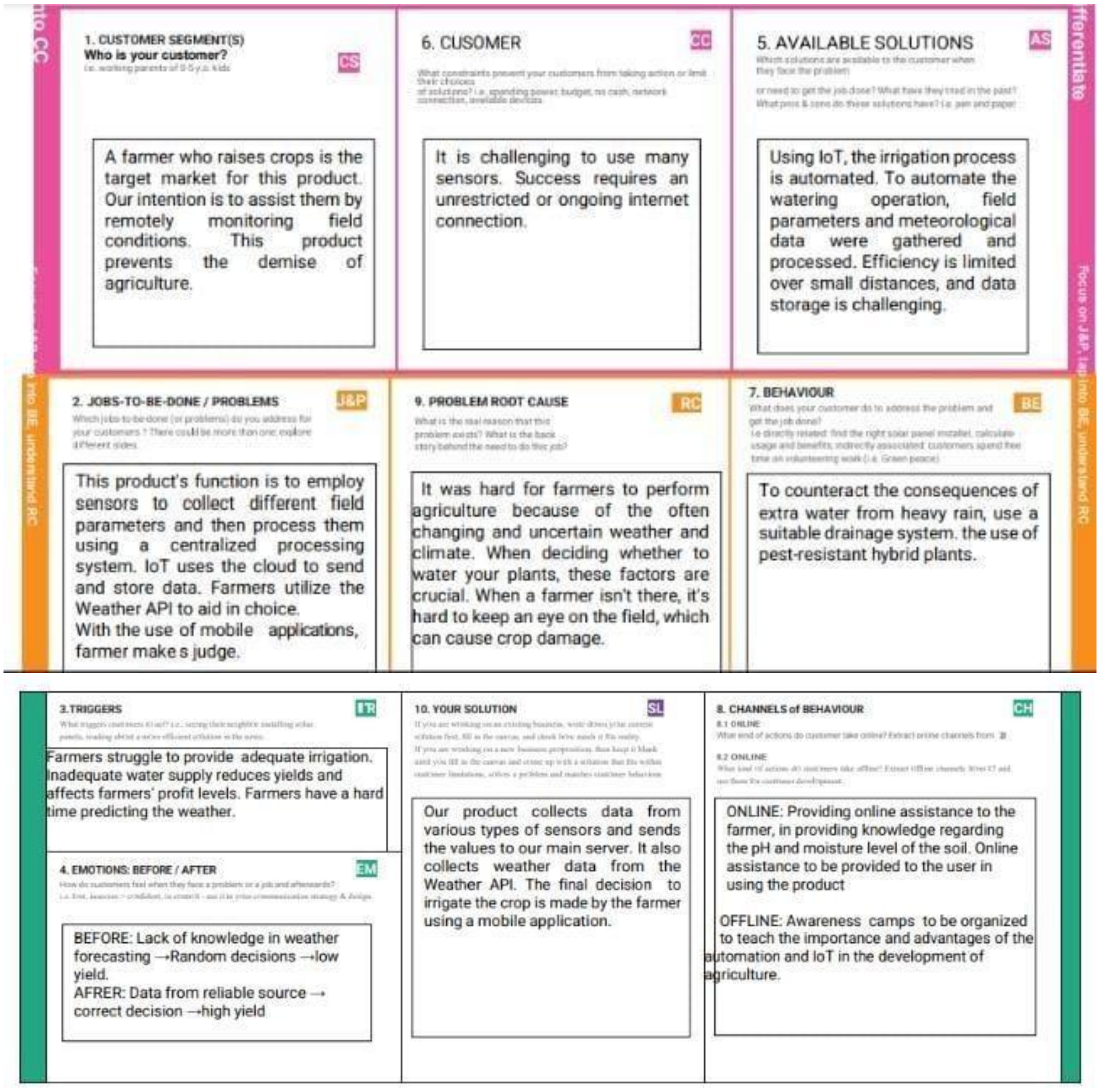


3.3 Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	○ Most of the farmers use large portions of farming land and it becomes very difficult to reachand track each corner of largelands. Sometime there is a possibility of uneven water sprinkles.
2.	Idea / Solution description	○ Smart Farming has enabled farmers to reduce waste and enhance productivity with the help of sensors (light, humidity, temperature, soil moisture, etc..)
3.	Novelty / Uniqueness	○ Role of SENSORS : IOT smart agriculture products are designed to help monitor crop fields using sensors and by automating irrigation systems. ○ As a result, farmers and associated brands can easilymonitor the field conditions from anywhere without any hassle
4.	Social Impact / Customer Satisfaction	○ Water conservation ○ Saves lot of time ○ Increased quality of production ○ insight
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	○ Scalability in smart farming refers to the adaptability of a system to increase the capacity, the number of technology devices such as sensors and actuators.

3.4 PROBLEM SOLUTION FIT:



1. CUSTOMER SEGMENT(S) CS

Who is your customer?

A farmer who raises crops is the target market for this product. Our intention is to assist them by remotely monitoring field conditions. This product prevents the demise of agriculture.

6. CUSOMER CC

It is challenging to use many sensors. Success requires an unrestricted or ongoing internet connection.

5. AVAILABLE SOLUTIONS AS

Using IoT, the irrigation process is automated. To automate the watering operation, field parameters and meteorological data were gathered and processed. Efficiency is limited over small distances, and data storage is challenging.

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

This product's function is to employ sensors to collect different field parameters and then process them using a centralized processing system. IoT uses the cloud to send and store data. Farmers utilize the Weather API to aid in choice.
With the use of mobile applications, farmer make s judge.

9. PROBLEM ROOT CAUSE RC

It was hard for farmers to perform agriculture because of the often changing and uncertain weather and climate. When deciding whether to water your plants, these factors are crucial. When a farmer isn't there, it's hard to keep an eye on the field, which can cause crop damage.

7. BEHAVIOUR BE

To counteract the consequences of extra water from heavy rain, use a suitable drainage system. the use of pest-resistant hybrid plants.

3.TRIGGERS TR

Farmers struggle to provide adequate irrigation. Inadequate water supply reduces yields and affects farmers' profit levels. Farmers have a hard time predicting the weather.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE: Lack of knowledge in weather forecasting →Random decisions →low yield.
AFRER: Data from reliable source → correct decision →high yield

10. YOUR SOLUTION SL

Our product collects data from various types of sensors and sends the values to our main server. It also collects weather data from the Weather API. The final decision to irrigate the crop is made by the farmer using a mobile application.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

8.2 ONLINE

ONLINE: Providing online assistance to the farmer, in providing knowledge regarding the pH and moisture level of the soil. Online assistance to be provided to the user in using the product

OFFLINE: Awareness camps to be organized to teach the importance and advantages of the automation and IoT in the development of agriculture.

Project Design Phase-II

4. Solution Requirements (Functional & Non-functional)

4.1 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	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Log in to system	Check Roles of Access. Check Credentials
FR-4	Manage Modules	Manage System Admins Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details Humidity details
FR-6	Log out	Exit

4.2 Non-Functional Requirements:

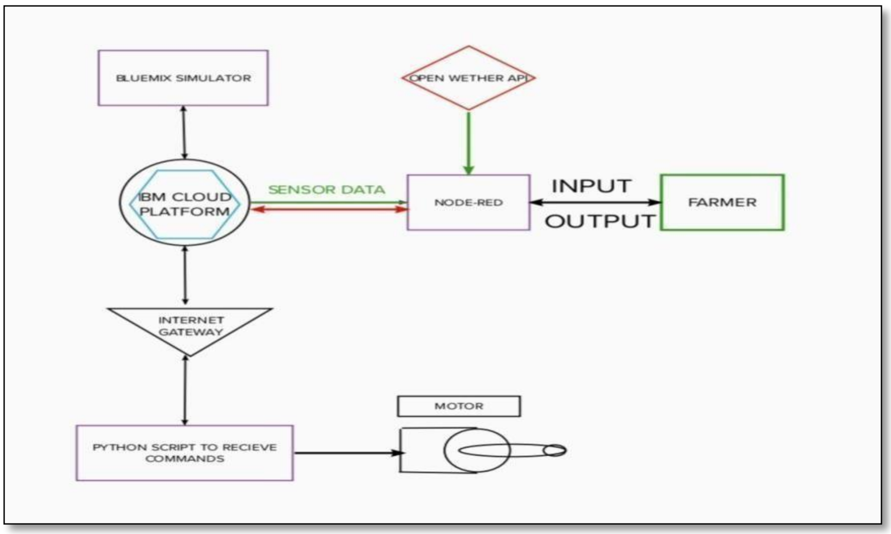
Following are the non-functional requirements of the proposed solution.

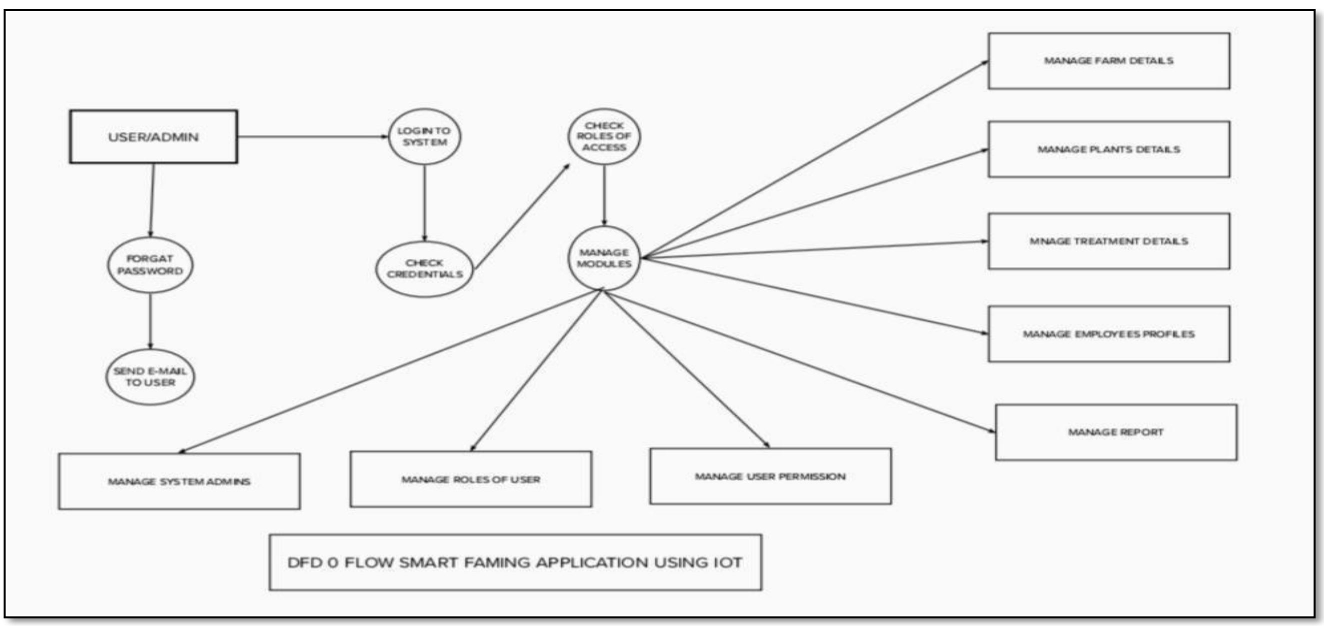
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability is defined as the ability to learn quickly, use something effectively, remember something, operate something without making a mistake, and enjoy something.
NFR-2	**Security**	Private and confidential information must be kept secure at all times, including during collection, processing, and storage.
NFR-3	**Reliability**	A superior cost-to-reliability trade-off is achieved with shared protection.To prevent agricultural service interruptions, the approach employs specialised and shared protectionmethods.
NFR-4	**Performance**	It will be more effective to monitor farming operations overall if integrated sensors are used to measure soil and ambient characteristics.
NFR-5	**Availability**	By tying information about crops, weather, and equipment together, it is feasible to automatically alter temperature, humidity, and other factors in farming equipment.

5. **PROJECT DESIGN**:

5.1 DATA-FLOW DAIGRAMS AND USER STORIES:

A Data Flow Diagram (DFD) is a traditional visual representation of the informationflows 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.





5.2 SOLUTIONS AND TECHNICAL ARCHITECTURAL:

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

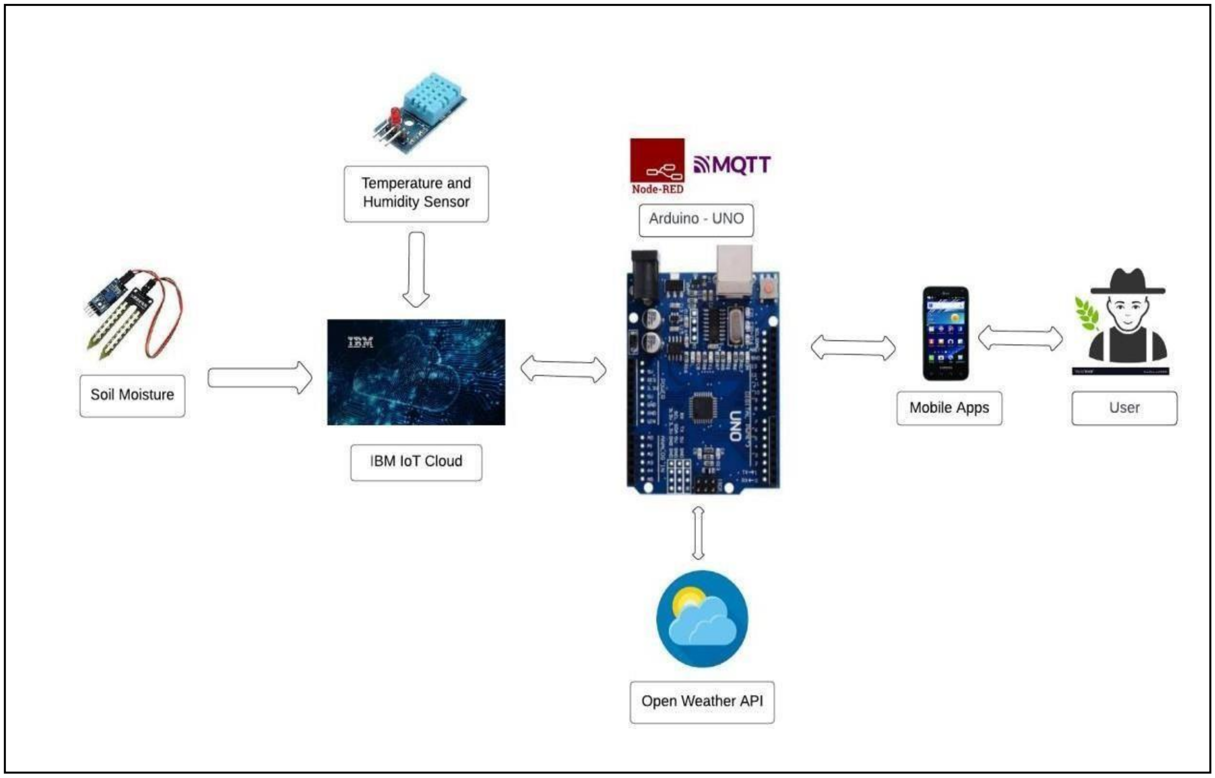


Table-1: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	MIT app
2.	Application Logic-1	Logic for a process in the application	Node red/IBM Watson/MIT app
3.	Application Logic-2	Logic for a process in the application	Node red/IBM Watson/MIT app
4.	Application Logic-3	Logic for a process in the application	Node red/IBM Watson/MIT app
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM cloud.
7.	Temperature sensor	Monitors the temperature of the crop	
8.	Humidity sensor	Monitors the humidity	
9.	Soil moisture sensor (Tensiometers)	Monitors the soil temperature	
10.	Weather sensor	Monitors the weather	.
11.	Solar panel		.
12.	RTC module	Date and time configuration	
13.	Relay	To get the soil moisture data	

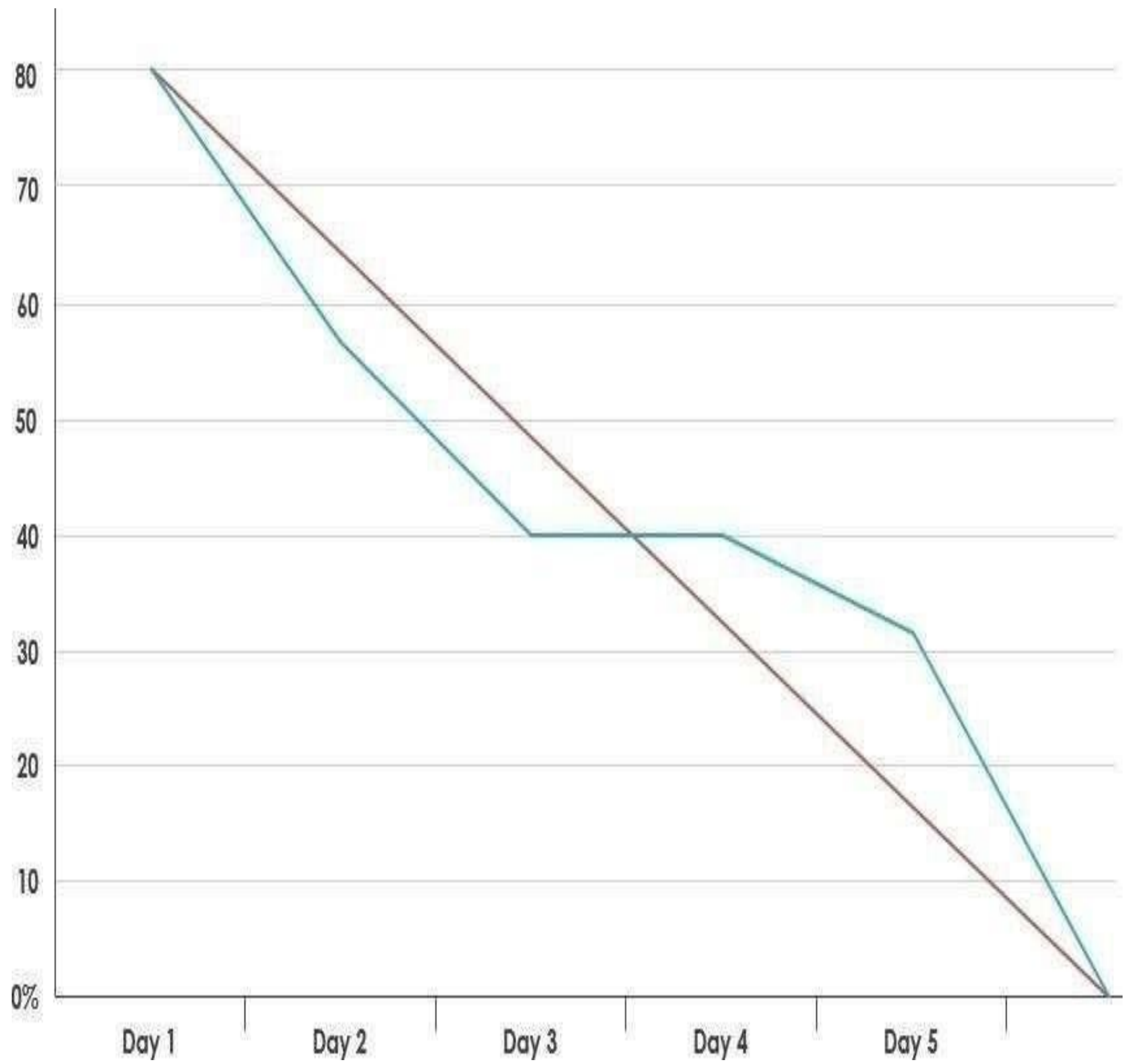
Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	MIT app,Node-Red	Software
2.	Scalable Architecture	Drone technology, pesticide monitoring ,Mineral identification in soil	Hardware

6. PROJECT PLANNING AND SCHEDULING:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Story Points	Team Members
Sprint-1	Hardware	USN-1	Sensors and wi-fi module with python code.	High	2	Vyshnavi, Nuhiya, Charani sri, Lavanya, Nandhika
Sprint-2	Software	USN-2	IBM Watson IoT platform, Workflows for IoTscenarios using Node-red	High	2	Vyshnavi, Nuhiya, Charani sri, Lavanya, Nandhika
Sprint-3	MIT app	USN-3	To develop a mobile application using MIT	High	2	Vyshnavi, Nuhiya, Charani sri, Lavanya, Nandhika
Sprint-4	Web UI	USN-4	To make the user to interact with software.	High	2	Vyshnavi, Nuhiya, Charanisri, Lavanya, Nandhika

Burndown Chart:



7. CODING & SOLUTIONS:

7.1 FEATURE :

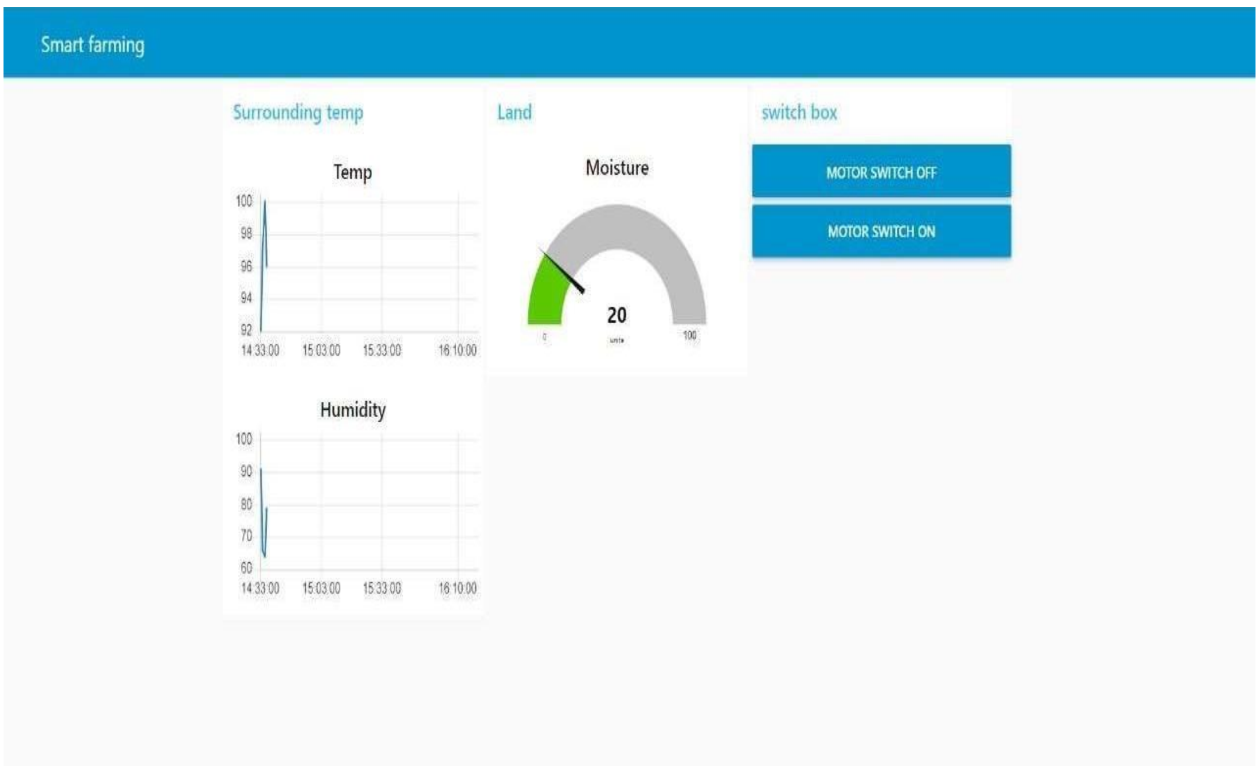
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


organization = "uojvy9"
deviceType="NodeMCU"
deviceid="12345"
authMethod="token"
authToken="12345678"


def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="motoron"):
        print("Motor is on")
    elif(m=="motoroff"):
        print("Motor is off")
    else :
    `   print("plese 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
```

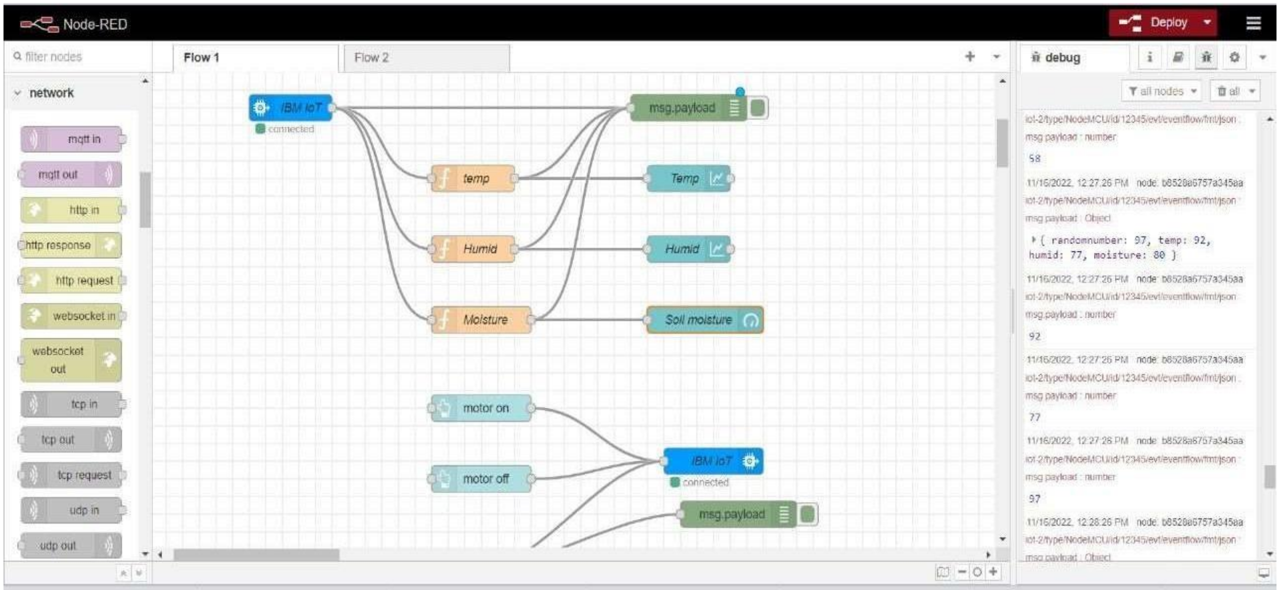
.

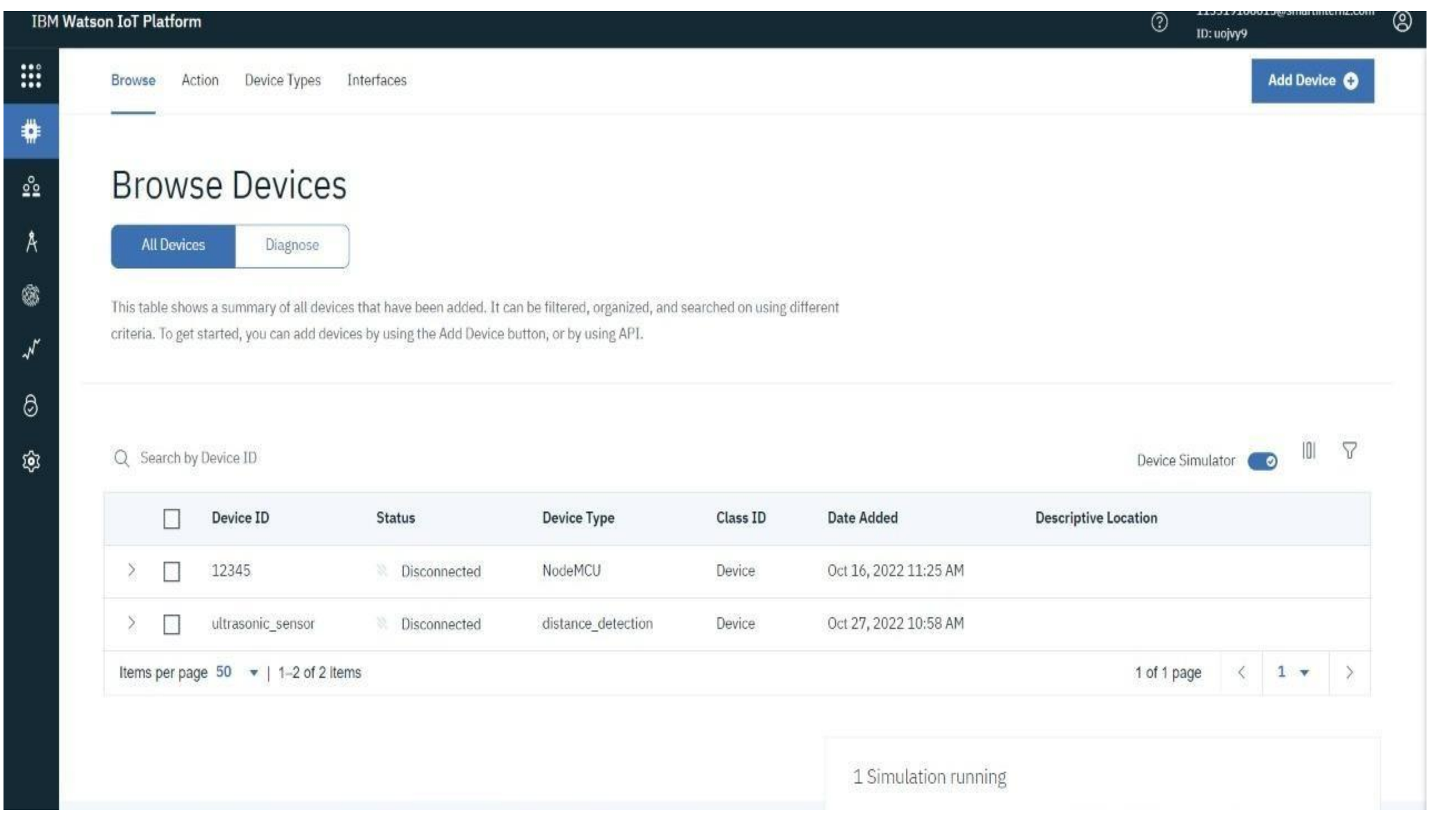
8. TESTING:

8.1 TEST CASE:

Web application using Node-RED.




Node-RED
Deploy
filter nodes
Flow 1
Flow 2
debug
network
mqtt in
mqtt out
http in
http response
http request
websocket in
websocket out
tcp in
tcp out
tcp request
udp in
udp out
IBM IoT
connected
msg.payload
temp
Humid
Moisture
Temp
Humid
Soil moisture
motor on
motor off
IBM IoT
connected
msg.payload
all nodes
all
{ randomnumber: 97, temp: 92, humid: 77, moisture: 80 }

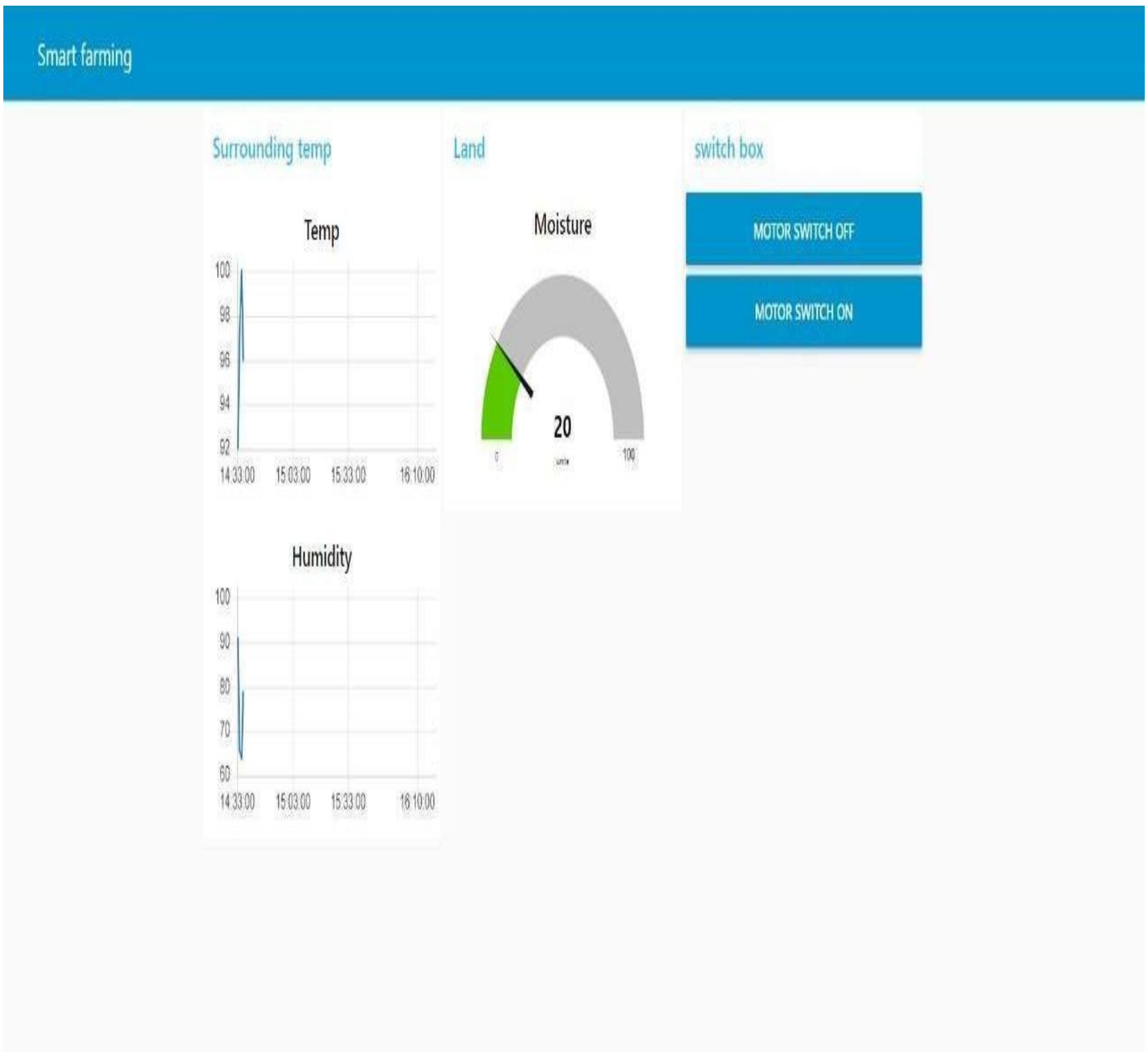

IBM Watson IoT Platform
ID: uojvy9
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345
Disconnected
NodeMCU
Device
Oct 16, 2022 11:25 AM
ultrasonic_sensor
Disconnected
distance_detection
Device
Oct 27, 2022 10:58 AM
Items per page 50
1–2 of 2 items
1 of 1 page
1 Simulation running

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


organization = "uojvy9"
deviceType="NodeMCU"
deviceid="12345"
authMethod="token"
authToken="12345678"


def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="motoron"):
        print("Motor is on")
    elif(m=="motoroff"):
        print("Motor is off")
    else :
    `   print("plese 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
```

8.2 User Acceptance Testing

9. RESULT:

9.1 Performance Metrics



10. ADVANTAGES AND DISADVANTAGES:

10.1 ADVANTAGES:

- All the data like climatic conditions and changes in them, soil orcrop conditions everything can be easily monitored.
- Risk of crop damage can be lowered to a greater extent.
- Many difficult challenges can be avoided making the process automated and the quality of crops can be maintained.
- The process included in farming can be controlled using the web applications from anywhere, anytime.

10.2 DISADVANTAGES:

- Smart Agriculture requires internet connectivity continuously, butrural parts cannot fulfl this requirement.
- Any faults in the sensors can cause great loss in the agriculture,due to wrong records and the actions of automated processes.
- IOT devices need much money to implement.

10. CONCLUSION:

An IOT based smart agriculture system using Watson IOT platform, Watson simulator, IBM cloud and Node-RED.

11.FUTURE SCOPE:

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

12. APPENDIX:

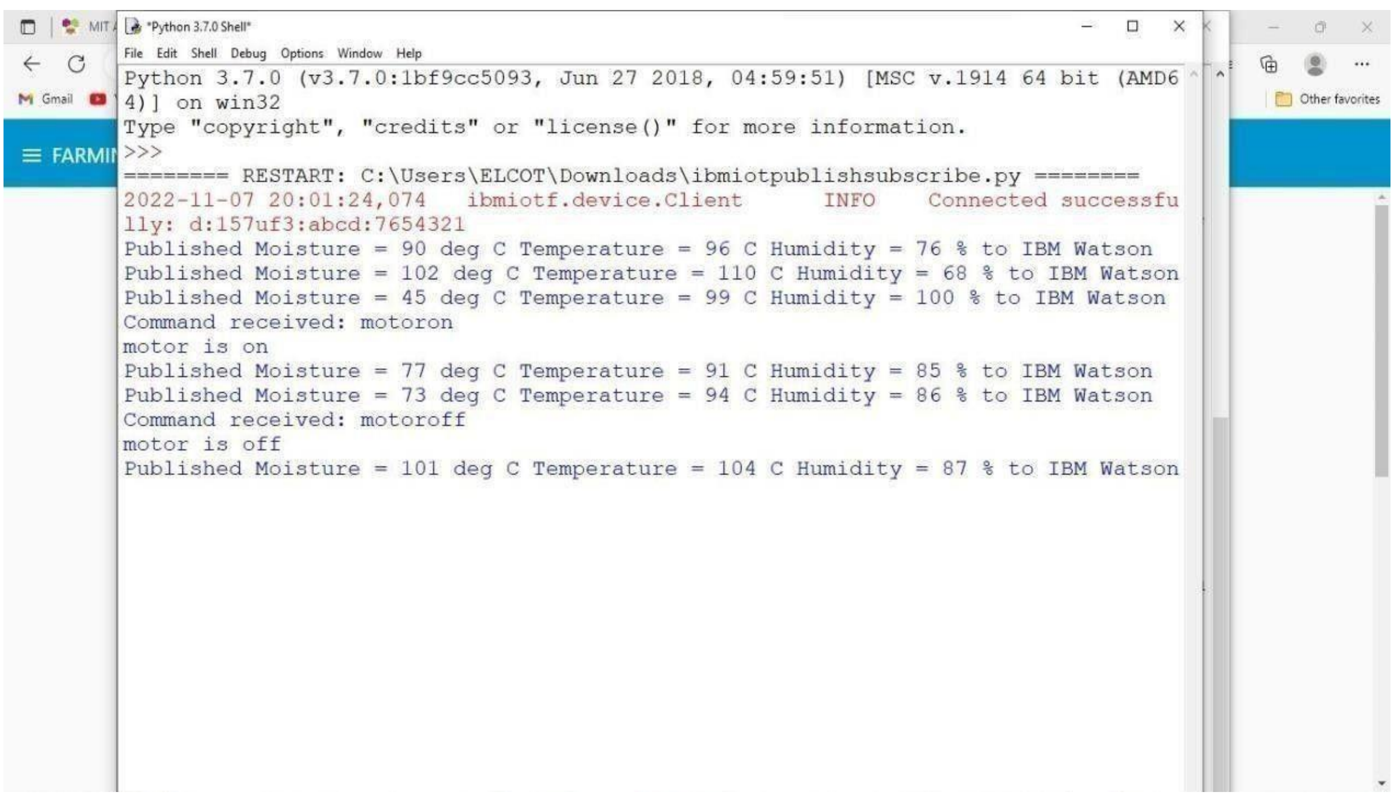
SOURCE CODE:

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

#Provide your IBM Watson Device
organization =  "uojvy9"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken ="12345678"
# Initialize GPIO
def  myCommandCallback(cmd):
    print("Commandreceived: %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 aneventof type "greeting" 10 times
deviceCli.connect()
while True:
#Get Sensor Data fromDHT11
   temp=random.randint(90,110)
   Humid=random.randint(60,100)
   Mois=random. randint(20,120)
   data = { 'temp' : temp, 'Humid': Humid ,'Mois': Mois}
#print data
def myOnPublishCallback():
  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()
```

OUTPUT:



Github link: **https://github.com/IBM-EPBL/IBM-Project-53868-1661504055.git**

DEMO LINK:
https://1drv.ms/v/s!Av3mbTmqylcNgxXzFIMG0y-pf5Eu?e=o4Vkb4

`

THANK YOU