PROJECT REPORT

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

Team ID **:** PNT2022TMID24462

Team Members **:** **KAMIREDDY CHATHURYA REDDY**

GANAPATHI CHIRUDIVYA

GANDHAVALLI HARIVANDANA

DIVYA.B

SMART FARMING

1. INTRODUCTION:

PROJECT OVERVIEW:

This is system that enables framers to monitor and their forms with a web based 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:

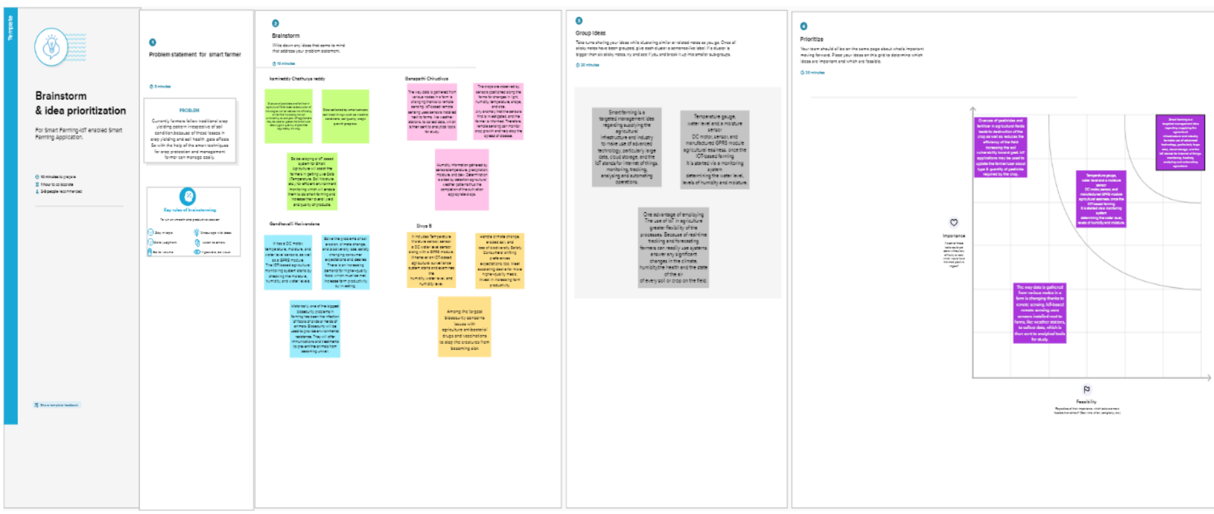
3.1 EMPATHY MAP CANVAS:

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 be either visual, concrete, or abstract.

Brainstorming is a group creative technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members.

IDEATION PROCESS



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)	To provide an effective decision support system employing a wireless sensor network that manages various agricultural activities and provides pertinent farm information on temperature, humidity, and soil moisture content. The weather is to blame for the rising water level. There are many distractions for farmers, which is bad for agriculture.
2.	Idea / Solution description	Solutions for Smart Agricultural Systems offer an integrated IOT platform in the agricultural sector that enables farmers to use sensors, smart gateways, and monitoring systems to gather data, manage numerous farm characteristics, and analyse real-time data to make educated decisions.
3.	Novelty / Uniqueness	The use of IOT principles in agriculture has been the focus of several prominent researchers working toward smart farming. But there are still a number of difficulties that have not yet found an appropriate solution. This study attempts to highlight prior work and unresolved issues in IOT-based agriculture.
4.	Social Impact / Customer Satisfaction	Decreases the pay for workers who are employed in the agricultural sector. It helps people to save lots of time. By enriching the customer experience overall, and also IOT can help strengthen customer relationships.
5.	Business Model (Revenue Model)	Farmers are charged a monthly subscription for the prediction and recommendation of irrigation scheduling based on sensor metrics such as temperature, humidity, and soil moisture.
6.	Scalability of the Solution	Scalability in smart farming refers to a system's ability to expand its capacity, such as the number of technological components like sensors and actuators, while enabling for quick analysis.

3.4 PROBLEM SOLUTIONS FIT :

Project Title: – IOT Enabled Smart Farming Application **Project Design Phase-I - Solution Fit** **Team ID:** [illegible]

Define CS, fit into CC	1.CUSTOMER SEGMENT(S) CS	6. CUSTOMER CONSTRAINTS CC	5. AVAILABLE SOLUTIONS	Explore AS, differentiate
	Farmers and Large landowners must understand the potential of IoT market for agriculture by installing smart technologies to increase competitiveness and sustainability in their production.	IoT in the agricultural sector are lack of information, high adoption costs, and security concerns, etc. Most of the farmers are not aware of the implementation of IoT in agriculture.	Remote crop and soil monitoring. With the help of smart farming system, moisture and fertility of soil along with crops growth rate can be monitored remotely through real time animation and graphics via a smartphones.	

	2. JOBS-TO-BE-DONE / PROBLEMS J&P	9. PROBLEM ROOT CAUSE RC	7. BEHAVIOUR BE	
	1.Lack of Security 2. Data collection frequency	1.Since IoT devices interact with older equipment they have access to the internet connection, there is no guarantee that they would be able to access drone mapping data or sensor readouts by taking benefit of public connection. 2. Because of the high variety of data types in the agricultural industry, ensuring the optimal data collection frequency can be problematic. The data from field-based, aerial and environmental sensors, apps, machinery, and equipment, as well as processed analytical data, can be a subject of restriction and regulations.	Smart farming applications should be tailored for use in the field. A business owner or farm manager should be able understand the working of it and get into the details about the issues.	

Identify strong TR & EM	3. TRIGGERS TR	10. YOUR SOLUTION SL	8.CHANNELS of BEHAVIOUR CH	Identify strong TR & EM
	Poor Internet Connectivity in Farm Most farms are located in remote locations where internet connectivity might not be strong enough to facilitate fast transmission speeds 4. EMOTIONS: BEFORE / AFTER EM BEFORE: Lack of knowledge in weather forecasting. Random decisions , low yield, lack of knowledge in hardware AFRER: Data from reliable source ,correct decision ,high yield	Our product provides better Crop Monitoring it involves the use of sensors, drones, and satellites to monitor crop health and identify locations requiring attention. Crop monitoring systems also include all data such as crop health, humidity, rainfall, temperature, and more.	ONLINE: Providing online assistance to the farmer, regarding the knowledge about the device, and by giving more knowledge about iot and assistance about the crop monitoring. OFFLINE: By conducting Awareness camps to teach the importance and advantages of the automation and IoT in the development of agriculture	

4.REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL ANALYSIS:

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 Credentials Check Roles of Access.
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 NON FUNCTIONAL REQUIREMENTS:

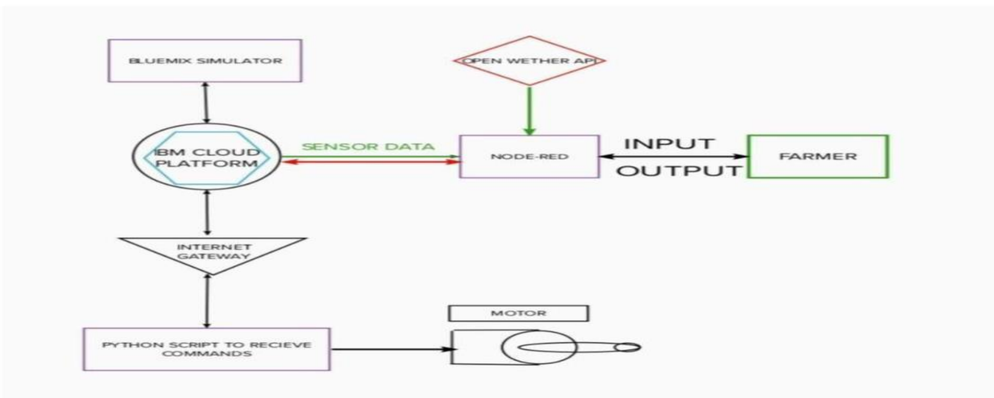
Following are the non-functional requirements of the proposed solution.

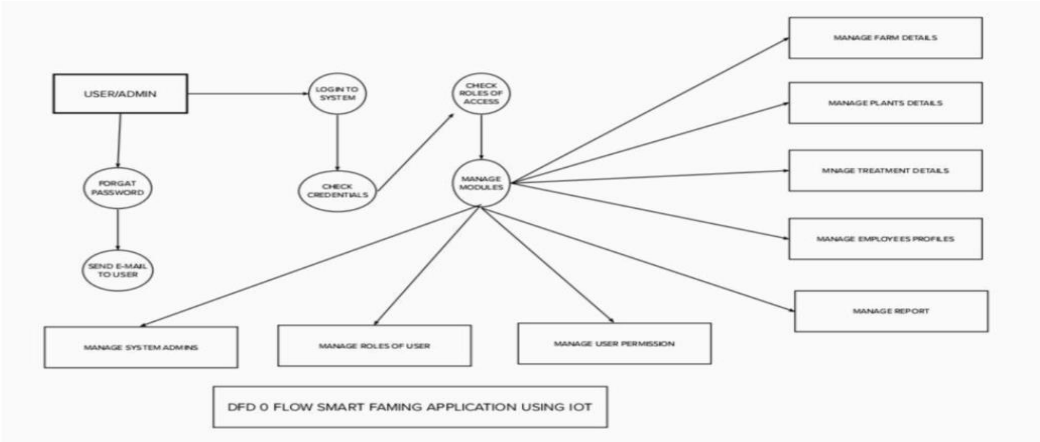
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Time consumability is less, Productivity is high.
NFR-2	**Security**	It has low level of security features due to integration of sensor data.
NFR-3	**Reliability**	Accuracy of data and hence it is Reliable.
NFR-4	**Performance**	Performance is high and highly productive.
NFR-5	**Availability**	With permitted network connectivity the application is accessible
NFR-6	**Scalability**	It is perfectly scalable many new constraints can be added

5 PROJECT DESIGN:

5.1 DATA FLOW DIAGRAMS AND USER STORIES:

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.





5.2 SOLUTIONS AND TECHNICAL ARCHITECTURAL:

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

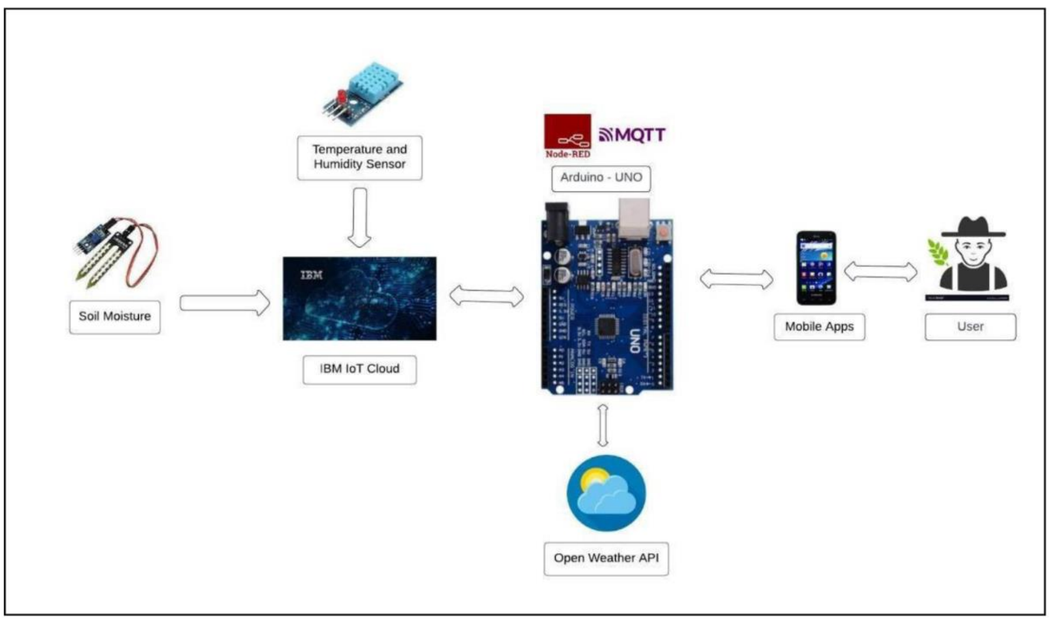


Table-1 : Components & Technologies:

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	

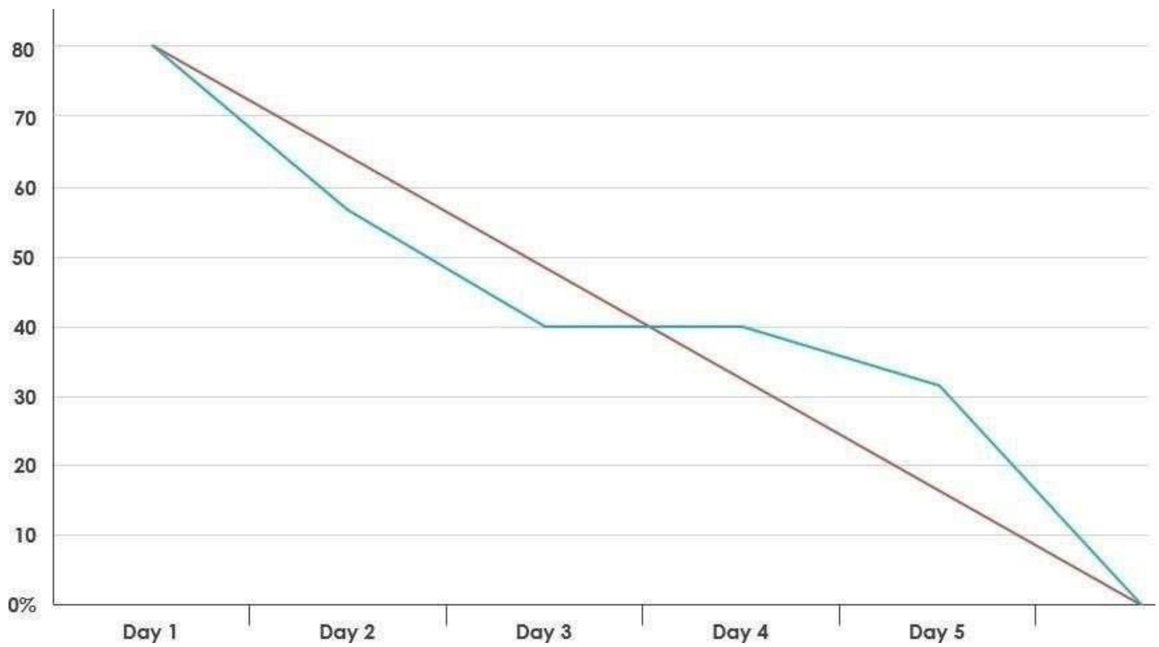
Table-2: Application Characteristics:

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	Story Points	Priority	Team Members
Sprint-1	Hardware	USN-1	Sensors and wi-fi module with python code.	2	High	Kamireddy chathurya Reddy,Ganapathi Chirudivya,Gandhavalli Harivandana,Divya B
Sprint-2	Software	USN-2	IBM Watson IoT platform, Workflows for IoT scenarios using Node-red	2	High	Kamireddy chathurya Reddy,Ganapathi Chirudivya,Gandhavalli Harivandana,Divya B
Sprint-3	MIT app	USN-3	To develop a mobile application using MIT	2	High	Kamireddy chathurya Reddy,Ganapathi Chirudivya,Gandhavalli Harivandana,Divya B
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Kamireddy chathurya Reddy,Ganapathi Chirudivya,Gandhavalli Harivandana,Divya B

Burndown Chart:



7 CODING & SOLUTIONS:

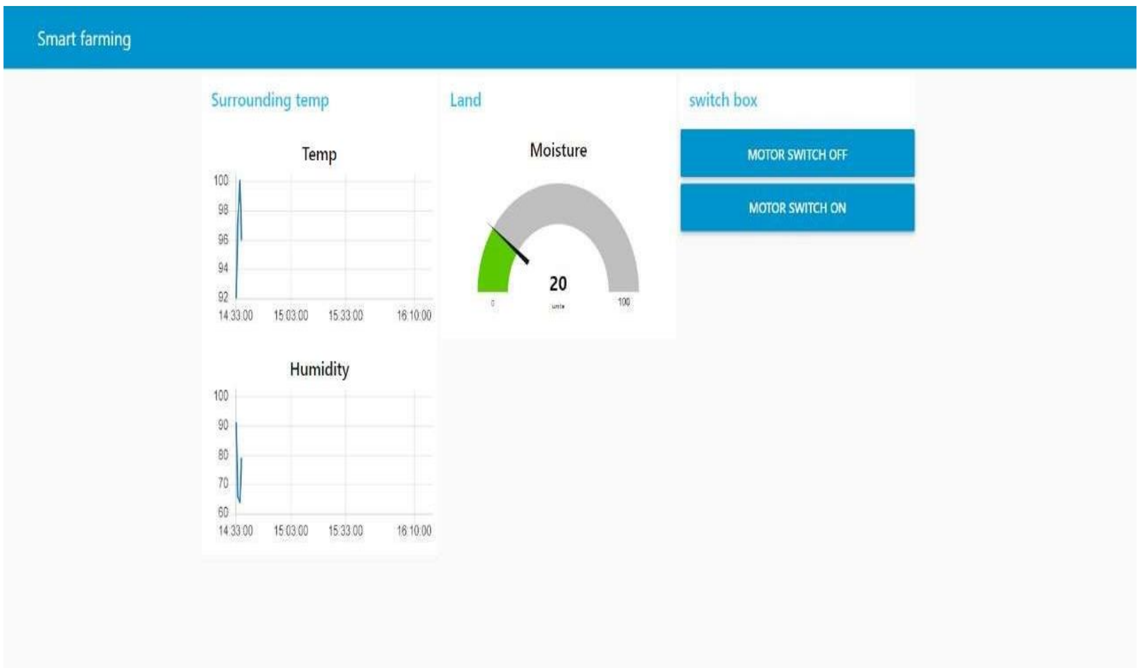
7.1 FEATURE :

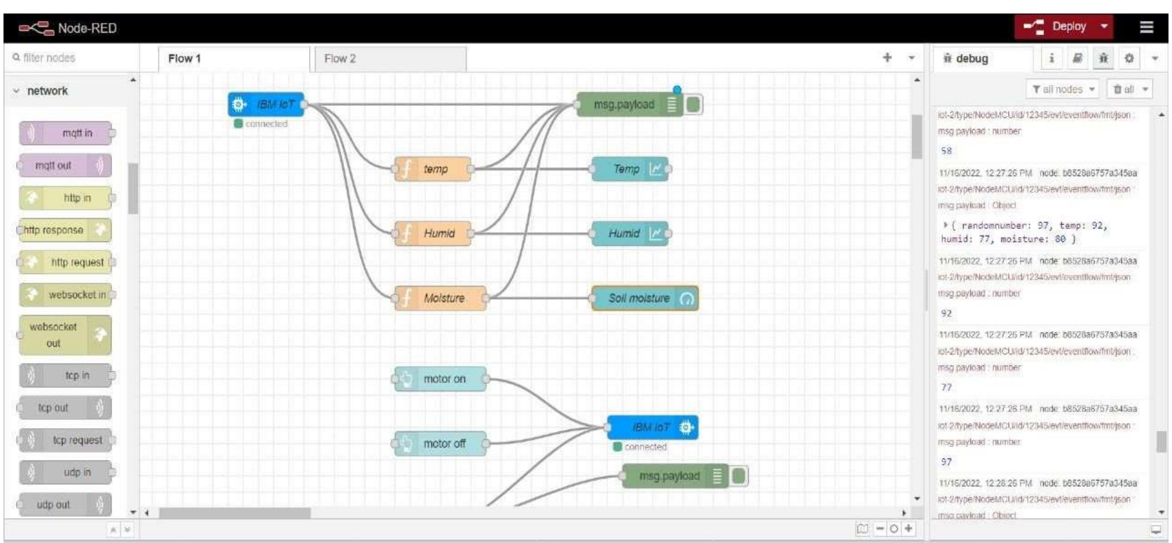
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


organization = "2pys2a"
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
```

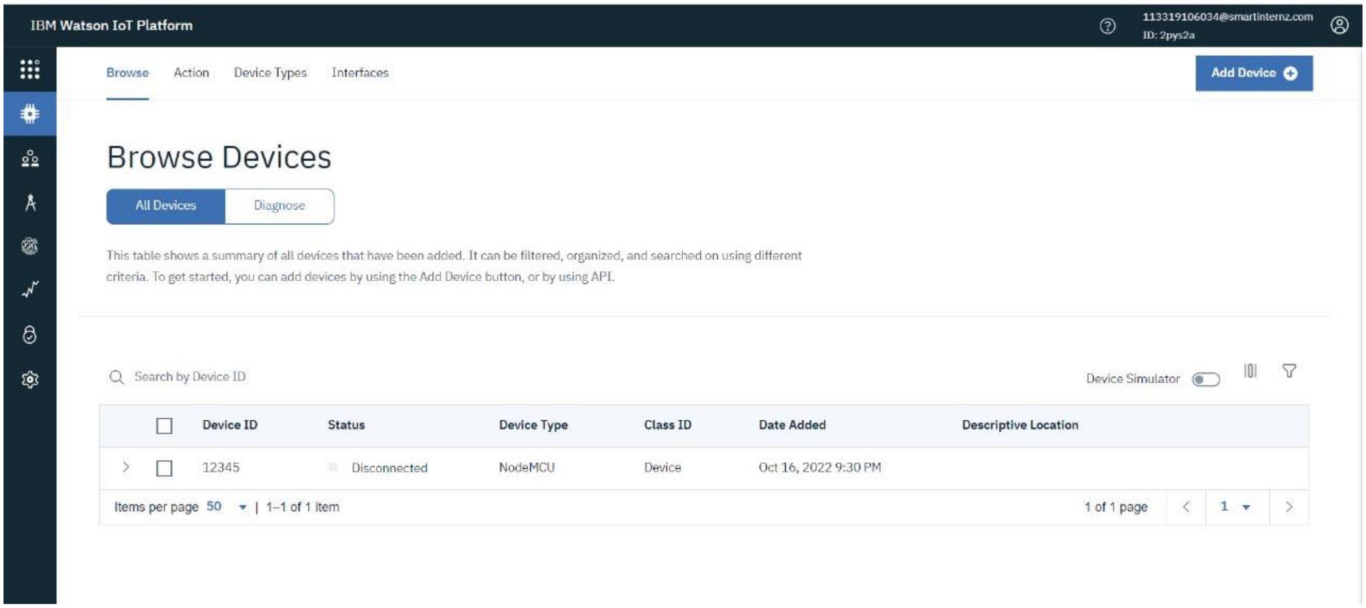
8 TESTING:

8.1 TEST CASE:

Web application using Node-RED.





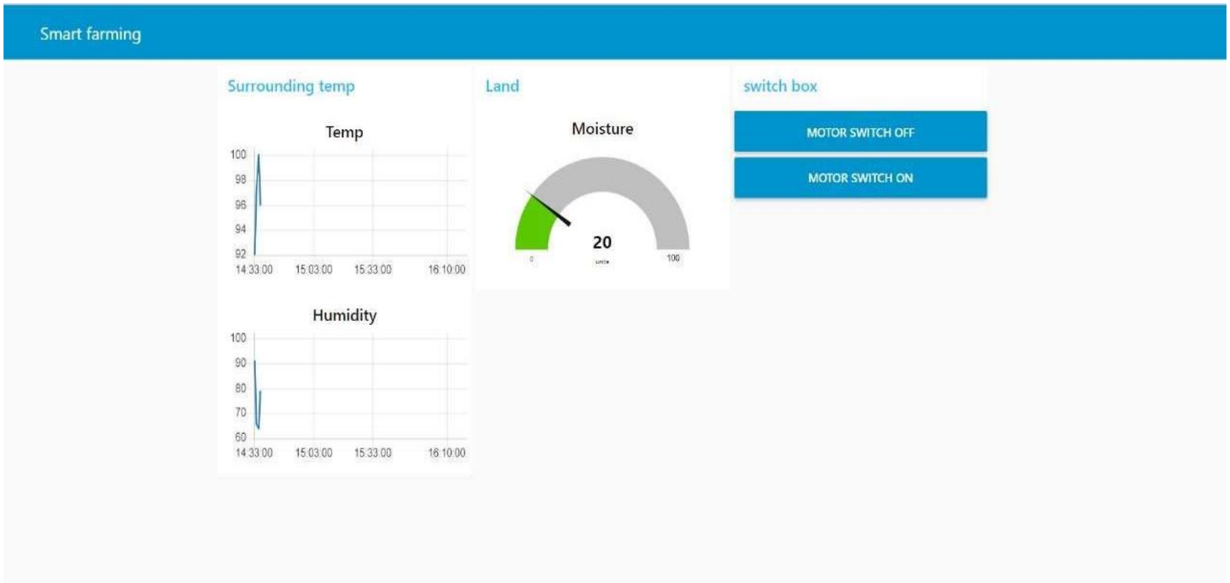



```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


organization = "2pys2a"
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 or crop 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, but rural parts cannot fulfil 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.

11 CONCLUSION:

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

12 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.

13 APPENDIX:

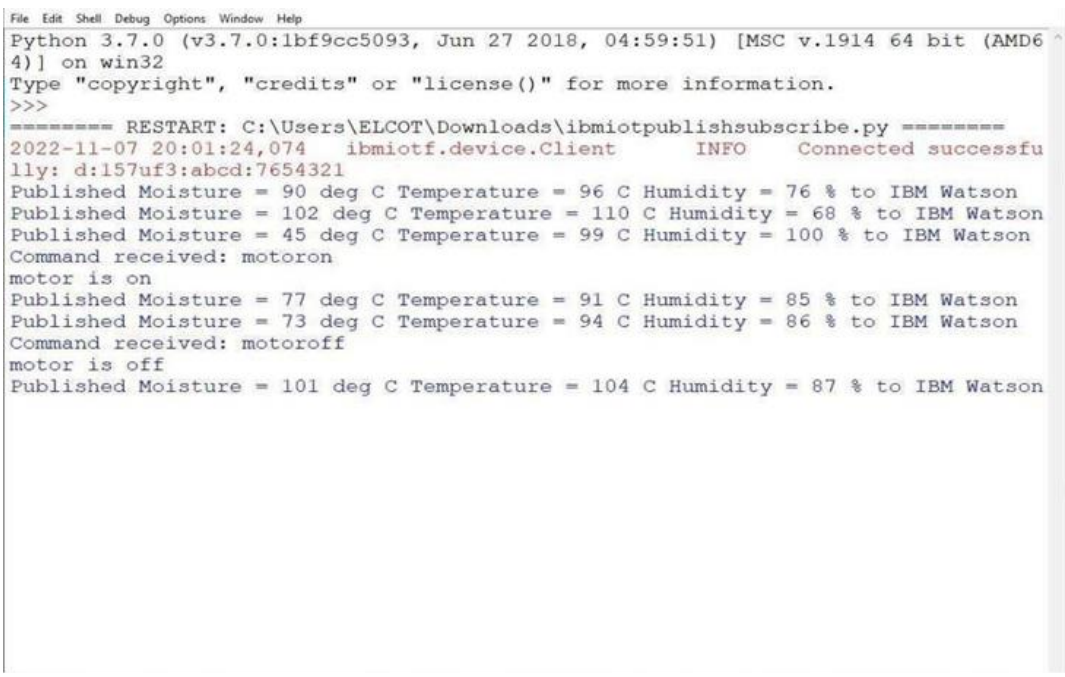
SOURCE CODE:

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

#Provide your IBM Watson Device
organization =  "c2pys2a"
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:



```
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py ========
2022-11-07 20:01:24,074   ibmiotf.device.Client      INFO    Connected successfu
lly: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

GITHUB LINK : **https://github.com/IBM-EPBL/IBM-Project-53584-1661419143.git**

DEMO LINK: https://drive.google.com/file/d/1_pOYDmzyKFLsTjOQj9_w-aNf-jq9xeg_/view?usp=drivesdk

THANK YOU…..