NAALAIYA THIRAN PROJECT - 2022
19ECI01-PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

PROJECT REPORT

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

Team ID: PNT2022TMID38789

Team:

GAYATHRI.V - TEAM LEAD

SUDHARSANA.S

SANTHIYA.S

HAZIRAMA.S

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CONTENTS

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.

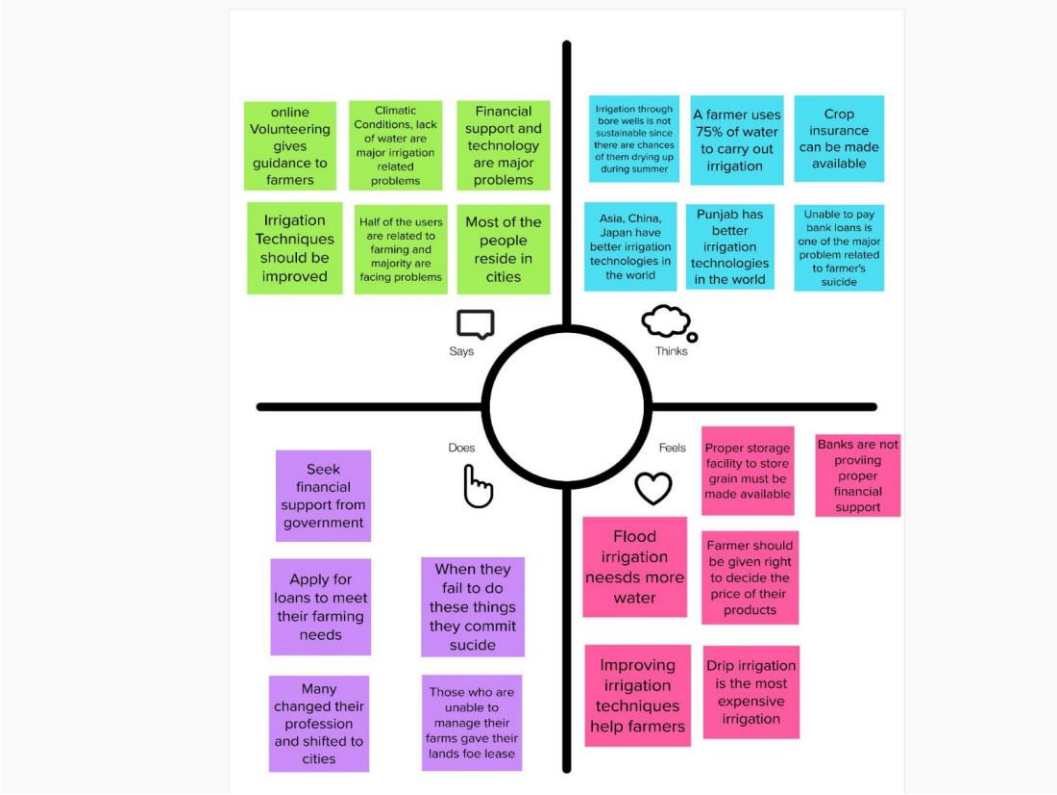
2.3 PROBLEM STATEMENT DEFINITION:

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:

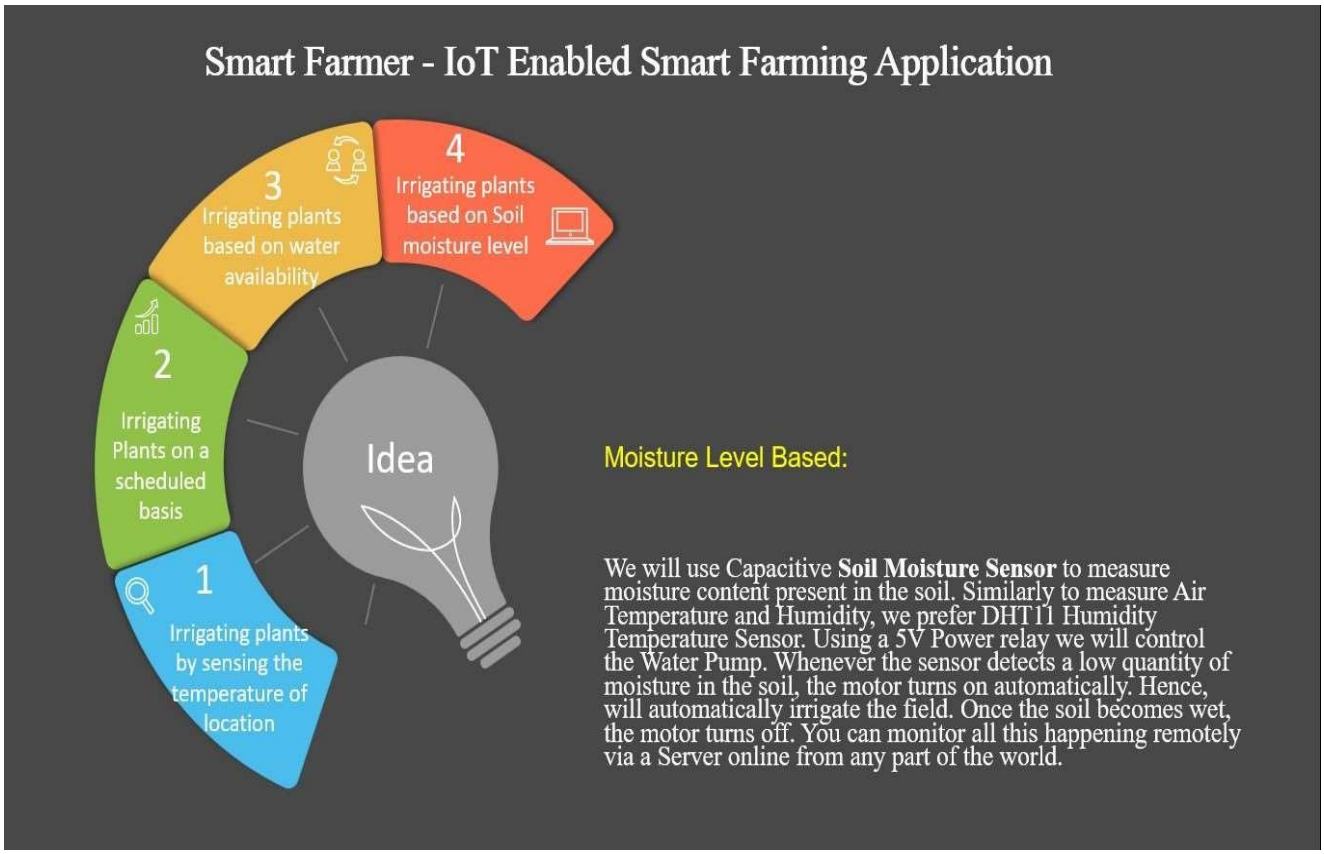


3.2 IDEATION & BRAINSTORMING:

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



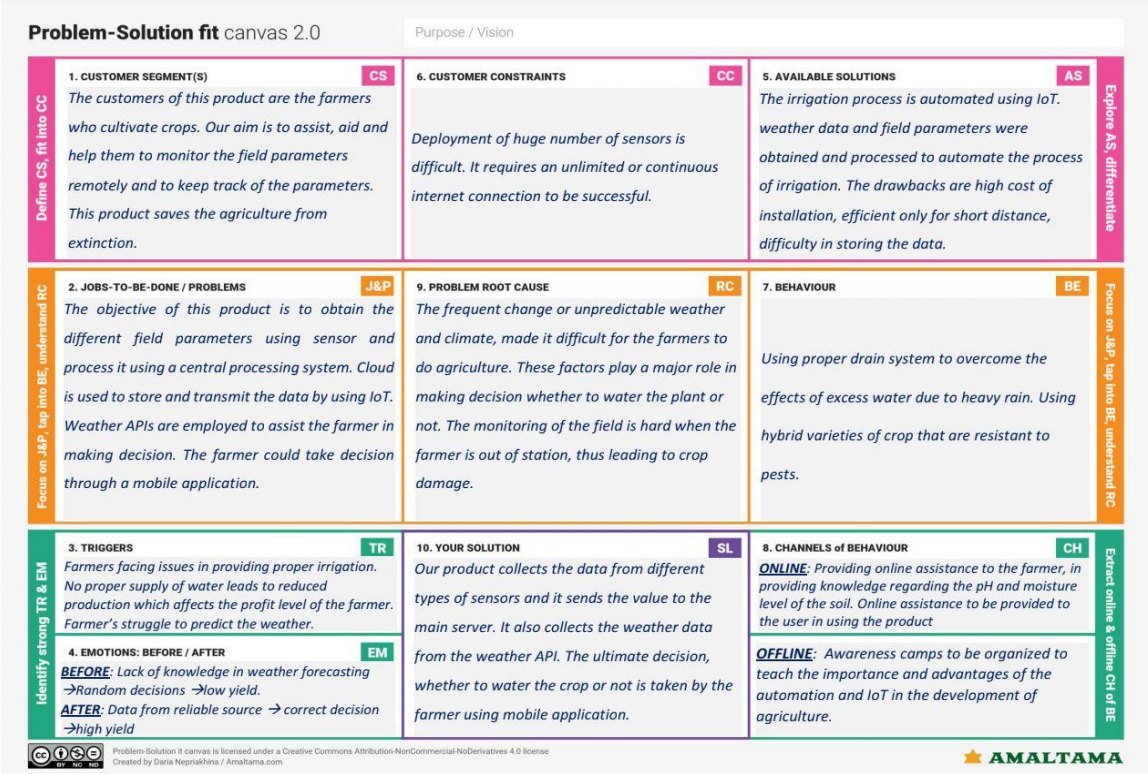
3.3 Proposed Solution Template:

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

Proposed Solution Template:

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	In agriculture, there are two major problems one is unpredictable climate change and another one is the yields of the crops that have been damaged by improper irrigation. Our project will give the solution to overcome these problems with help of IOT.
2.	Idea / Solution description	It collects the data from different types of sensors and it sends the value to the main server. It also collects the weather data from the weather API. The ultimate decision, whether to water the crop or not is taken by the farmer using mobile application.
3.	Novelty / Uniqueness	It depends on IOT thus eliminating the need of physical work of farmers and thus increasing the productivity in every possible manner. The weather data are taken from the reliable source.
4.	Social Impact / Customer Satisfaction	The informations collected are from reliable sources and hence the farmer could make more precise decision, thereby the productivity increases.
5.	Business Model (Revenue Model)	Smart farming is an advanced and innovative way to get maximum cultivation and minimize the human efforts.
6.	Scalability of the Solution	Automatic farming equipment adjustment is made feasible by integrating information such as crops/weather and equipment to automatically alter temperature, humidity, and so on. With the use of sensors, it has enabled farmers to reduce waste and increase output.

3.4 PROBLEM SOLUTIONS FIT :



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	IoT devices	Sensors and Wifi module.
FR-2	Software	Web UI, Node-red, IBM Watson, MIT app

4.2 NON FUNCTIONAL REQUIREMENTS:

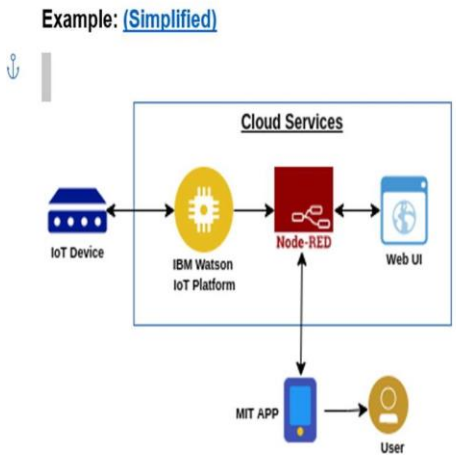
Following are the non-functional requirements of the proposed solution.

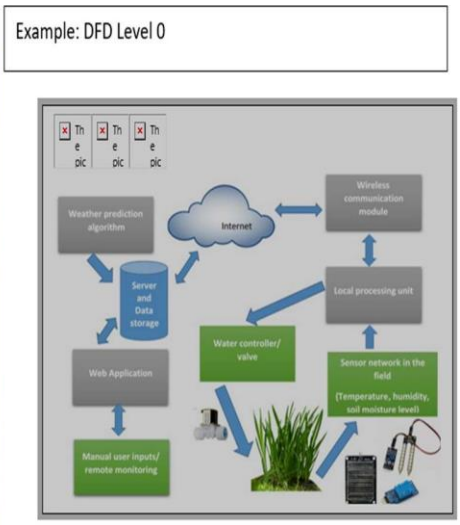
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Time consumability is less, Productivity is high.
NFR-2	**Security**	It has low level of security features due to integration of sensor data.
NFR-3	**Reliability**	Accuracy of data and hence it is Reliable.
NFR-4	**Performance**	Performance 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 DAIGRAMS 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

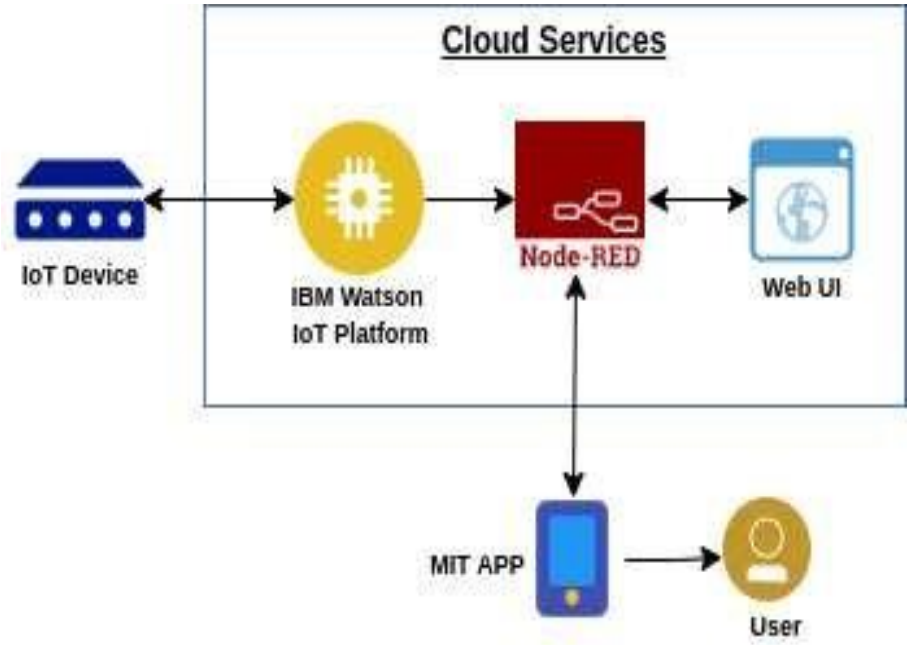


Table-1 : Components & Technologies:

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

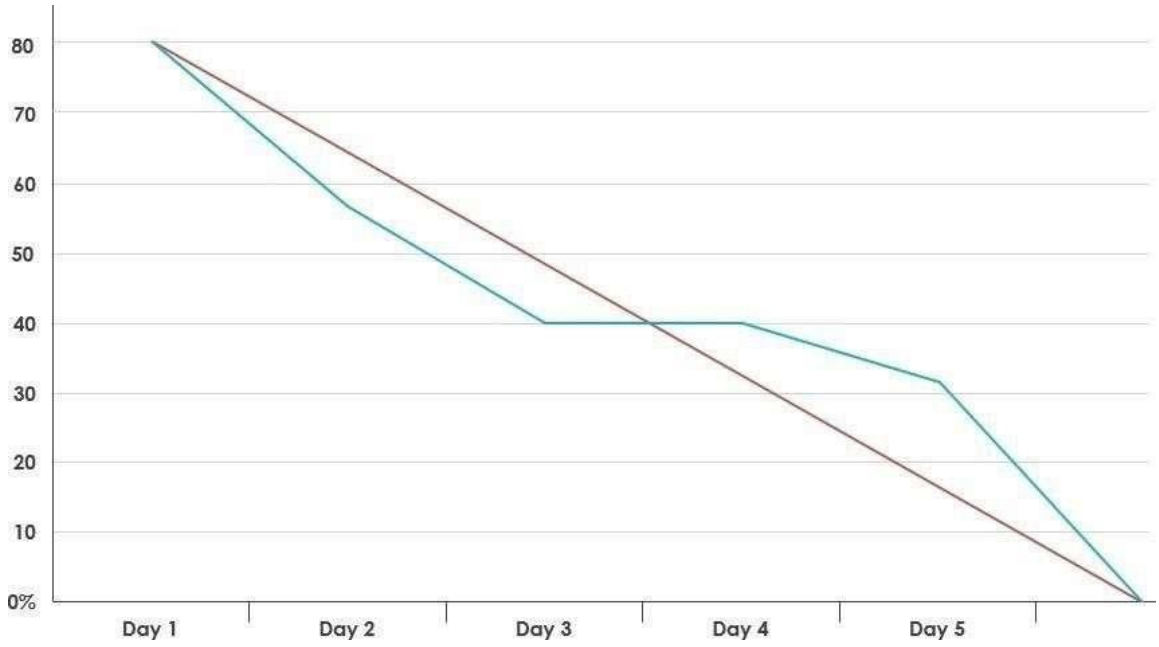
6. PROJECT PLANNING AND SCHEDULING:

Sprint	Functional Requirement (Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Member
Sprint-1	Registration (Farmer Mobile User)	UNS-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Gayathri.V
Sprint-1	Login	UNS-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Gayathri.V

Sprint-2	User Interface	UNS-3	As a user, I can register for the application through Facebook	3	Low	Sudharsana.S
Sprint-1	Data Visualization	UNS-4	As a user, I can register for the application through GMAIL	2	Medium	Santhiya.S
Sprint-3	Registration (Farmer -Web User)	USN - 1	As a user, I can log into the application by entering email and password	3	High	Hazirama.S
Sprint - 2	Login	USN - 2	As a registered user, I need to easily login log into my registered account via the web page in minimum time	3	High	Gayathri.V
Sprint - 4	Web UI	USN - 3	As a user, I need to have a friendly user interface to easily view and access the resources	3	Medium	Gayathri.V
Sprint - 1	Registration (Chemical Manufacturer - Web user)	USN - 1	As a new user, I want to first register using my organization email and create a password for the account.	2	High	Sudharsana.S

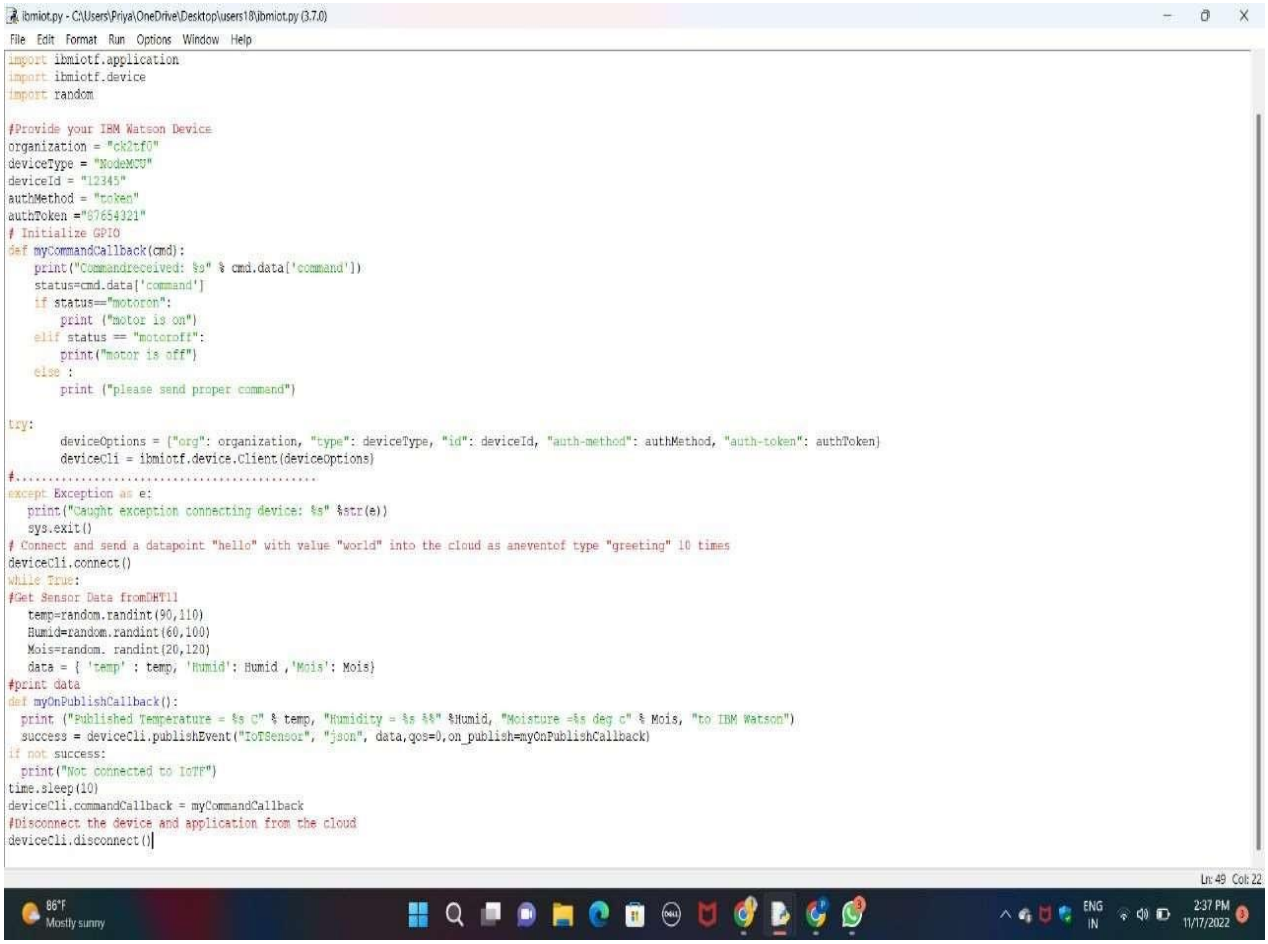
Sprint - 4	Login	USN - 2	As a registered user, I need to easily log in using the registered account via the web page.	3	High	Santhiya.S
Sprint - 3	Web UI	USN - 3	As a user, I need to have a user friendly interface to easily view and access the resources.	3	Medium	Hazirama.S
Sprint - 1	Registration (Chemical Manufacturer - Mobile User)	USN - 1	As a user, I want to first register using my email and create a password for the account.	1	High	Gayathri.V
Sprint - 1	Login	USN - 2	As a registered user, I need to easily log in to the application.	2	Low	Gayathri.V Sudharsana.S

Burndown Chart:



7. CODING & SOLUTIONS:

FEATURE :

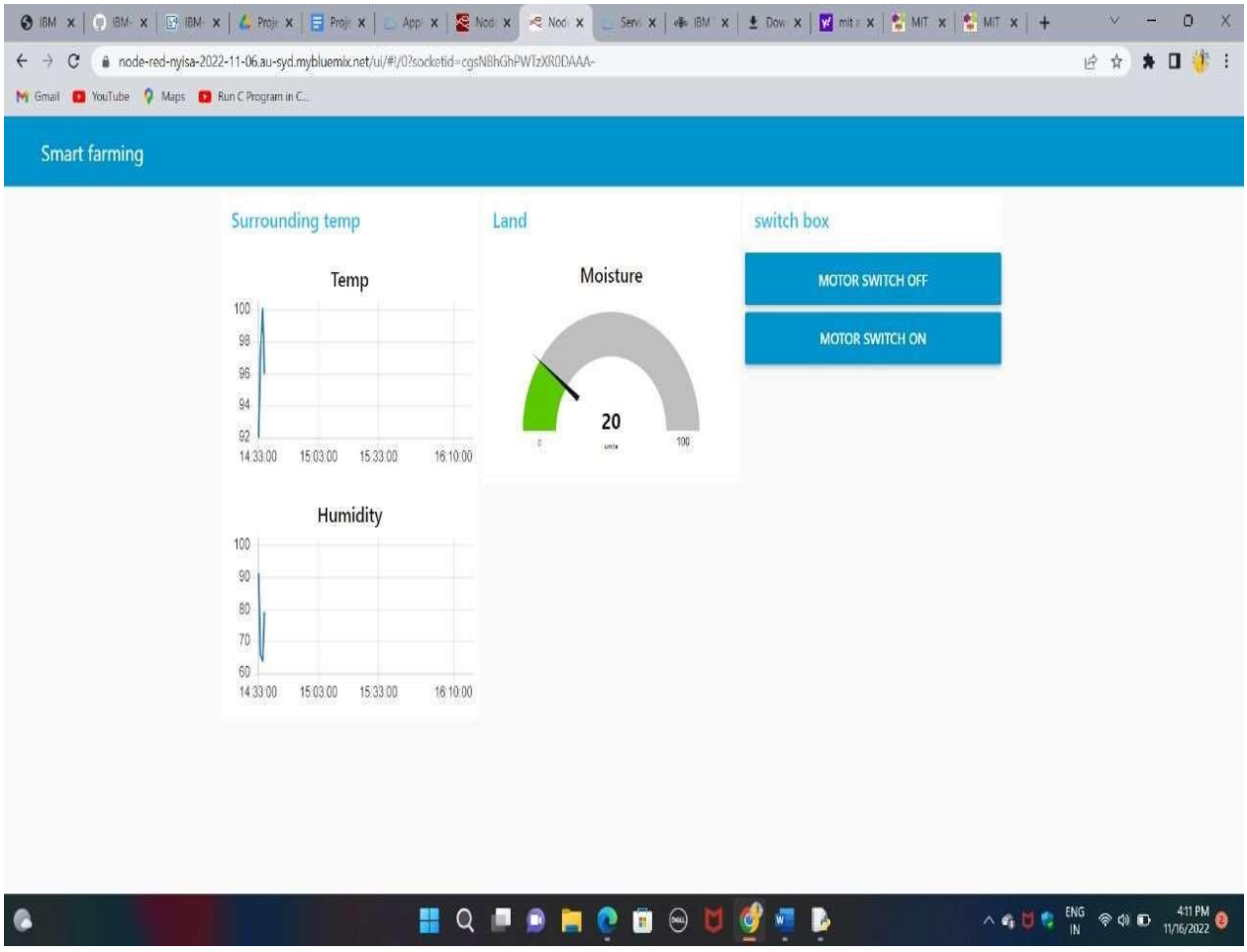


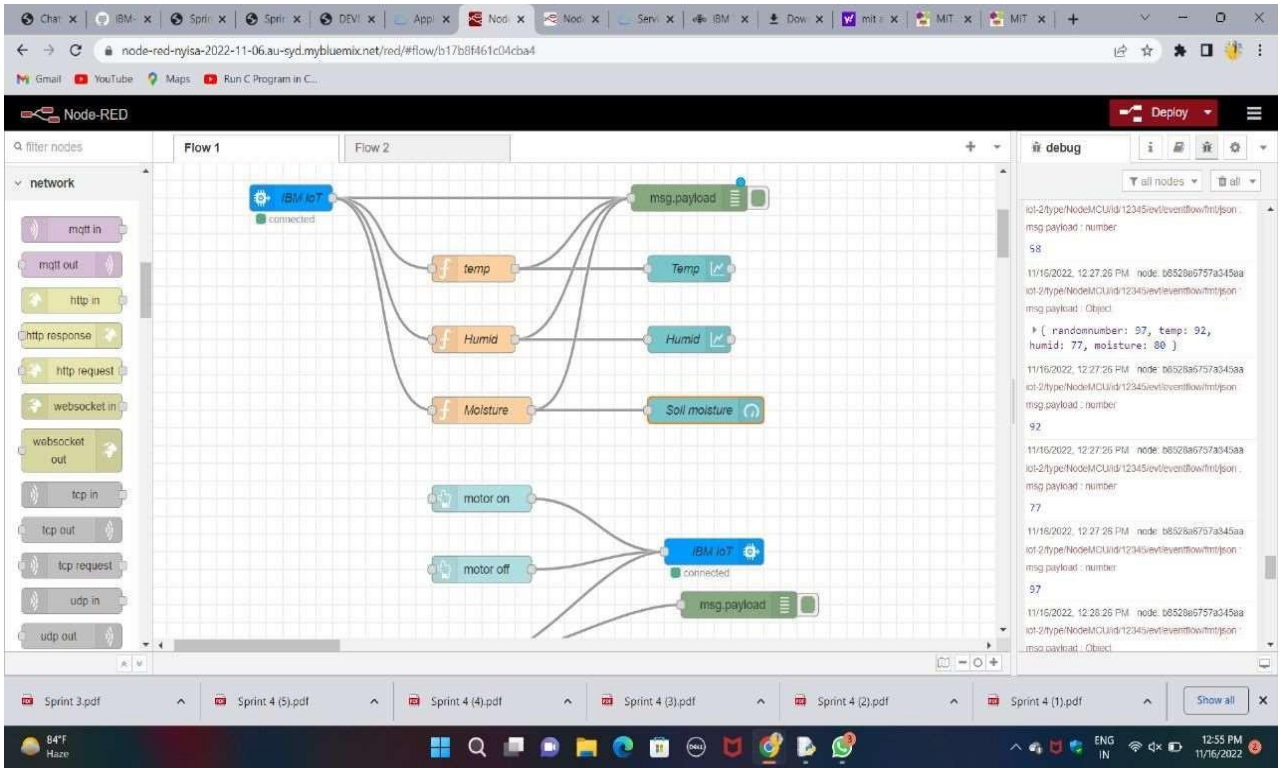
.

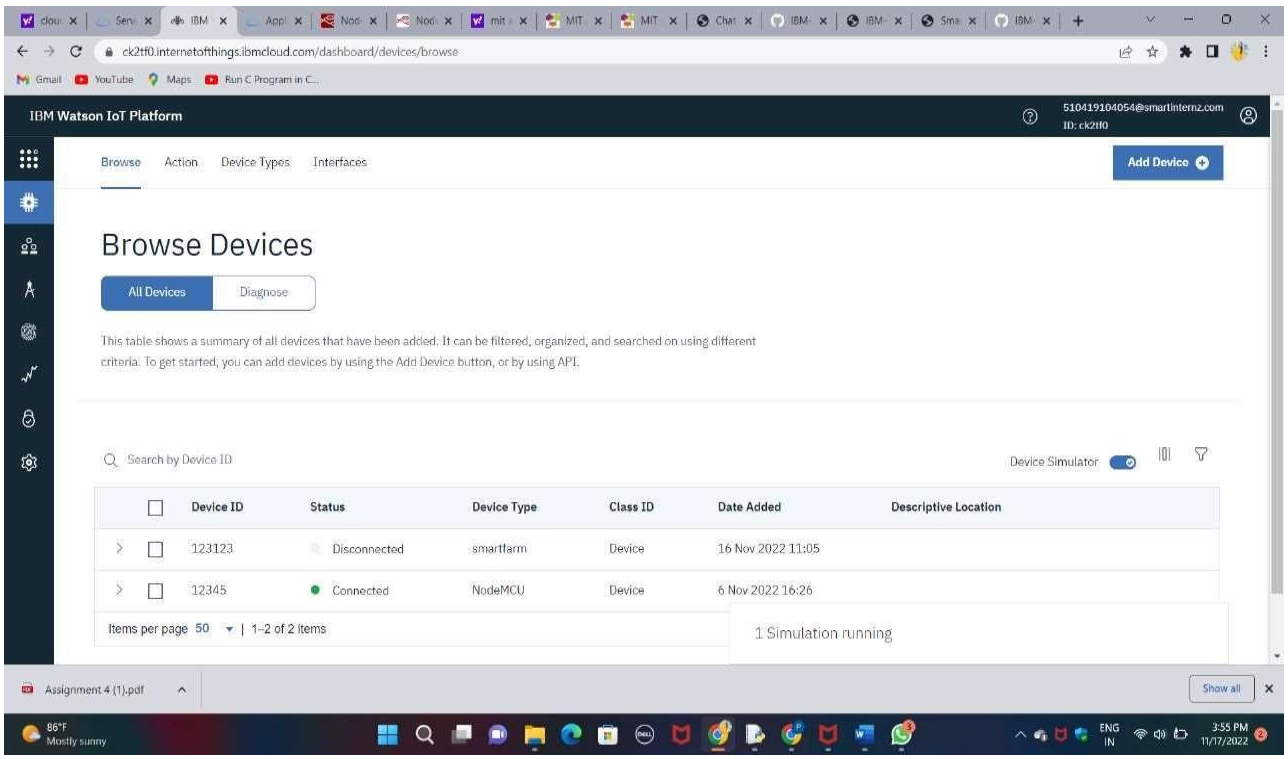
8. TESTING:

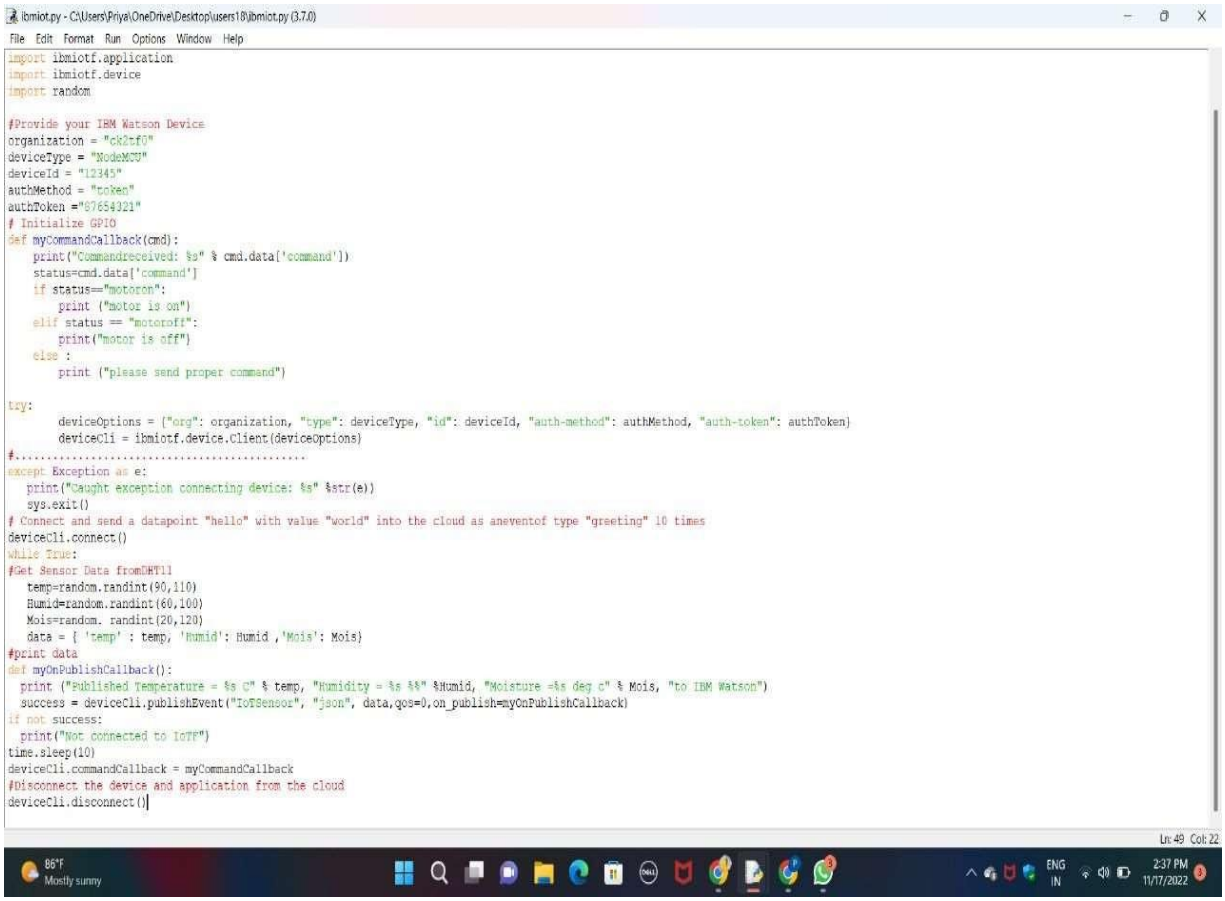
8.1 TEST CASE:

Web application using Node-RED.

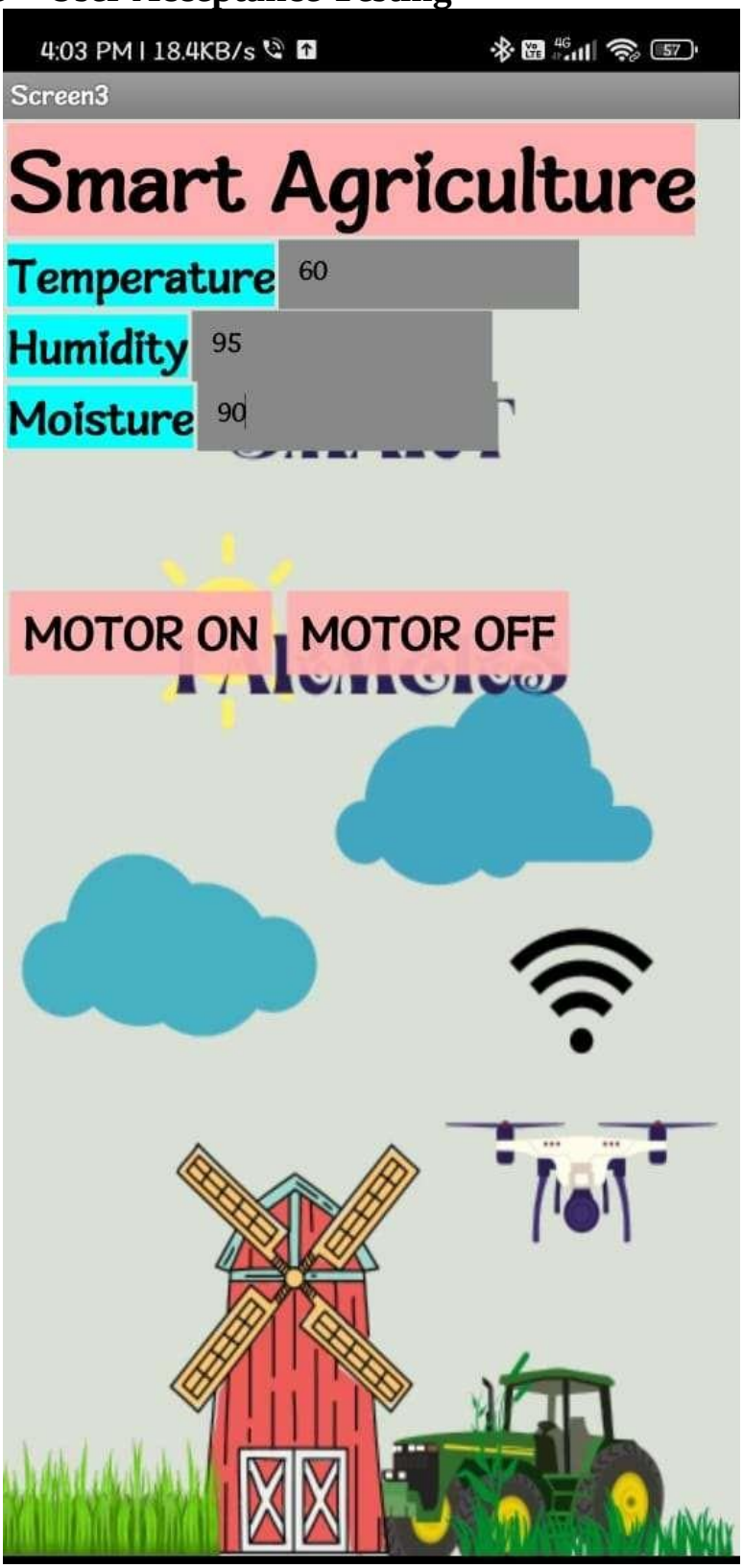



Node-RED
Deploy
Flow 1
Flow 2
network
mqtt in
mqtt out
http in
http response
http request
websocket in
websocket out
tcp in
tcp out
tcp request
udp in
udp out
IBM IoT
connected
msg.payload
temp
Temp
Humid
Humid
Moisture
Soil moisture
motor on
motor off
IBM IoT
connected
msg.payload
debug
{ randomnumber: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


IBM Watson IoT Platform
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
123123
Disconnected
smartfarm
Device
16 Nov 2022 11:05
12345
Connected
NodeMCU
Device
6 Nov 2022 16:26
Items per page 50
1–2 of 2 items
1 Simulation running

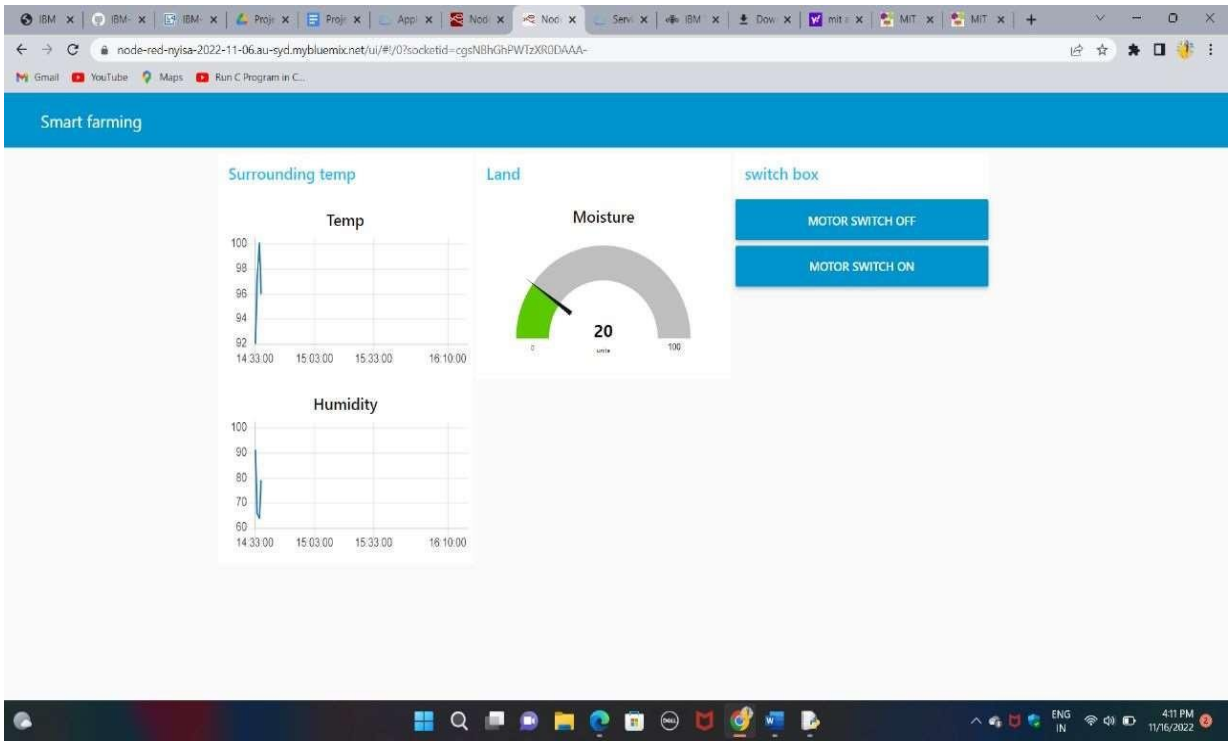

ibmiot.py - C:\Users\Priya\OneDrive\Desktop\users18\ibmiot.py (3.7.0)
File Edit Format Run Options Window Help
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device
organization = "ck2tf0"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken ="87654321"
# 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()
Ln: 49 Col: 22
86°F
Mostly sunny
ENG
IN
2:37 PM
11/17/2022

8.3 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 =  "ck2tf0"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken ="87654321"
# 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-3167-1658503853

THANK YOU…..