

PROJECT REPORT

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

Team ID: PNT2022TMID22889

Team:

Team leader:

SUBHASHINI.N(19ECR140)



Team members:

VIJAYALAKSHMI.S(19ECR152)

VIKRAM.M(19ECR153)

VASUDEVAN.D(19ECR147)

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 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 concerns , 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.

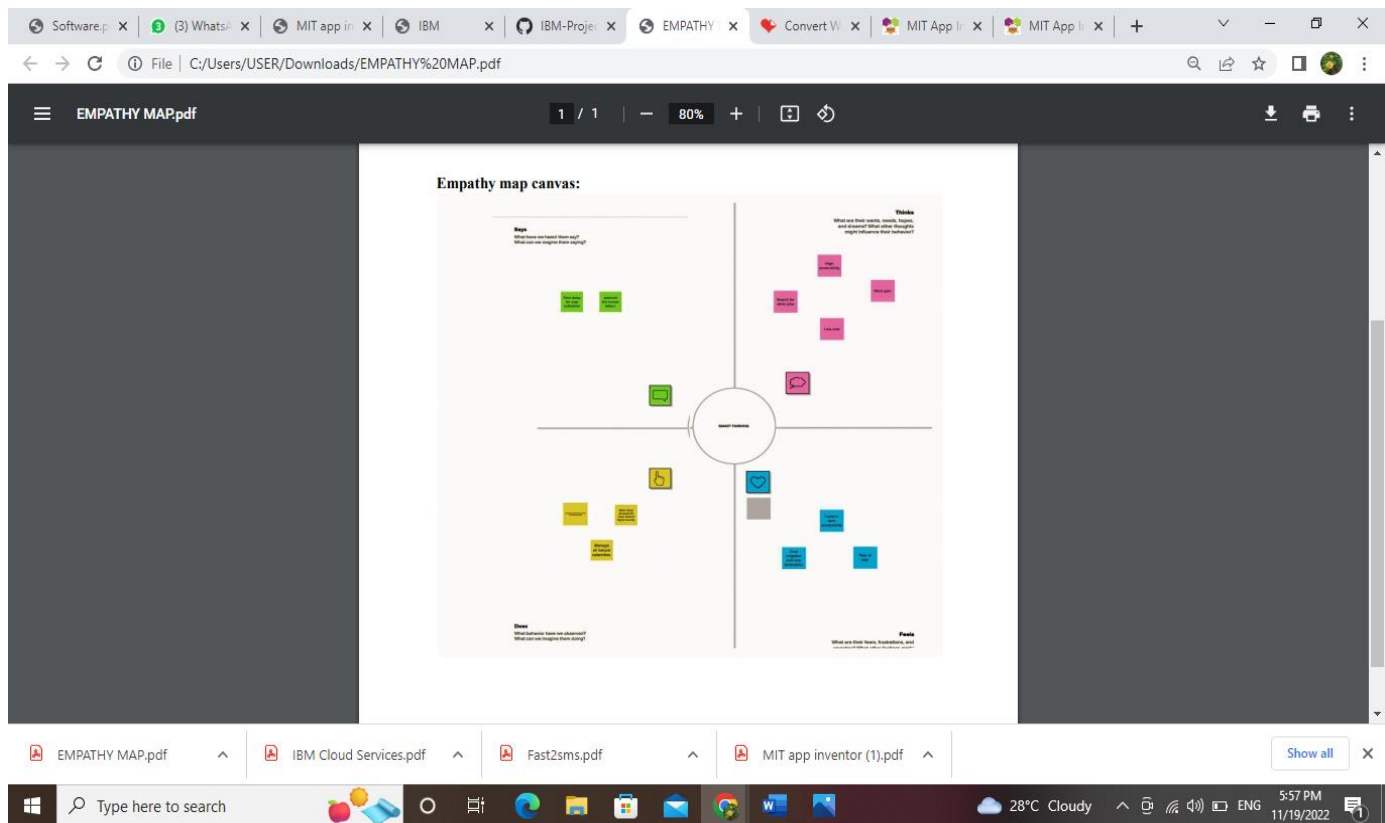
2.3 PROBLEM STATEMENT DEFINITION:

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 idea is understood as a basic element of thought that can be either visual, concrete, or abstract.

IDEATION

Step 1: Team Gathering, Collaboration and Select the Problem Statement



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

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 so invite. Share relevant information or pre-work ahead.
- B Set the goal**
Think about the problem you'll be focusing on solving 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

How might we [your problem statement]?

Key rules of brainstorming

To run a smooth and productive session

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

Step 2: Brainstorm, Idea Listing and Grouping

2 Brainstorm

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

10 minutes

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sticky-note-like label. If a cluster is larger than one sticky note, try and see if you can break it up into smaller sub-groups.

20 minutes

[Add your own sticky notes to this board](#)

Subhashini N

Smart farming using remote sensing and data analysis to optimize crop yield and reduce risk.	Smart Greenhouses	Remote sensing and data analysis to optimize crop yield and reduce risk.
Farming can help the environment thrive.	Solar powered and remote operated pumps save cost of electricity.	Smart farming delivers high quality crop production.

Vikram M

In smart farming, we use data to optimize crop yield and reduce risk.	Remote management	Increase production quantity and quality.
Crop protection	Farmers have to grow more product in deteriorating soil.	Real time analysis of soil demand.

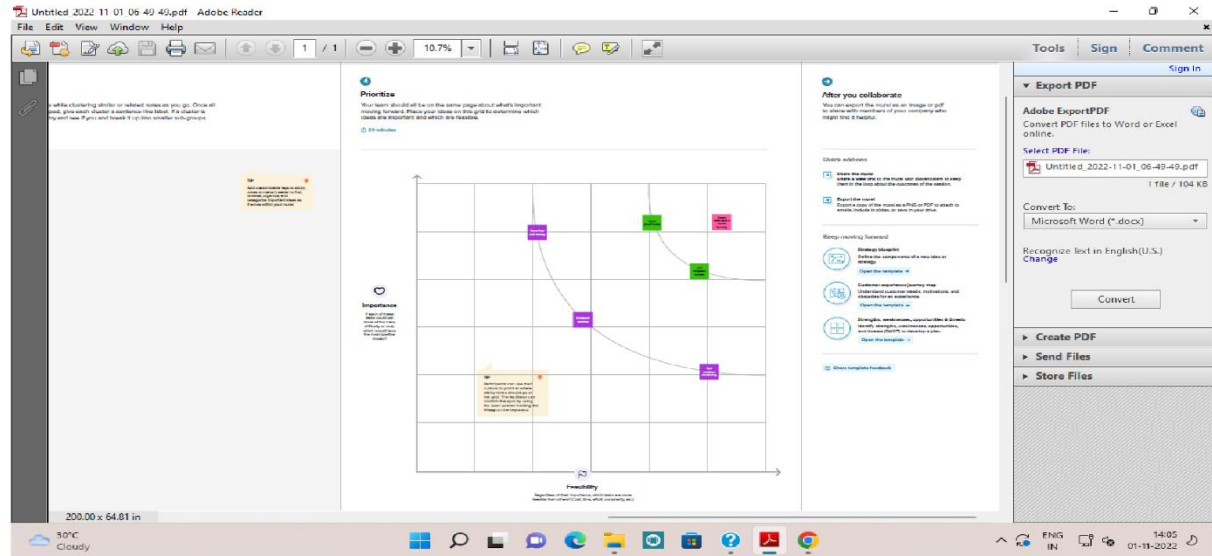
Vijaya lakshmi S

Smart farming using remote sensing and data analysis to optimize crop yield and reduce risk.	Smart Greenhouses	Remote sensing and data analysis to optimize crop yield and reduce risk.
IOT Irrigation control	Farmers can monitor their crops 24/7	The inputs of smart farming help you grow high quality crop production and delivery.

Vasudevan D

Smart farming using remote sensing and data analysis to optimize crop yield and reduce risk.	Smart Greenhouses	Remote sensing and data analysis to optimize crop yield and reduce risk.
Smart farming is cost effective method	Smart Greenhouses	Smart farming is cost effective method.

Step 3: Idea Prioritization



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.

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)	Farmers are under pressure to produce more food and use less energy and water in the process. A remote monitoring and control system will help farmers deal effectively with these pressures.
2.	Idea / Solution description	New technologies were used to monitor temperature, humidity and all other natural calamities.
3.	Novelty / Uniqueness	IoT sensors were used in measuring volumetric water content, soil oxygen level, soil water potential and to measure soil temperature.
4.	Social Impact / Customer Satisfaction	Smart farming reduces wastage of crops, increases the productivity of grains that leads to more gain and enable management of a greater number of resources through remote sensing.
5.	Business Model (Revenue Model)	As the productivity increases the customers satisfaction also increases and this may lead more need for many applications hence the revenue also increases.
6.	Scalability of the Solution	This is more scalable because of the adaptability of a system to increase the capacity.

3.4 PROBLEM SOLUTIONS FIT :

ProjectTitle:SmartFarmer-IoT Enabled Smart Farming Application

ProjectDesignPhase-I-SolutionFit

TeamID:PNT2022TMD22889

Define CS, fit into CC	<p>1. CUSTOMER SEGMENT(S) CS</p> <p>Who is your customer? i.e. working parents of 0-5 y.o. kids</p> <p>Farmers are the customers</p>	<p>6. CUSTOMER CONSTRAINTS CC</p> <p>What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.</p> <p>1.Inadequate crop protection 2.Minimal labor cost. 3.Limited nutrients availability.</p>	<p>5. AVAILABLE SOLUTIONS AS</p> <p>Which solutions are available to the customers when they face the problem? Or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking</p> <p>1.Promotes local food consumption. 2.Improves crop distribution. 3.Promotes more welfare for farmers.</p>	Explore AS, differentiate
Focus on J&P, fit into BE, understand RC	<p>2. JOBS-TO-BE-DONE / PROBLEMS J&P</p> <p>Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.</p> <p>1.Planting,cultivating. 2.Monitoring climatic conditions. 3.Supervising farm labor.</p>	<p>9. PROBLEM ROOT CAUSE RC</p> <p>What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations</p> <p>Smart farming helps in reducing overall Costs and improves the quality of crops.</p>	<p>7. BEHAVIOUR BE</p> <p>What does your customer do to address the problem or i.e. get the job done? Indirectly related: find the right solar panel installer, calculate usage and benefits, indirectly associated: customer's specific free time or volunteering work (i.e. Greenpeace).</p> <p>To monitor all agricultural activities such as land, soil moisture, atmospheric pressure and humidity.</p>	Focus on J&P, fit into BE, understand RC
	<p>3.TRIGGERS TR</p> <p>What triggers customer to act? i.e. seeing the neighbour installing solar panels, reading about more efficient solution in the news.</p> <p>1.Decrease in variety of crops. 2. Loss of agricultural land.</p>	<p>10.YOURSOLUTION SL</p> <p>If you are working on an existing business, write down your current solution first, fill in the canvas and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you outline the canvas and come up with a solution that fits within customer limitations, solve a problem and matches customer behaviour.</p>	<p>8. CHANNELS of BEHAVIOUR CH</p> <p>ONLINE What kind of actions do customers take online? Extract online channels from #7</p> <p>OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</p>	

4.REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL ANALYSIS:

Following are the functional requirements of the proposed solution.

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	EMAIL: Enter email address PASSWORD: Enter Password
FR-2	User Confirmation	Confirmation via Email. Thanks for email confirmation.
FR-3	Log in to system	Serve your authenticated content
FR-4	Manage Modules	Manage System Admins Manage Roles of Users Manage User permission
FR-5	Check weather conditions	Humidity monitoring status Temperature monitoring status
FR-6	Log out	Exit

4.2 NON FUNCTIONAL REQUIREMENTS:

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

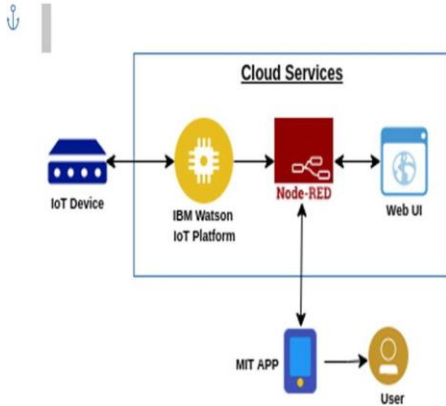
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability includes easy understanding and learning ability, efficiency in use, remember ability, lack of errors in operation and objective pleasure.
NFR-2	Security	Sensitive and private data must be protected from their production until the decision-making and storage stages.
NFR-3	Reliability	The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages.
NFR-4	Performance	The idea of implementing integrated sensors with sensing soil and environmental parameters in farming will be more efficient.

5. PROJECT DESIGN:

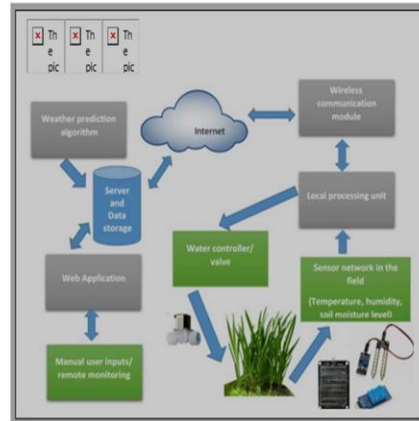
5.1 DATA FLOW DAIGRAMS AND USER STORIES:

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.

Example: (Simplified)



Example: DFD Level 0



5.2 SOLUTIONS AND TECHNICAL ARCHITECTURAL:

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

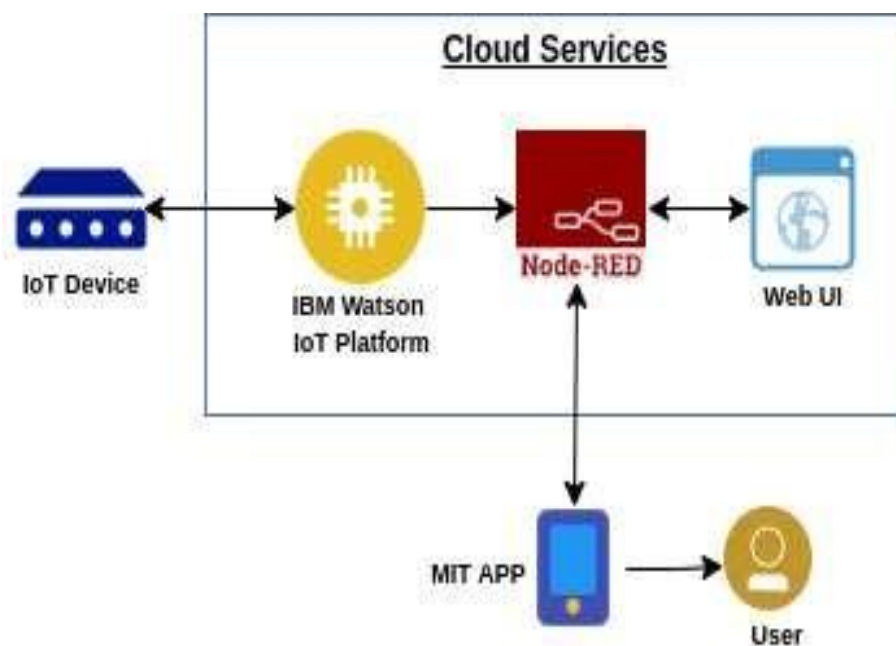


Table-1 : Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	The communication protocol being used might act as an interface	MIT App Inventor
2.	Arduino UNO	It is used as a processing Unit	Python
3.	MQTT protocol	The data to be collected and sent to the farmer via MQTT protocol providing the data to easily monitor the crops	IBM Watson IOT service, IBM Watson Assistant
4.	Cloud Database	Database Service on Cloud	IBM Cloud
5.	File Storage	Different soil parameters obtained values	IBM Block Storage
6.	External API	To monitor the weather	Open Weather API

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	MQTT protocol	Python
2.	Security Implementations	Sensitive and private data must be protected from their production until the decision-making and storage stages.	Node-Red, MIT App Inventor
3.	Scalable Architecture	Scalability is a major concern for IoT platforms. It has been shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousand.	Node-Red service
4.	Availability	Available feasible	Open weather App
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	MIT app inventor

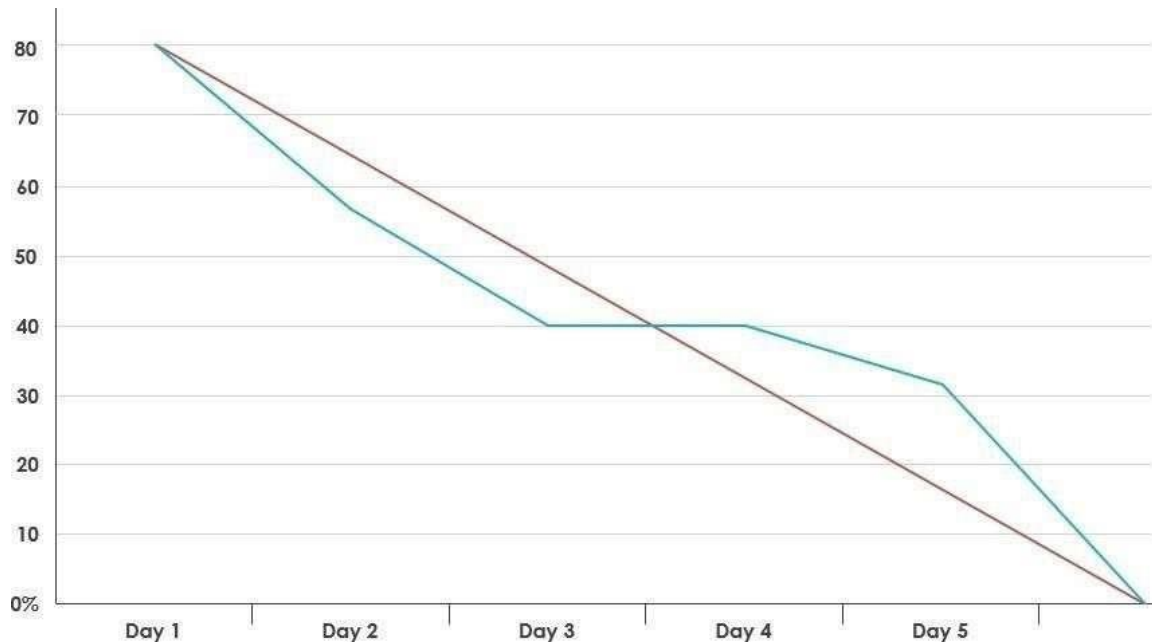
6.PROJECT PLANNING AND SCHEDULING

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	N.Subhashini(Leader)
Sprint-1	Login	UNS-2	As a user, I will receive confirmation email once I have registered for the application	1	High	S.Vijayalakshmi(Member 1)

Sprint-2	User Interface	UNS-3	As a user, I can register for the application through Facebook	3	Low	M.Vikram (Member 2)
Sprint-1	Data Visualization	UNS-4	As a user, I can register for the application through GMAIL	2	Medium	D.Vasudevan (Member 3)
Sprint-3	Registration (Farmer -Web User)	USN - 1	As a user, I can log into the application by entering email and password	3	High	N.Subhashini(Leader)
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	S.Vijayalaksh mi (Member 1)
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	M.Vikram (Member 2)

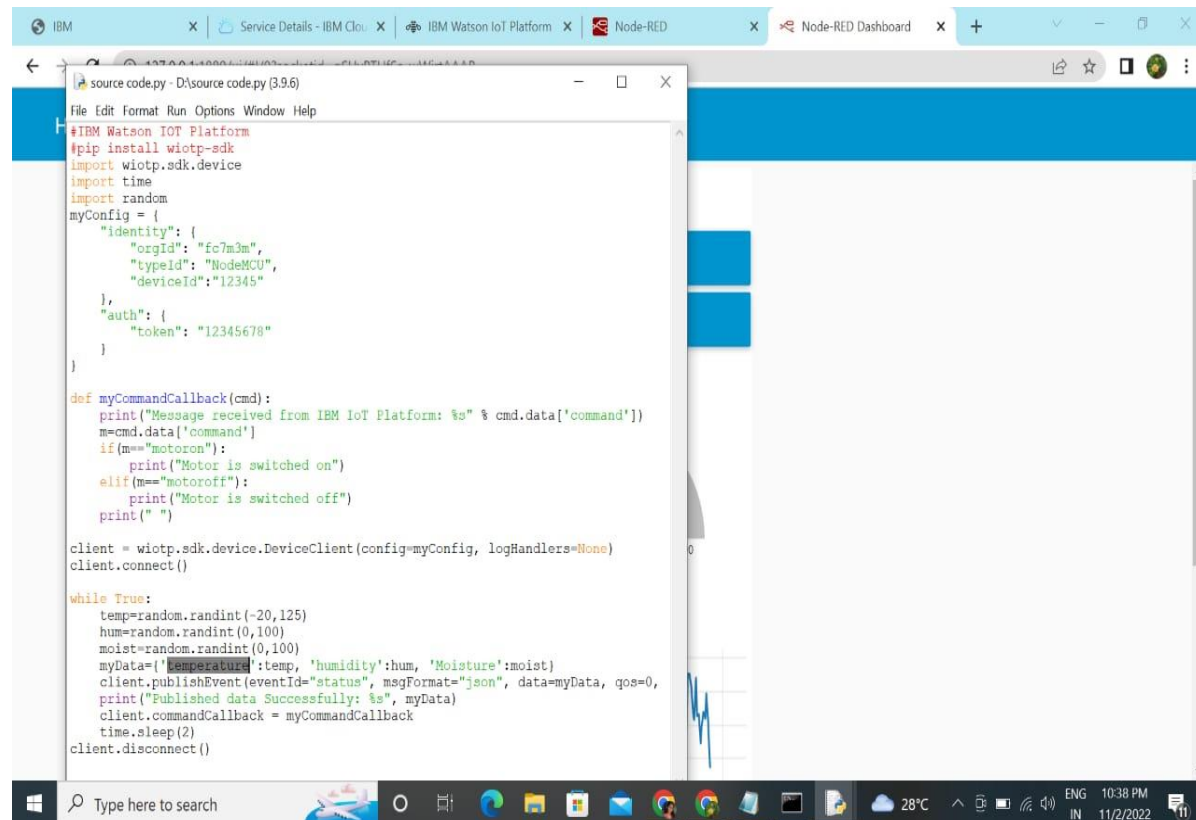
Sprint - 1	Registration (Chemical Manufacturer - Web user)	USN - 1	As a new user, I want to first register using my organization email and create a password for the account.	2	High	D.Vasudevan (Member 3)
Sprint - 4	Login	USN - 2	As a registered user, I need to easily log in using the registered account via the web page.	3	High	N.Subhashini (Leader)
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	S.Vijayalaksh mi (Member 1)
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	M.Vikram (Member 2)
Sprint - 1	Login	USN - 2	As a registered user, I need to easily log in to the application.	2	Low	D.Vasudevan (Member 3)

Burndown Chart:



7.CODING & SOLUTIONS:

FEATURE :



The screenshot displays a Windows desktop environment. In the foreground, a code editor window titled 'source code.py - D:\source code.py (3.9.6)' is open, showing Python code for interacting with the IBM Watson IoT Platform. The code includes imports for the SDK, time, and random modules, a configuration dictionary for device identity and authentication, a callback function for handling commands, and a loop for publishing status data. In the background, a Node-RED dashboard is visible, featuring a blue header and a graph visualization. The Windows taskbar at the bottom shows the search bar, task view button, and various application icons, along with system status information like temperature (28°C) and time (10:38 PM, 11/2/2022).

```
#IBM Watson IoT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "fc7m3m",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "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 switched on")
    elif(m=="motoroff"):
        print("Motor is switched off")
    print(" ")

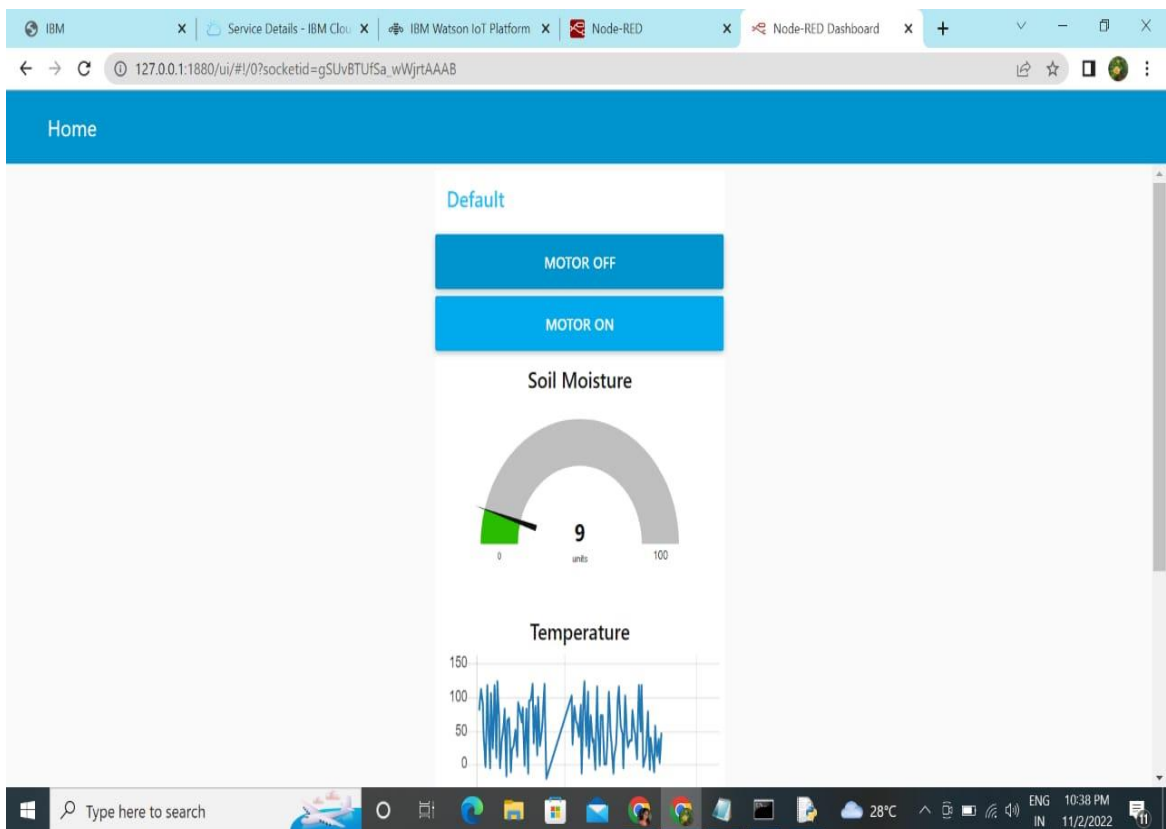
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

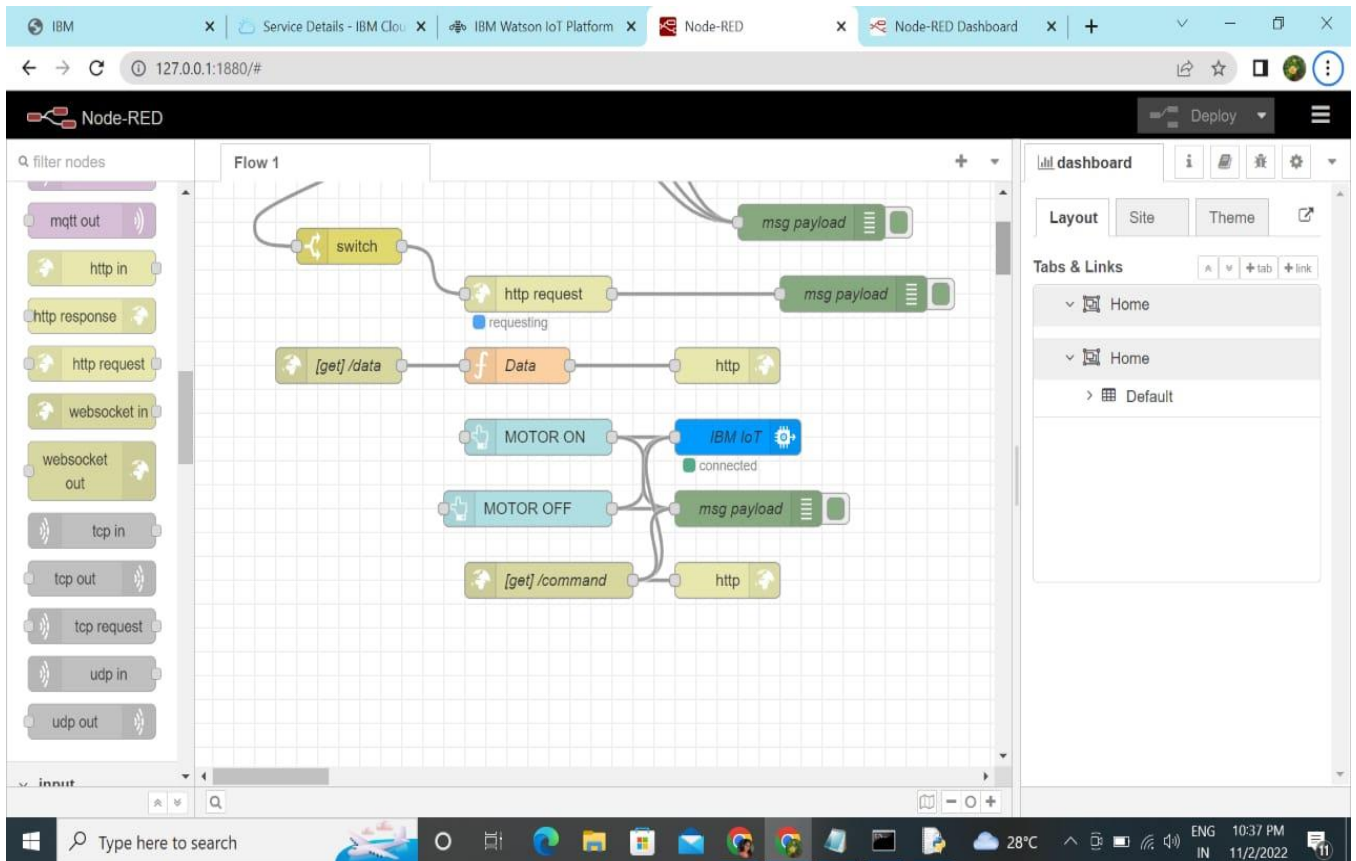
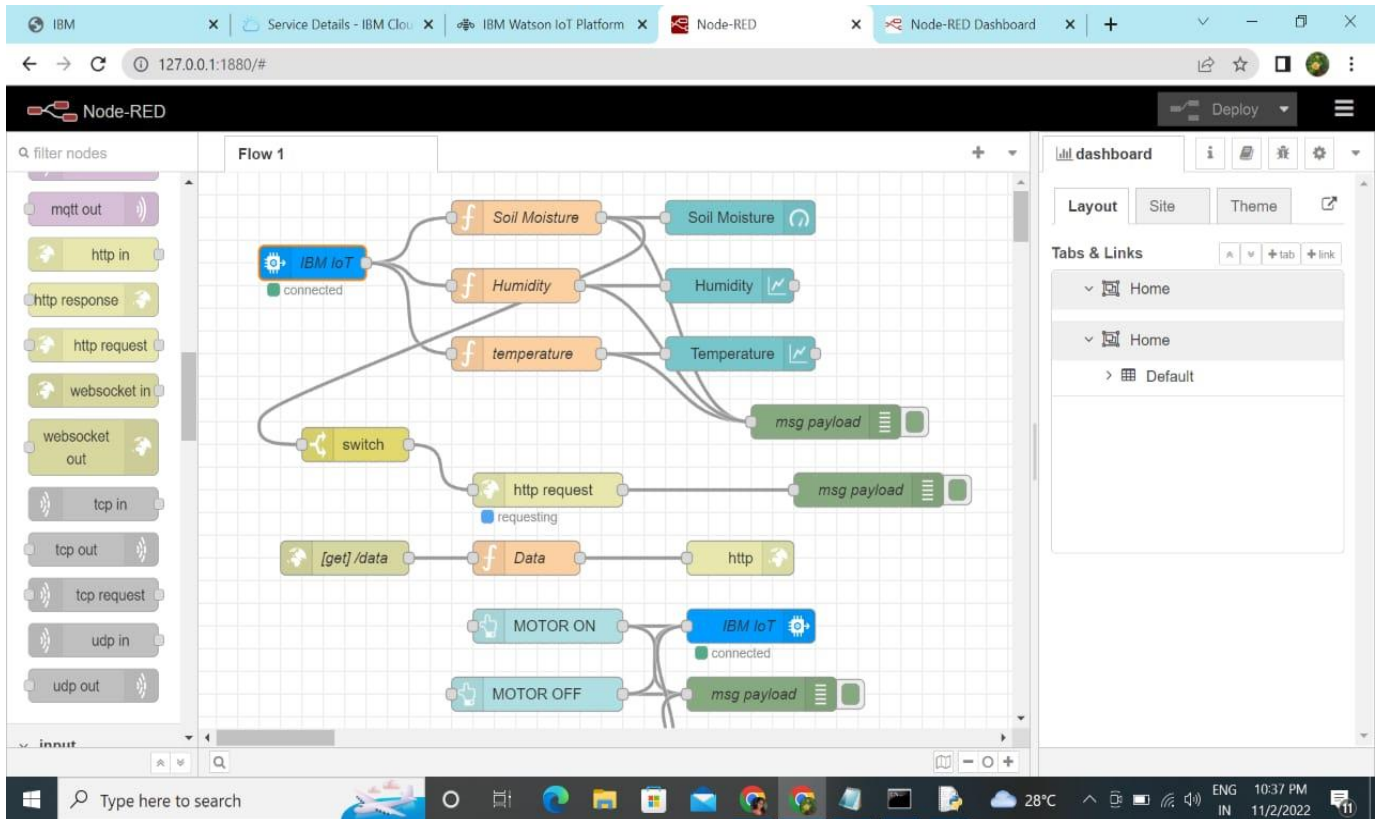
while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    moist=random.randint(0,100)
    myData={"Temperature":temp, 'humidity':hum, 'Moisture':moist}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
    client.disconnect()
```


8. TESTING:

8.1 TEST CASE:

Web application using Node-RED.





source code.py - D:\source code.py (3.9.6)

```
File Edit Format Run Options Window Help
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "fc7m3m",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

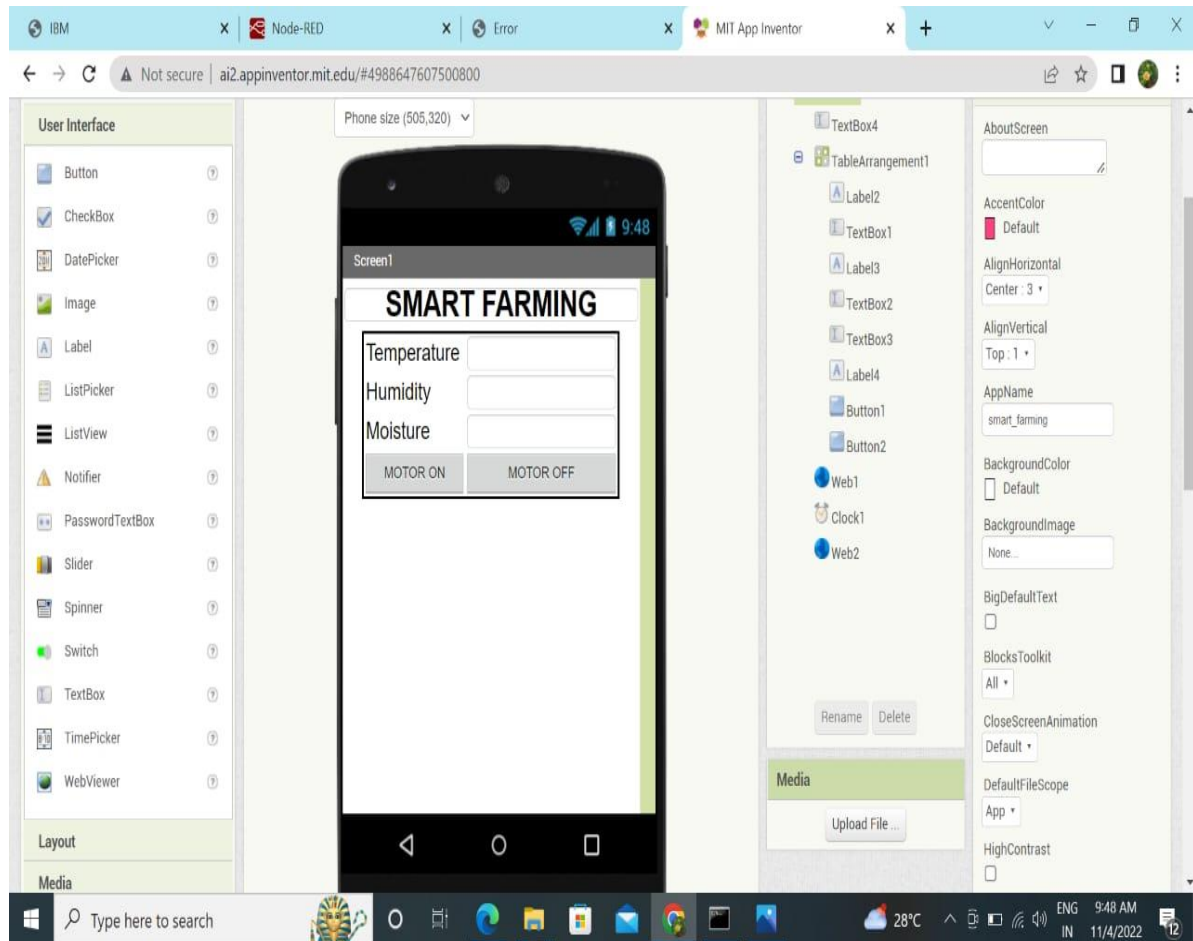
def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="motoron"):
        print("Motor is switched on")
    elif(m=="motoroff"):
        print("Motor is switched off")
    print(" ")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    moist=random.randint(0,100)
    myData={'Temperature':temp, 'humidity':hum, 'Moisture':moist}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

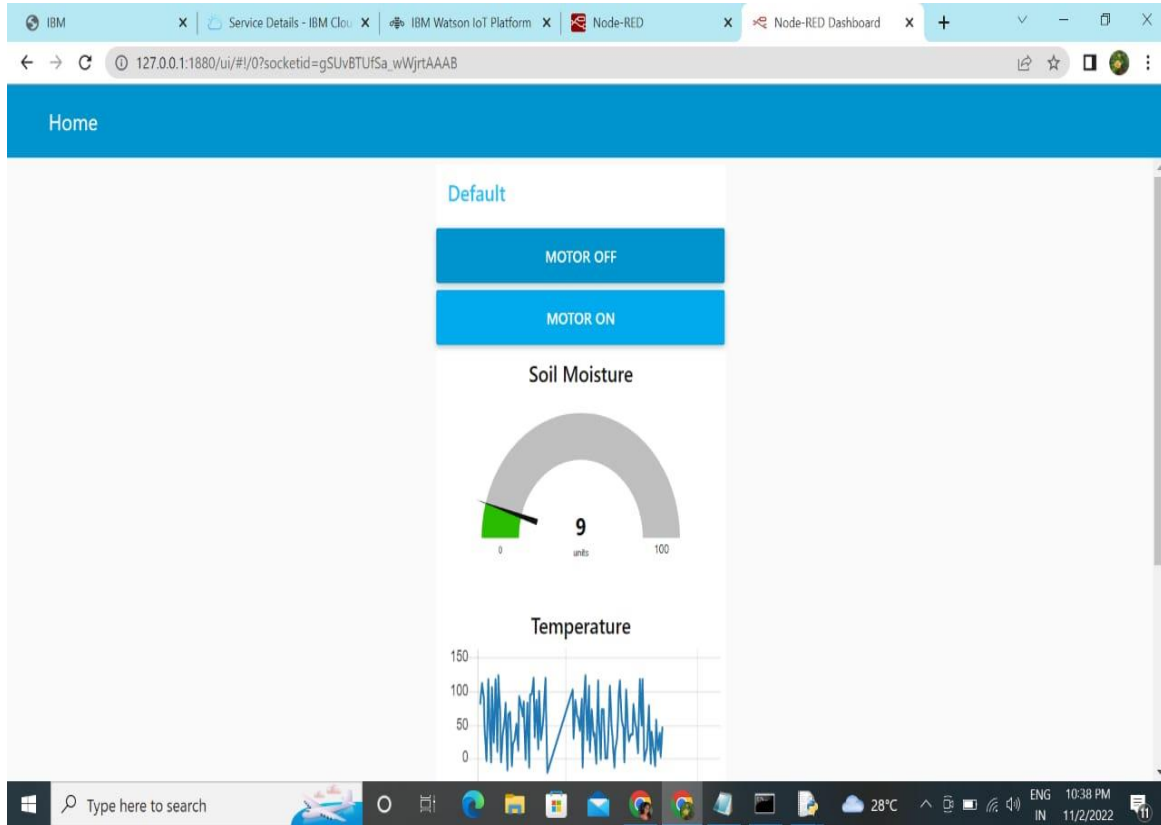
28°C 10:38 PM 11/2/2022

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:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "3q2j4g",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "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 switched on")
elif(m=="motoroff"):
    print("Motor is switched off")
print(" ")
```

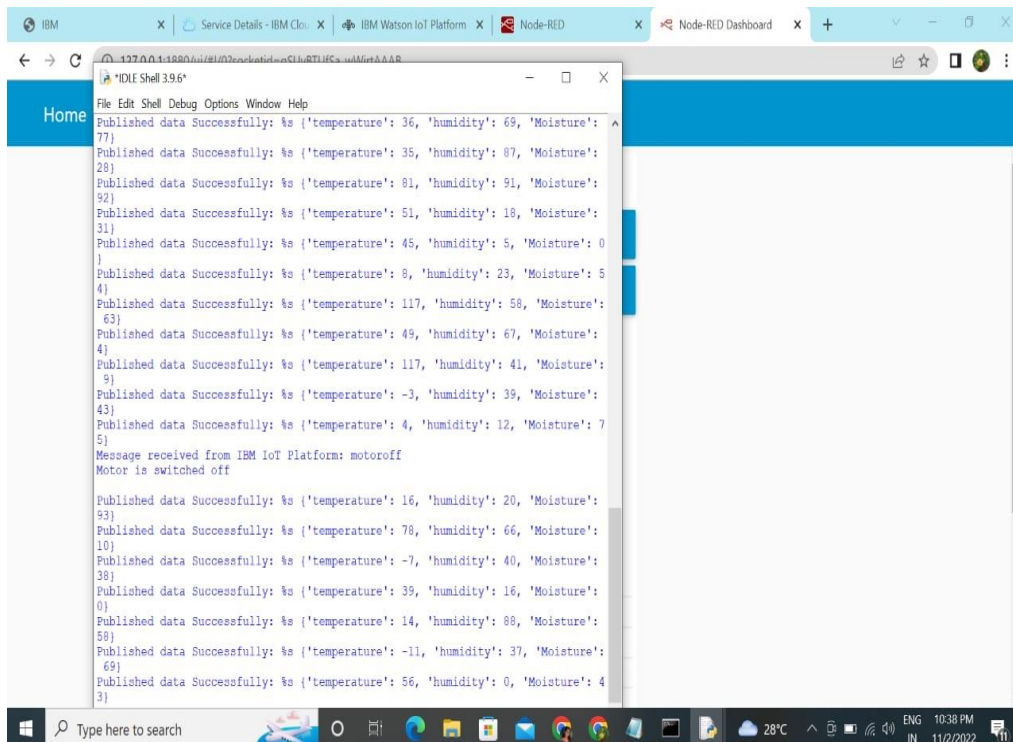
```

client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    ph=random.randint(0,14)
    myData={'temperature':temp, 'humidity':hum, 'PH':ph}
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()

```

OUTPUT



```

Home
Published data Successfully: %s {'temperature': 36, 'humidity': 69, 'Moisture': 77}
Published data Successfully: %s {'temperature': 35, 'humidity': 87, 'Moisture': 28}
Published data Successfully: %s {'temperature': 81, 'humidity': 91, 'Moisture': 92}
Published data Successfully: %s {'temperature': 51, 'humidity': 18, 'Moisture': 31}
Published data Successfully: %s {'temperature': 45, 'humidity': 5, 'Moisture': 0}
Published data Successfully: %s {'temperature': 8, 'humidity': 23, 'Moisture': 54}
Published data Successfully: %s {'temperature': 117, 'humidity': 58, 'Moisture': 63}
Published data Successfully: %s {'temperature': 49, 'humidity': 67, 'Moisture': 4}
Published data Successfully: %s {'temperature': 117, 'humidity': 41, 'Moisture': 9}
Published data Successfully: %s {'temperature': -3, 'humidity': 39, 'Moisture': 43}
Published data Successfully: %s {'temperature': 4, 'humidity': 12, 'Moisture': 75}
Message received from IBM IoT Platform: motoroff
Motor is switched off
Published data Successfully: %s {'temperature': 16, 'humidity': 20, 'Moisture': 93}
Published data Successfully: %s {'temperature': 78, 'humidity': 66, 'Moisture': 10}
Published data Successfully: %s {'temperature': -7, 'humidity': 40, 'Moisture': 38}
Published data Successfully: %s {'temperature': 39, 'humidity': 16, 'Moisture': 0}
Published data Successfully: %s {'temperature': 14, 'humidity': 88, 'Moisture': 58}
Published data Successfully: %s {'temperature': -11, 'humidity': 37, 'Moisture': 69}
Published data Successfully: %s {'temperature': 56, 'humidity': 0, 'Moisture': 43}

```


Github link : [https://github.com/IBM-EPBL /IBM-Project-35279-1660283138](https://github.com/IBM-EPBL/IBM-Project-35279-1660283138)

Project Demo link :

<https://drive.google.com/file/d/1KT3G7RQl8IKrdO5NliJbEw3XwlbVYmt-/view?usp=drivesdk>

THANK YOU....