Mepco Schlenk Engineering College

SMART FARMER – IOT ENABLED SMART FARMING APPLICATION

IBM NALAIYATHIRAN

Project Report

TITLE	Smart Farmer IoT Enabled Smart Farming Application
DOMAIN NAME	INTERNET OF THINGS
TEAM ID	PNT2022TMID18437
TEAM LEADERNAME	A.Prabhu
TEAM MEMBER NAME	K.Rahul khan S.J.Sri Aakash S.Sarveshwaran
MENTOR NAME	Mr.S.Sridhar Raj

1.INTRODUCTION

1.1 Project Overview

Agriculture plays a important role in country's economy and provides a large scale employment to the people. However, agriculture is highly dependent upon weather and climate. For example, changes in temperature, soil moisture, carbon dioxide may result in low yield of crops. It is Significant to monitor environmental parameters in order to manage crop growth and increase the agricultural production yield. The sensed information is not only important for decision making but also for evaluating impacts of agricultural practices on environment. Nowadays, it is more necessary than ever to increase the crop yields food grain production. Cloud connected, wireless system aid in this crop yield maximization, which automates day-to-day agricultural tasks and real time monitoring for smart decision-making.

1.2 Purpose

- Need for technology to monitor important parameters like soil moisture, temperature, Humidity etc. to improve the cultivation process.
- Development of certain techniques to reduce the workforce, energy and time for cultivation.
- Development of a feasible method to control the electrical equipment in the farm from any part of the world.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. Controlling the device from longer distance from web application.
2. Transfer of node data to the gateway at faster rate.
3. Unavailability of data's such as PH level, potassium, Nitrogen etc related to the soil.

2.2 PROPOSED SOLUTION

1. To control a device from longer distance from web application.
2. To display the data in the web application.

2.3 References

1) **Water Management in Agriculture: A Survey on Current Challenges and Technological Solutions**

Authors: ABDELMADJID SAAD , ABOU EL HASSAN BENYAMINA , and ABDOULAYE GAMATIE
Published: IEEE access
Description:

This paper aims at optimizing water usage, and improving the quality and quantity of agricultural crops and minimizing the need for direct human intervention which is achieved by smoothing the water monitoring process, by applying the right automation level, and allowing farmers getting connected anywhere and anytime to their farms

2) **Smart Agriculture Wireless Sensor Routing Protocol and Node Location Algorithm Based on Internet of Things Technology**

Authors: DINGZHU XUE and WEI HUANG

Published: IEEE Sensors Journal

Description:

This paper aims at the problem, of low positioning accuracy and large error of DV-HOP algorithm, which is solved by an improved method of DV-HOP algorithm based on average HOP distance to make positioning more precise.

3) **Printed Sensor Technologies for Monitoring Applications in Smart Farming: A Review**

Authors: ZHENG LIU and RAKIBA RAYHANA

Published: IEEE Transactions on Instrumentation and Measurement

Description:

This paper highlights the enhanced performance of the printed sensors along with the wireless and IoT technologies, and the application domains in different areas that can be expanded to aid the smart farming practice

4. A Systematic Review on Monitoring and Advanced Control Strategies in Smart Agriculture

Authors: SYEDA IQRA HASSAN, MUHAMMAD MANSOOR ALAM, USMAN ILLAHI, MOHAMMED A. AL GHAMDI
Published: IEEE Access
Description:

This paper helps to cope up with the pressure due to the changes in climate, erosion of soil, biodiversity loss and from end users.

5. Smart Irrigation System for Precision Agriculture—The ARETh0U5A IoT Platform

Authors: SPYRIDON NIKOLAIDIS, ACHILLES D. BOURSIANIS and MARIA S. PAPADOPOULOU
Published: IEEE Sensors Journal
Description:

This paper highlights the architecture of an intelligent irrigation system for precision agriculture by ARETh0U5A IoT platform that is developed to perform intelligent irrigation practices and policies in water irrigation management of a perennial olive field.

2.4 Problem Statement Definition

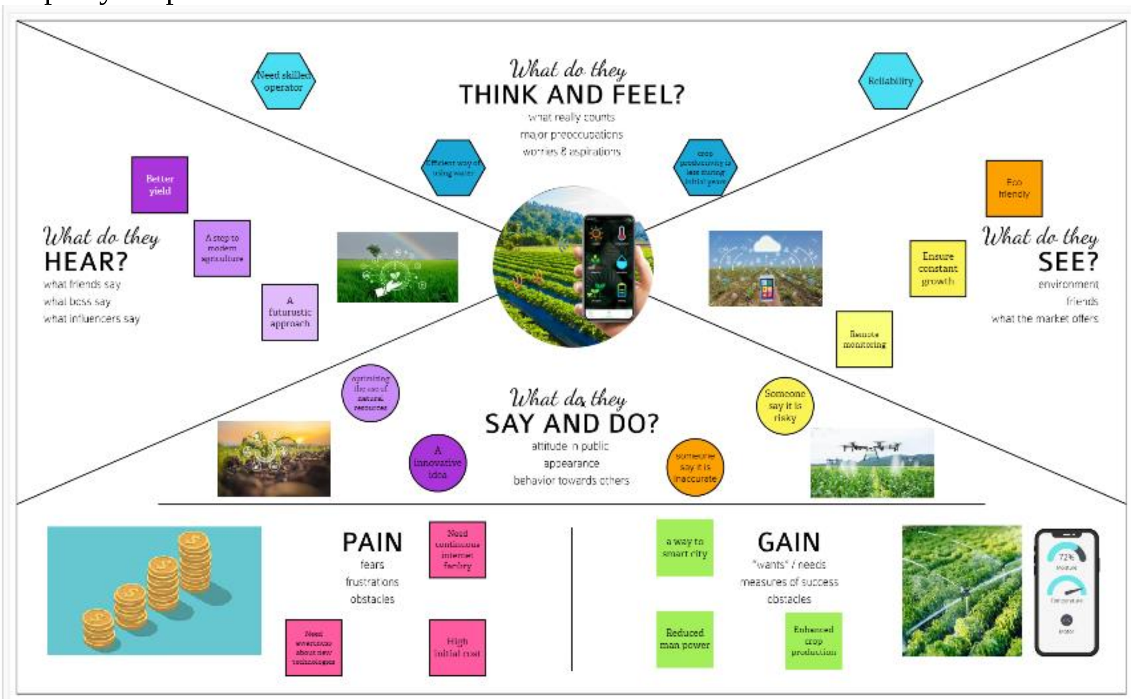
I am	Commercial Farmer	Farmer who produces crops, fruits, vegetables, cereals and pulses in large scale
I'm trying to	Increase Yield	Periodic monitoring and timely actions will increase the yield of produce and reduces chances of crop failure
But	Fail to take actions	Proper actions couldn't be taken due to lack of a comprehensive monitoring of each and every region of the field
Because	Difficulty in monitoring	Large scale of the farm makes it difficult to monitor the entire farm and take timely actions
Which makes me feel	Demoralized	Profit is what sought after in a business so missing the opportunity to make more money demoralizes a business man

I am	Subsistence Farmer	Farmer who grows crops only for their own use, without any surplus for trade
I'm trying to	Irrigate efficiently	Due to lack of abilities to know the soil moisture of the field, farmer tend to over irrigate or under irrigate the field
But	Inundate the fields	To prevent dryness, they tend to over irrigate the field and also require to manually on the motor
Because	Lack of water flow control	Lack of motor control makes the farmer unable to operate motor based on soil moisture and weather conditions
Which makes me feel	Helpless	Subsistence farmers couldn't afford to employ people to monitor the farm continuously so are helpless

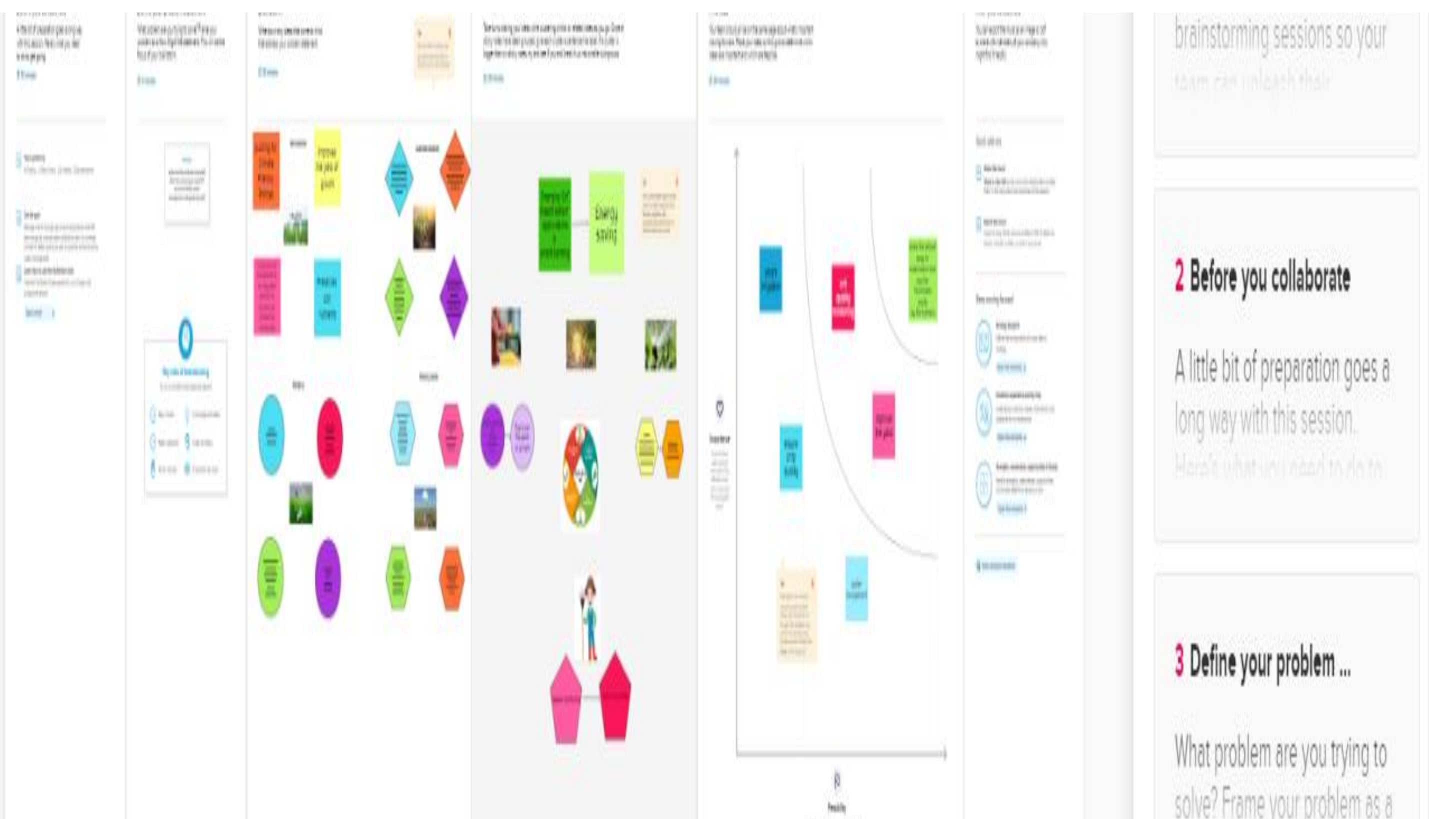
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Commercial farmer	Increase the yield	Fail to take actions	Difficulty in monitoring	Demoralized
PS-2	Subsistence farmer	Irrigate efficiently	Inundate the fields	Lack of water flow control	Helpless

3 IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming



3.3Proposed solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a app based solution for soil health monitoring and efficient water management
2.	Idea / Solution description	1.Monitoring the field parameters like pH,nutrient contents regularly and provide suggestion through an app to use fertilizer 2. Monitoring the temperature,moisture content of the soil to provide a efficient way to manage water
3.	Novelty / Uniqueness	Effective irrigation of the field using IOT techniques
4.	Social Impact / Customer Satisfaction	This proposed system provides many facilities,which help the farmer to maintain the crop field without much loss
5.	Business Model (Revenue Model)	This prototype can be developed as a product with minimum cost and high performance
6.	Scalability of the Solution	The design scale of the solution has been planned in a compact manner

3.4 Problem solution fit:

CUSTOMER SEGMENT(S) CS ❖ Farmers are our customer	**CUSTOMER CONSTRAINTS** CC ❖ Budget ❖ Network connection ❖ Lack of professional support	**AVAILABLE SOLUTIONS** AS ❖ Employing IOT technologies ❖ Remote monitoring of field
JOBS-TO-BE-DONE / PROBLEMS J&P ❖ To irrigate the field efficiently ❖ To increase the crop yield	**PROBLEM ROOT CAUSE** RC ❖ Irregular irrigation ❖ Poor soil quality ❖ Climatic conditions	**CHANNELS OF BEHAVIOUR** BE ONLINE: ❖ We notify the details about the farm field through a mobile application OFFLINE: ❖ Supplying fertilizers to the field by the user
EMOTIONS ❖ BEFORE: Worry about the crop production and yield ❖ AFTER: Better agricultural practice and increased productivity	**YOUR SOLUTION** ❖ Automating the irrigation process ❖ Regular monitoring of field parameters to improve the yield	**TRIGGERS** ❖ Improper water management ❖ Low productivity ❖ Neighbour farmers using modern techniques

4. REQUIREMENT ANALYSIS

4.1 functional requirement:

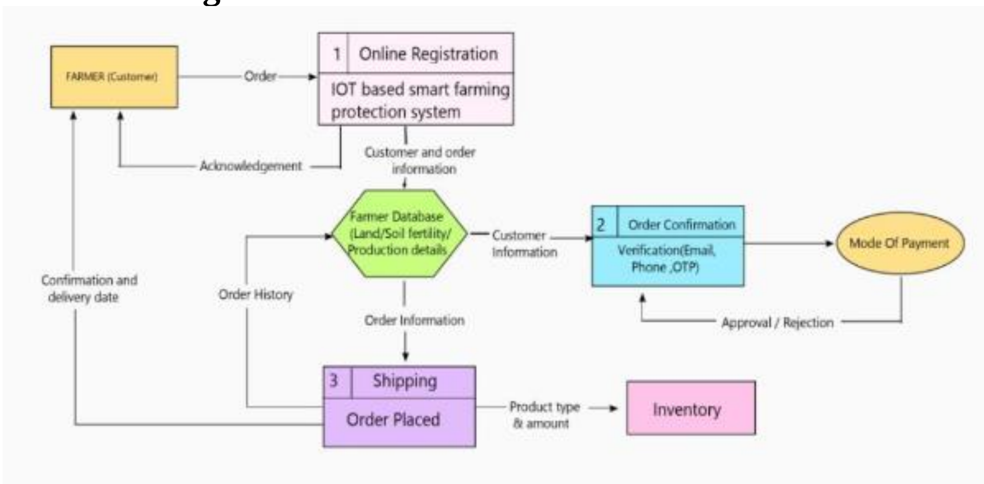
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Basic requirements	Smart phone with minimum 2GB RAM and 8GB ROM
FR-4	Access permission	User should enable their location, wifi.
FR-5	User details	Name, email id, mobile number.

4.2 Non functional requirement

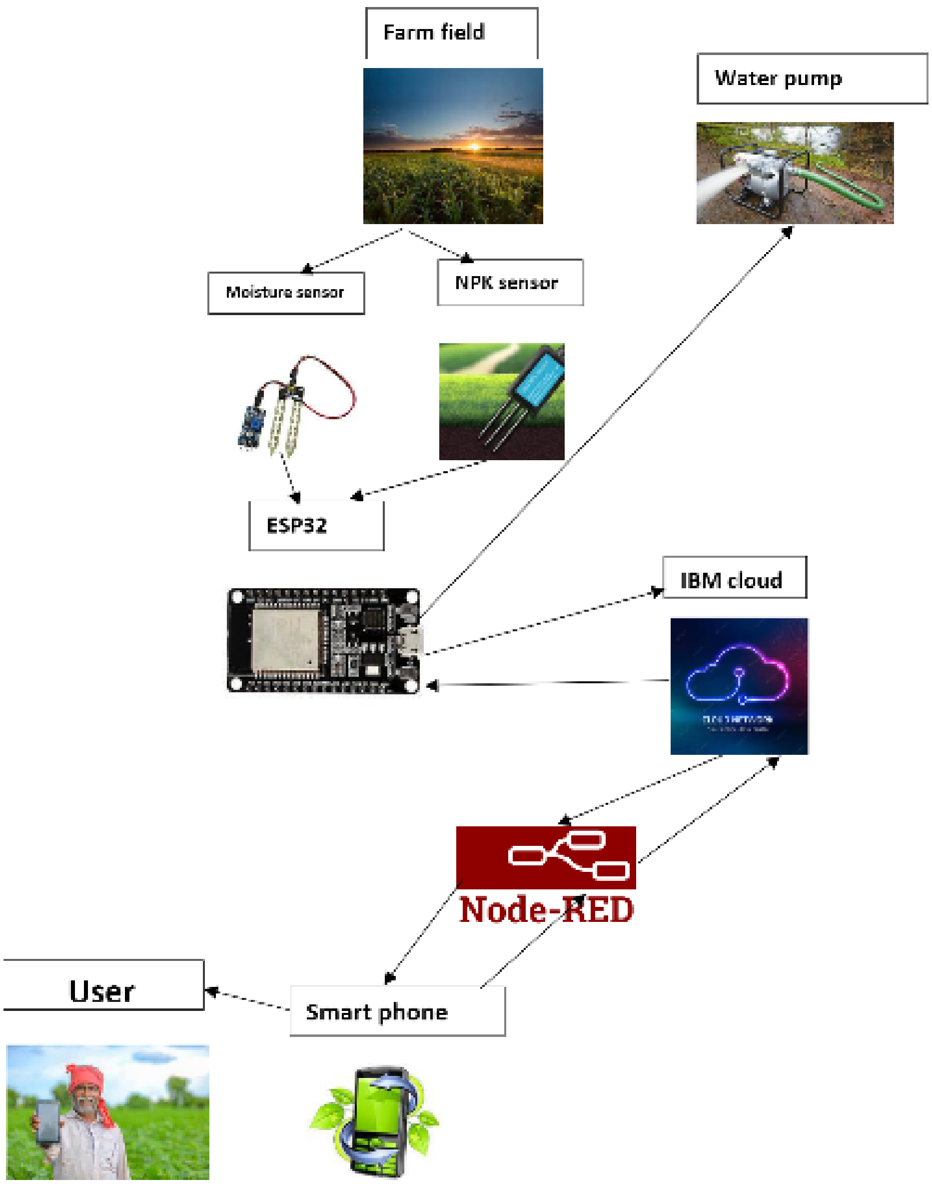
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	❖ Can be used in all agricultural fields ❖ Remote monitoring of the field ❖ Efficient way of irrigation
NFR-2	**Security**	❖ Access be available only to the authorized user ❖ All the details about the user are protected from the unathorized entity
NFR-3	**Reliability**	It can be accessed easily and accurately
NFR-4	**Performance**	It increases the quantity and quality of the crop
NFR-5	**Availability**	It is easily available at the presence of proper network connection
NFR-6	**Scalability**	It is scalable, but for a very large field ,we need more sensors

5.Project design

5.1 data flow diagram:



5.2 solution architecture:



5.3 user stories

User Type	Functional Requirement	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, and password	I can access my dashboard or account	High	Sprint 1
		USN-2	As a user, I will receive confirmation email once I have registered for application	I can receive confirmation mail and click confirm	Medium	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
		USN-5	As a user I can reset the password, if I forget the password		High	Sprint 1
	Dashboard	USN 6	As a user I want to see everything in single widget		Medium	Sprint 2
		USN 7	As a user I want a graphical representation		Low	Sprint 2
Customer(Web User)	Dashboard	USN-8	As a user I want a graphical representation of data for better understanding		High	Sprint 2
		USN-9	As a user I want to see a dashboard where I can customize myself		Low	Sprint 2
Customer(Mobile and Web)	Iot Device Setup	USN 10	As a user I need multiple sensors for better output		High	Sprint 2

		USN-11	As a user I need a low cost IoT devices for farming		High	Sprint 2
Customer Care Executive	User Problems	USN-12	As a user I should know how to use the application	There will be manual guide	Medium	Sprint 3
		USN-13	As a user I need my application to work on mobiles		High	Sprint 3
		USN-14	As a user if I am facing any issue in the application	There will be a query form	High	Sprint 3
Administrator	Query Clarification	USN-15	As a admin I give solutions to their queries		High	Sprint 3
	Particular Access	USN-18	As a admin I have to give access only to authorized persons		High	Sprint 3
Customer(Mobile User)	Application	USN-19	As a user I have to control my devices	Commands for devices	High	Sprint 4
			As a user I need more information about plants in application		Medium	Sprint 4

6 PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation creation and Python code development	USN-1	Connect Sensors Wi-fi Module with python code	2	High	Prabhu Rahul khan Sarveshwaran Sri aakash
Sprint-2	Connecting python code with IBM Watson platform and node-red work flow	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High	Prabhu Rahul khan Sarveshwaran Sri aakash
Sprint-3	Creating MIT App Inventor and designing front end like username and password	USN-3	Developing an application for the Smart farmer project using MIT App Inventor	2	High	Prabhu Rahul khan Sarveshwaran Sri aakash
Sprint-3	Developing the backend of the mit app using blocks	USN-3	Design the Modules and test the app	2	High	Prabhu Rahul khan Sarveshwaran Sri aakash
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Prabhu Rahul khan Sarveshwaran Sri aakash

6.2 Sprint Delivery Schedule:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed(as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	28 Oct 2022
Sprint-2	20	5 Days	31 Oct 2022	04 Nov 2022	20	03 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	4 Days	14 Nov 2022	17 Nov 2022	20	16 Nov 2022

TESTING

7.1 Test Cases

1. Verify user is able to see the Login/Signup popup when user clicked on My account button.
2. Verify the UI elements in Login/Signup popup.
3. Verify user is able to log into application with Valid credentials.
4. Verify user is able to log into application with InValid credentials.

7.2 User Acceptance Testing

7.2.1 Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

7.2.2 Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	77

7.2.3 Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8 RESULTS

8.1 Performance Metrics

Hence a helpful and useful system is built for farmers to assist them in farming and also prevent them from natural calamities. It also saves farmers time to maintain all these things as this is working on cloud he can turn on/off motor from anywhere so basically it helps farmers and make them relived thus helping our economy to grow.

9 ADVANTAGES & DISADVANTAGES

ADVANTAGE

9.2 Communicating the device at larger distance through web application. It will play an important role in reducing the man power and travelling expenses of a farmer.

9.3 Monitoring the parameter like temperature, humidity etc will play an important role in improving the growth of the plant.

DISADVANTAGE

1. Since the real time sensor will be connected to the controller, the controller requires continuous supply of internet to transfer the data.

10. CONCLUSION

The various parameters like temperature, humidity etc were monitored using web application. The device like motor can also controlled by the web application.

11. FUTURE SCOPE

1. The various data of soil nutrients is not added in the web browser, that can be added to the web application.
2. Controlling the device through mobile application will play important role in enhancing this project.
3. Providing the GPS and GIS information will also improve productivity of the farmer.

12. APPENDIX

12.1 Source Code

1)Python Code

tank

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


#Provide your IBM Watson Device Credentials
organization = "ptrr0s"
deviceType = "Obstacle"
deviceId = "100"
authMethod = "token"
authToken = "12345678"
dis = 0

def myOnPublishCallback():
    print("Published Level = %s %%" % dis, "to IBM Watson")

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status = cmd.data['command']
    if status == "MotorOn":
        print("Motor is running")
    elif status == "MotorOff":
        print("Motor is switched Off")
    else:
        print("please enter a valid 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:
        dis =random.randint(0,100)
        data = { 'Distance' : dis}
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if dis > 95 :
            print("Motor is switched Off, Tank Full!")
        if not success:
            print("Not connected to IoTF")
        deviceCli.commandCallback = myCommandCallback
        time.sleep(10)

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

field:

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


#Provide your IBM Watson Device Credentials
organization = "ptrr0s"
deviceType = "Field1"
deviceId = "1001"
deviceType1 = "Field2"
deviceId1 = "1002"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

'''elif status == "Servo2On":
        print ("Control valve for Field2 is Open!")
    elif status == "Servo2Off":
        print ("Control valve for Field2 is Closed!")'''

def myCommandCallback1(cmd):
    #print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "Servo2On":
        print ("Control valve for Field2 is Open!")
    elif status == "Servo2Off":
        print ("Control valve for Field2 is Closed!")
    #else:
    #    print("please enter a valid command")



def myCommandCallback(cmd):
    #print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "Servo1On":
        print ("Control valve for Field1 is Open!")
    elif status == "Servo1Off":
        print ("Control valve for Field1 is Closed!")
    #else:
    #    print("please enter a valid command")
```

```
    #print(cmd)

try:
        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"auth-method": authMethod, "auth-token": authToken}
        deviceOptions1 = {"org": organization, "type": deviceType1, "id": deviceId1,
"auth-method": authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
        deviceCli1 = ibmiotf.device.Client(deviceOptions1)
        #...............................................

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()
deviceCli1.connect()

def myOnPublishCallback():
    #print("Published Moisture = %s %%" % soil, "Nitrogen = %s %%" % N)#, "RandNo =
%s %%" % No, "to IBM Watson")
    print()

while True:
        soil = random.randint(0,100)
        N = random.randint(0,100)
        P = random.randint(0,100)
        Ka = random.randint(0,100)
        data = { 'Moisture' : soil, 'Nitrogen': N, 'Phosphorus' : P, 'Potassium' :
Ka}#, 'randomNumber' : No}
        #print("-----Field 1 Parameters-----")
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        soil = random.randint(0,100)
        N = random.randint(0,100)
        P = random.randint(0,100)
        Ka = random.randint(0,100)
        data = { 'Moisture' : soil, 'Nitrogen': N, 'Phosphorus' : P, 'Potassium' :
Ka}
        #print("-----Field 2 Parameters-----")
        success1 = deviceCli1.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        deviceCli.commandCallback = myCommandCallback
        deviceCli1.commandCallback = myCommandCallback1
        time.sleep(10)
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GitHub & Project Demo Link

GitHub : IBM-EPBL/**IBM-Project-4270-1658727092**

Project Demo Link **: https://drive.google.com/file/d/1Oz4RRL1SQM-UB0ZXId7anI_mqb8DDtRo/view?usp=sharing**