SmartFarmer - IoT Enabled Smart Farming Application

IBM- NALAIYA THIRAN

Team ID : PNT2022TMID12914

Team Leader Name : Karthikeyan K

Team Members:

Kavin M
Sri Vidya Niikethithan R
Vignesh P

Industry Mentor(s) Name: Bharadwaj

Faculty Mentor(s) Name: Suguna R

CHAPTER 1

INTRODUCTION

One of the largest livelihood providers in India is Agriculture. Agriculture plays an essential role in supporting human life. The rise in population is proportional to the increase in agriculture production. Basically, Agriculture production depends upon the seasonal situations which do not have enough water sources. To get beneficial results in agriculture and to overcome the problems, IoT based smart agriculture system is employed. Global and regional scale agricultural monitoring systems aim to provide up-to-date information regarding food production.

In IoT-based smart farming, a system is built for monitoring the crop field with the help of sensors like light, humidity, temperature, soil moisture, etc. The farmers can monitor the field conditions from anywhere. IoT-based smart farming is highly efficient when compared with the conventional approach. The proposed IoT based Irrigation System uses ESP8266 NodeMCU Module and DHT11 Sensor. It will not only automatically irrigate the water based on the moisture level in the soil but also send the Data to ThingSpeak’s Server to keep track of the land condition.

Due to the recent advances in sensors for the irrigation systems for agriculture and the evolution of WSN and IoT technologies, these can be applied in the development of automatic irrigation systems. The system will determine the parameters that are monitored in irrigation systems regarding water quantity and quality, soil characteristics, weather conditions, and fertilizer usage and provide an overview of the most utilized nodes and wireless technologies employed to implement WSN and IoT based smart irrigation systems.

1.1 Project Overview

The objectives of this report is to proposed IoT based Smart Farming System which will enable farmers to have live data of agriculture environment at very low cost so that live monitoring can be done.

- Gain knowledge of ThingSpeak’s IoT Platform. Connecting IoT devices to the ThingSpeak’s IoT platform and exchanging the sensor data.
- Gain knowledge using Node-RED for communicating with a mobile application. Creating a Mobile Application through which the user interacts with the IoT device.

Project Flow:

- The parameters like temperature, humidity, and soil moisture are updated to the ThingSpeak's IoT platform.
- The device will subscribe to the commands from the mobile application and control the motors accordingly.
- APIs are developed using Node-RED service for communicating with Mobile Application.
- A mobile application is developed using the MIT App inventor to monitor the sensor parameters and control the devices.
- Configure the Node-RED and create APIs for communicating with mobile application.
- Develop a mobile application to display the sensor parameters and control the motors.

1.2 Purpose:

Purpose of this project IoT enables easy collection and management of tons of data collected from sensors and with integration of cloud computing services like Agriculture fields maps, cloud storage etc., data can be accessed live from anywhere and everywhere enabling live monitoring and end to end connectivity among all the parties concerned. IoT is regarded as key component for Smart Farming as with accurate sensors and smart equipment's, farmers can increase the food production by 70% till year 2050 as depicted by experts. With IOT production costs can be reduced to e remarkable level which will in turn increase profitability and sustainability, IOT efficiency level would be increased in terms of usage of soil, Water, fertilizers, pesticides etc and various factors would be lead to the protection.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing Problem:

- Agriculture is the foundation of our Nation. In long time past days agriculturists used to figure the ripeness of soil and influenced presumptions to develop which to kind of product.
- They didn't think about the dampness, level of water and especially climate condition which horrible an agriculturist more.
- They utilize pesticides in view of a few suspicions which made lead a genuine impact to the yield if the supposition isn't right.
- The profitability relies upon the last phase of the harvest on which agriculturist depends.

2.2 References:

Author: Divya J , Divya M ,Janani V, H.A.C. Dharmagunawardhana, H.G.C.R. Laksiri, J.V. Wijayakulasooriya.

Benefits of Smart Farming:

Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.In large farmland, Internet of Things equipped drone helps to receive the current state of crops and send the live pictures of farmland.Analyzing farmland from the land using its Solutions you will know the current situation of fields and crops in.

According to studies:

86% of the studied farmers use some kind of “precision farming”. 95% acknowledged that “precision farming” is very helpful to use. 70% plan to expand their usage of "precision farming technologies". Challenges for Building the Internet of Things Platform A unified solution which can be integrated with different types of Internet of Things devices. The most common challenge for the Internet of Things in agriculture is connectivity. Every area doesn’t have proper internet connectivity. The second most common challenge for Internet of Things based Advanced Farming is the lack of awareness among consumers .Due to various service providers, it becomes really difficult to maintain interoperability between different IoT systems. A scalable solution that can be integrated with thousands of IoT devices for large farms. Solutions for Building IoT based Intelligent FarmingSmart Farming has enabled farmers to reduce waste and enhance productivity with the help of sensors (light, humidity, temperature, soil moisture, etc.) and automation of irrigation systems. Further with the help of these sensors, farmers can monitor the field conditions from anywhere. Internet of Things based Advanced Farming is highly efficient when compared with the conventional approach. The applications of intelligent Agriculture solutions not only targets conventional, large farming. With operations, but could also be new levers to uplift other growing or common trends in agricultural like organic farming, family farming (complex or small spaces, particular cattle and/or cultures, preservation of specific or high-quality varieties, etc.), and enhance highly transparent Farming.

Applications of Internet of Things in Smart Farming

In Internet of Things based smart agriculture, a system is formed to monitor the farmland with the help of sensors, which senses components like temperature, light, humidity, soil moisture, etc. Then, automate the irrigation system and allow farmers to monitor their field conditions from anywhere through IoT Analytics Platform. To make the agricultural process even smarter and accurate, precision agriculture is used. This makes agricultural practice more controlled and precise in terms of raising livestock and farming. Internet of Things based Advanced Farming plays a vital role when it comes to the use of IT and other elements like sensors, agricultural drones, autonomous vehicles, control systems, automated hardware, robotics, variable speed technology, and others. The below highlighted are the applications of Internet of Things in smart farming:

Weather Monitoring

Weather plays a very significant role when it comes to the Agriculture sector. In agriculture, there is almost everything dependable upon the climate condition. In smart Farming, temperature humidity, light intensity, and soil moisture can be monitored through various sensors. These are again used by the reactive system to trigger alerts or automate the process such as water and air control.

Smart Irrigation on Agriculture Land

In smart irrigation, automated sprinkler systems or intelligent pumps are used. Soil moistures sensors are used in different areas to get the moisture of the soil in agricultural land. Based on the results from the soil moisture sensors, the intelligent pumps or intelligent sprinklers are turned On/Off.

Monitoring Soil Quality

Farmers usually use a sampling method to calculate soil fertility, moisture content. Fortunately, this sampling doesn't give accurate results as chemical decomposition varies from location to location. Meanwhile, this not much helpful. To resolve this thing, it plays an essential role in Farming. Sensors can be installed at a uniform distance across the length and breadth of the farmland to collect the accurate soil data, which can be further used in the dashboard or mobile application for the farm monitoring.

Livestock Monitoring

Internet of Things devices can be used to collect data regarding the location, well-being, and the health of the cattle. This data can be further used for identification of the sick animals so that they can be separated from the others, thereby preventing the spread of diseases. This Live Stock Monitoring also lowers the labour costs with the help of Internet of Things based sensors.

Drone Monitoring

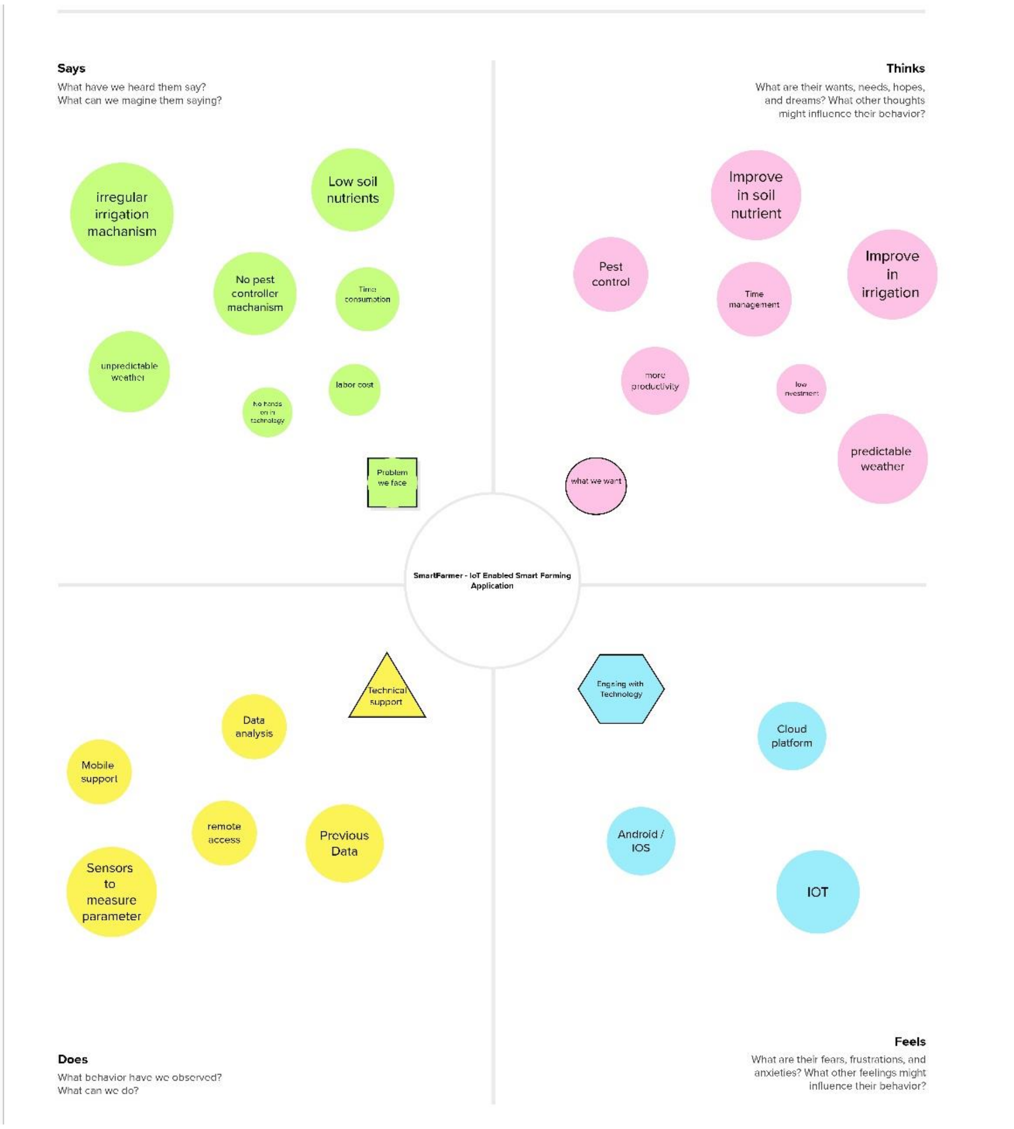
Drone monitoring is helping large farms to reduce the cost of monitoring, or the use of Geo-positioning sensors can set a stable path. Moreover, the data collected from these drones are sent back to the server where it can be used for analysing and decision-making.

2.3 Problem Statement Definition:

IoT based SMART FARMING SYSTEM is regarded as IoT gadget focusing on Live Monitoring of Environmental data in terms of Temperature, Moisture and other types depending on the sensors integrated with it. The system provides the concept of “Plug & Sense” in which farmers can directly implement smart farming by as such putting the System on the field and getting Live Data feeds on various devices like Smart Phones, Tablets etc. and the data generated via sensors can be easily shared and viewed by agriculture consultants anywhere remotely via Cloud Computing technology integration. The system also enables analysis of various sorts of data via Big Data Analytics from time to time.

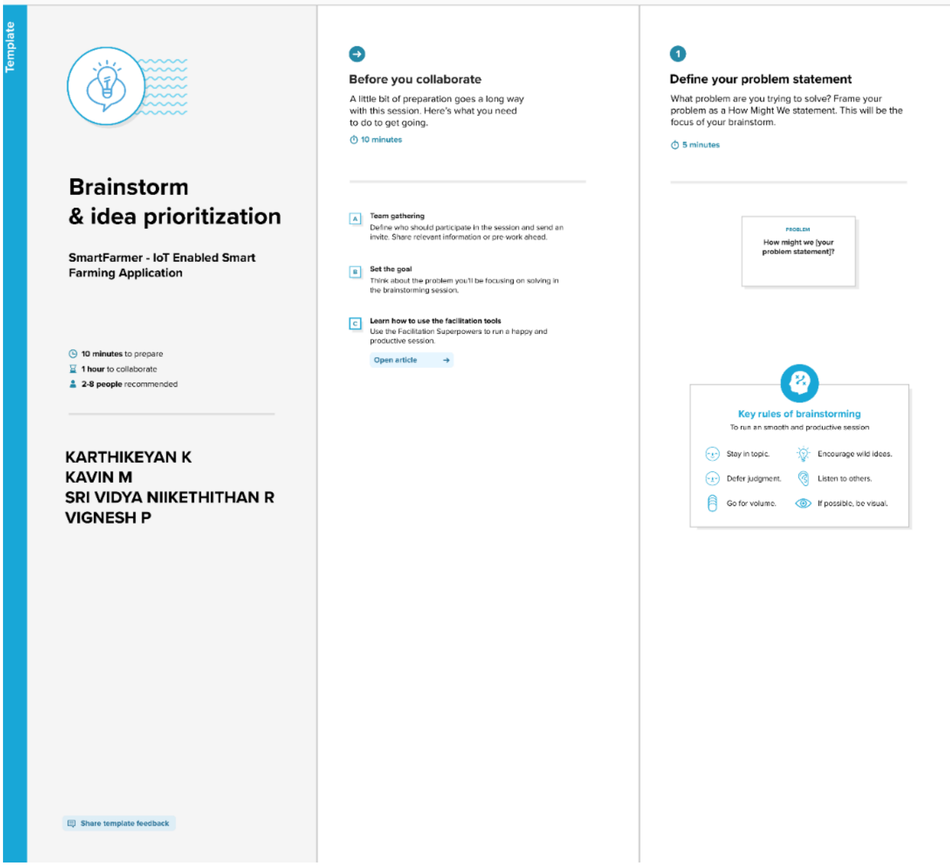
CHAPTER 3
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

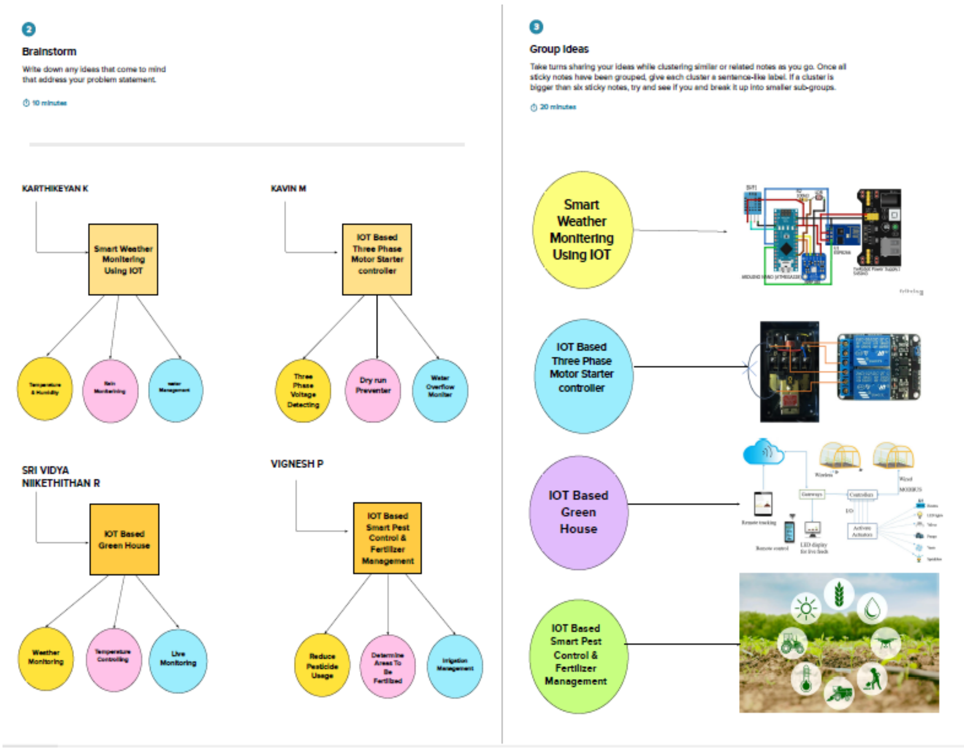


3.2 Ideation & Brainstorming

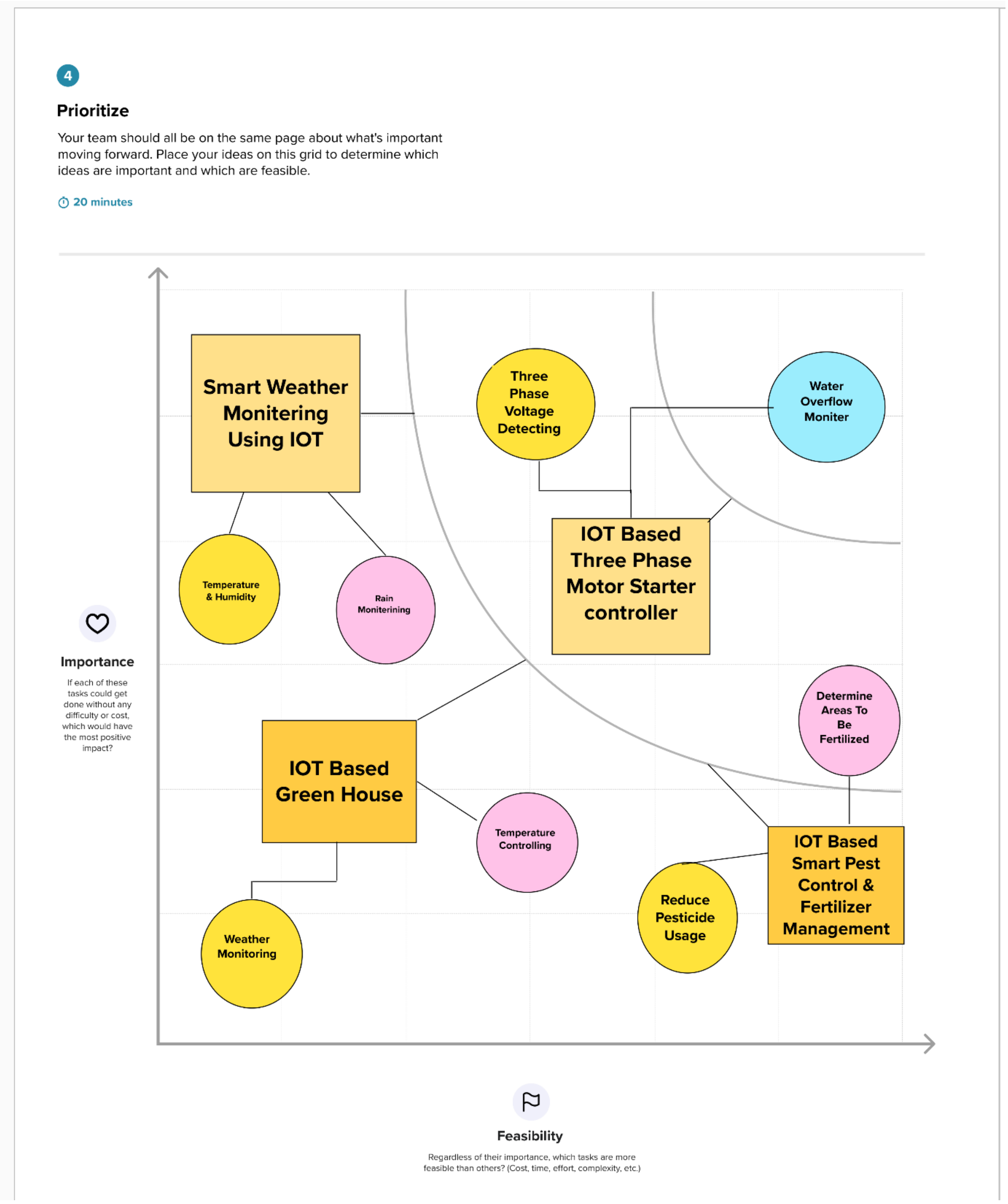
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea-Prioritization



3.3 Proposed Solution

Proposed Solution :

S.No.	Parameter
1.	Problem Statement (Problem to be solved)
2.	Idea / Solution description
3.	Novelty / Uniqueness
4.	Social Impact / Customer Satisfaction
5.	Business Model (Revenue Model)
6.	Scalability of the Solution

Problem Statement (Problem to be solved)

The traditional agriculture and allied sector cannot meet the requirements of modern agriculture which requires high-yield, high quality and efficient output. Thus, it is very important to turn towards modernization of existing methods and using the information technology and data over a certain period to predict the best possible productivity and crop suitable on the very particular land. The adoptions of access to high-speed internet, mobile devices, and reliable, low-cost satellites (for imagery and positioning) are few key technologies characterizing the precision agriculture trend.

Idea / Solution description

IOT based agriculture is the only solution for the high productivity from agriculture industries with efficient usage of data available. Internet-of-Things (IoT) technology have enabled a new paradigm of smart farming managing agricultural produce, land, and animals more effectively and efficiently. The most critical requirement in developing smart agriculture is the inclusion of “Things”-oriented sensor network systems, which require sensors with high integration, intelligence, ubiquitous connectivity, security, and energy independence. The focus is on the implementation of edge-based devices for producing, processing, and transferring a large amount of data which can be reliably utilized to interpret the past and predict the future, in order to make more timely or accurate decisions, both in sustainable/productive agriculture and in livestock applications.

Novelty / Uniqueness

Data, tons of data, collected by smart agriculture sensors, e.g. weather conditions, soil quality, crop’s growth progress or cattle’s health. This data can be used to track the state of business in general as well as staff performance, equipment efficiency, etc.
Better control over the internal processes and, as a result, lower production risks.
The ability to foresee the output of your production allows you to plan for better product distribution. If you know exactly how much crops you are going to harvest, you can make sure

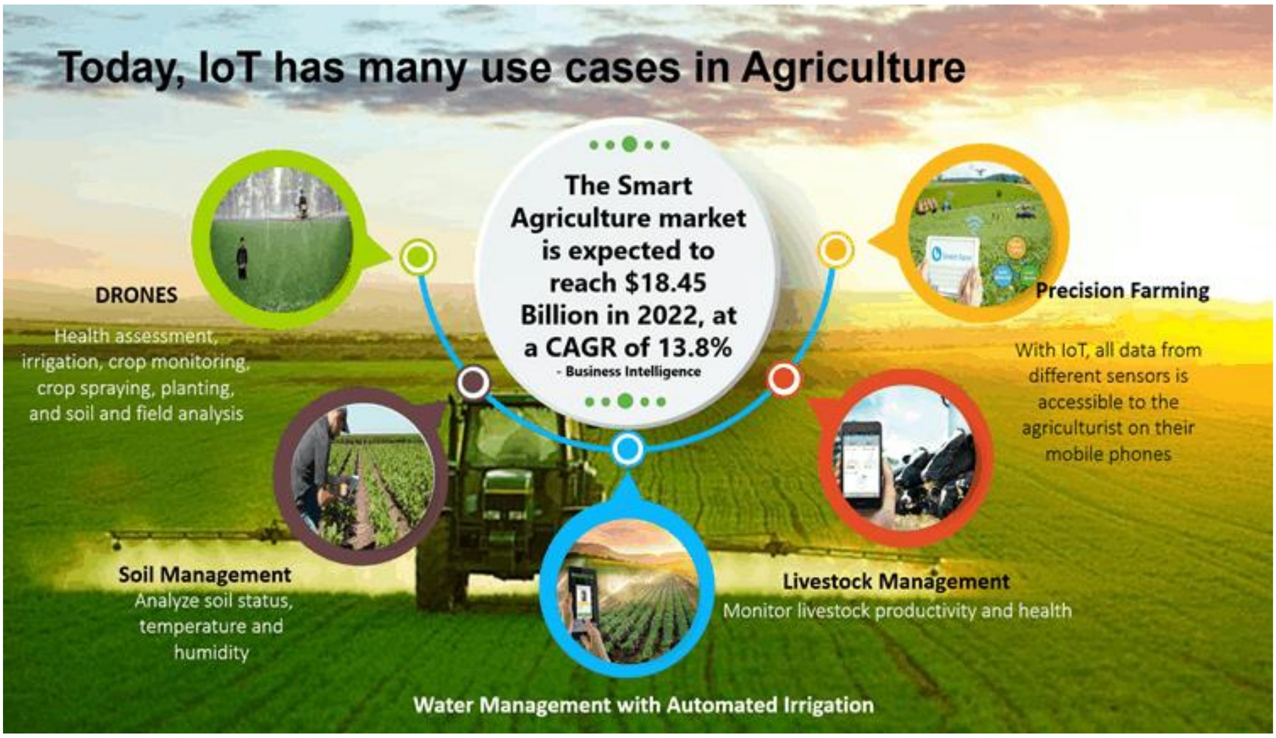
your product won't lie around unsold. Cost management and waste reduction thanks to the increased control over the production. Being able to see any anomalies in the crop growth or livestock health, you will be able to mitigate the risks of losing your yield. Increased business efficiency through process automation. By using smart devices, you can automate multiple processes across your production cycle, e.g. irrigation, fertilizing, or pest control. Enhanced product quality and volumes. Achieve better control over the production process and maintain higher standards of crop quality and growth capacity through automation.

Social Impact / Customer Satisfaction

IOT based agriculture application has high impact on society where it gives new hope to the modern farmer in form of high productivity in low cost and efficient work experience.

Customer Satisfaction improves more in IOT agriculture as the proper handle of resource are used in limited time with efficient manner.

Business Model (Revenue Model)



Scalability of the Solution

3.4 Problem Solution fit

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Based on scale:
Small, micro, Marginal
Base on farm:
Surplus
Gross revenue
Land under cultivation
Base on service:
Commercial product
Small scale supplyment.

6. CUSTOMER CONSTRAINTS CC

The country's sustainable agricultural development has many obstacles
Agricultural
water-use shortage
Cultivated land loss
usage of fertilizers and pesticides
Environmental degradation
Poor soil nutrient

5. AVAILABLE SOLUTIONS AS

IoT in agriculture uses robots, drones, remote sensors, and computer imaging combined with continuously progressing machine learning and analytical tools for monitoring crops, surveying, and mapping the fields, and providing data to farmers for rational farm management plans to save both time and money

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

1. Monitoring of climate conditions and weather
2. handling of Greenhouse automation
3. Crop management and measurement
4. Cattle monitoring and management
5. Precision farming with efficiency
6. Agricultural drones and sprayers
7. Predictive analytics for smart farming
8. End-to-end farm management systems

9. PROBLEM ROOT CAUSE RC

1. Connectivity in rural areas and remote access
2. Cope with climate change, soil erosion and biodiversity loss
3.customer's changing tastes and expectations
4. Meet rising demand for more food of higher quality
5. High adaptive cost
6. Lack of information
7. Social impacts and customer satifaction
8.

7. BEHAVIOUR BE

Online: farmers can monitor all the sensor parameters by using web or mobile application even if the farmer is not near his field Agriculture has been mainly of an economic nature but the quite different social approach has grown more recently.

Focus on J&P, tap into BE, understand RC

3. TRIGGERS TR

Optical information, virtual fence technologies allow cattle herd management based remote-sensing signals and sensors or actuators attached to the livestock

4. EMOTIONS: BEFORE / AFTER EM

Increased production: the optimization of all the processes related to agriculture and livestock- rearing increases production rates.
Water saving: weather forecasts and sensors that measure soil moisture and for the right length of time.
BEFORE: Farmers are feeling helpless ,frustrated in order to be always present to manage end to end farming
AFTER: Farmers will feel much relaxed as well as encouraged ,if production output increases

10. YOUR SOLUTION SL

Sensor : Interact with physically environment and measure the physical parameter sends the data top IOT platform.
Where IOT platform analysis the given data.
Sensors and biosensors in this context refer to devices that ensure data about a specific physical, chemical.
The global positioning system(GPS): is satellite based standard sensing technology used for tracking farm animal's location.
we use iot enabled products to provide knowledge as well as guidelines, help
,amount of usage of modern tools ,sensors to detect problems ,iot devices provides reminder incase of emergencies , better product network with iot connection .

8. CHANNELS of BEHAVIOUR CH

8.1 **ONLINE**
Configuration of device with IOT platform gathering information and processing the information with previous values.
Sending back the message to motor or other device on the field
Remote access availability.

8.2 **OFFLINE**
The fixing of sensor in the right place with proper protection of sensors. The sensitivity of sensors may be affect by external things.
Fixing of movable motors and irrigation system in the center places which covers maximum area with minimum wastage of water .

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR NO.	FUNCTIONAL REQUIREMENT (EPIC)	SUB-REQUIREMENT (STORY /SUB-TASK)
1.	User Registration	Registration through Form
2.	User Confirmation	Confirmation via Email Confirmation via OTP
3.	User Login	Log in with your Email Id and Password
4.	Forgot Password	Log in with Email Confirmation Of OTP
5.	Feedback Form	Make a note of the problems and issues faced by the user when using the application
6.	Basic information (Weather, soil type)	To find the basic information of a particular area
7.	Agriculture Note	To list agriculture-related information like how to plant, how many litters of water that plant needs in a day etc
8.	Sensors	To show various data from different sensors like temperature, humidity, soil moisture
9.	Database Management	To show various agriculture-related data are stored
10.	Data analysis	Analysing of data and processing the data to the particular agriculture area
11.	Control remote device	Using the data analysed the device in remote region is controlled either automatically or by the user
12.	Exit	After the user checked the information, the user can exit the application

4.2 Non-Functional requirements

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

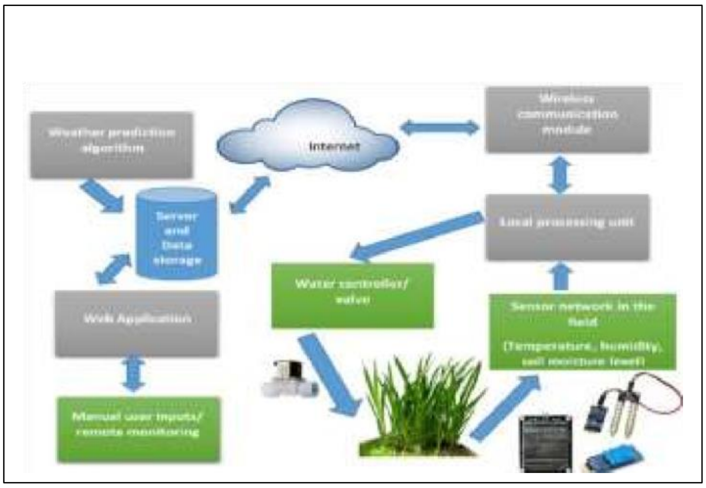
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• User friendly guidelines for users to avail the features. • Most simplistic user interface for ease of use.
NFR-2	**Security**	• All the details about the user are protected from unauthorized access. • Detection and identification of any misfunctions of sensors.
NFR-3	**Reliability**	• Implementing Mesh IoT Networks • Building a Multi-layered defense for IoT Networks.
NFR-4	**Performance**	The use of modern technology solutions helps to achieve the maximum performances thus resulting in better quality and quantity yields.
NFR-5	**Availability**	This app is available for all platforms
NFR-6	**Scalability**	Scalability refers to the ability to increase available resources and system capability without the need to go through a major system redesign or implementation.

CHAPTER 5
PROJECT DESIGN

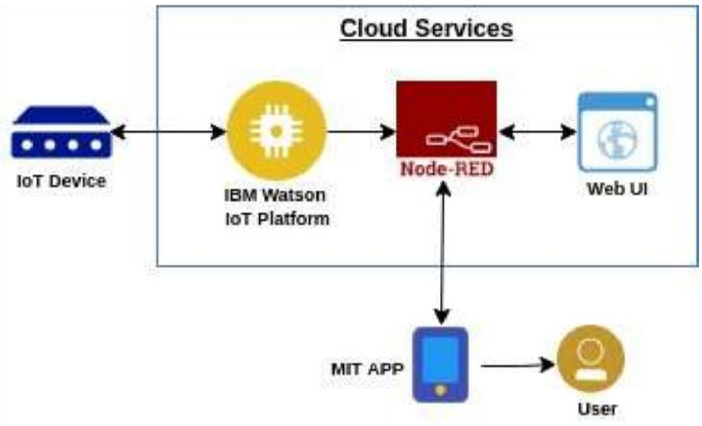
5.1 Data Flow Diagrams

Data Flow Diagrams:

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)



5.2 Solution & Technical Architecture

Solution Architecture:

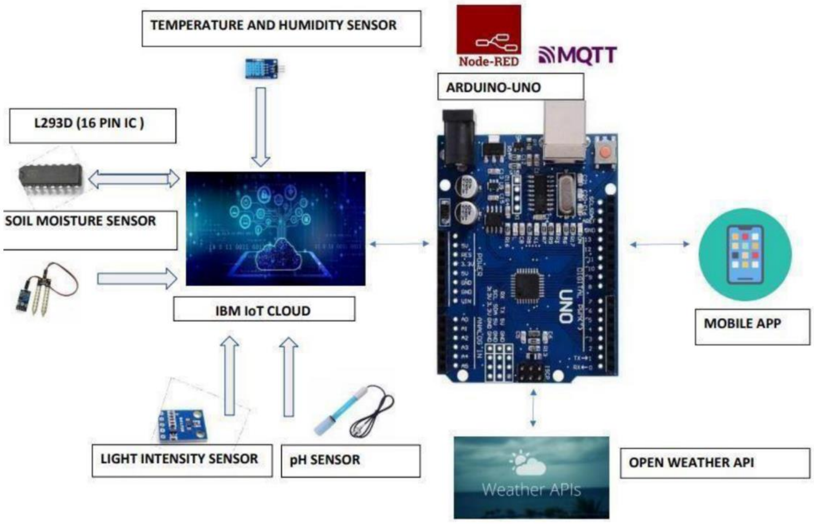
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:

SOFTWARE FOR SMART AGRICULTURE

FUNCTIONS Smart &Green



Technical Architecture:

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

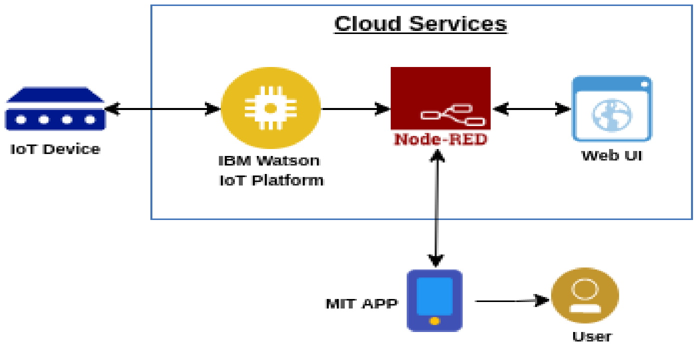


Table-1 : Components & Technologies:

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.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
4.	Cloud Database	Database Service on Cloud	IBM cloud
5.	Temperature sensor	Monitors the temperature of the crop	
6.	Humidity sensor	Monitors the humidity	
7.	Soil moisture sensor (Tensiometers)	Monitors the soil temperature	
8.	Weather sensor	Monitors the weather	.
9.	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

5.3 User Stories

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	IoT devices	USN-1	Sensors and wi-fi module		High	Sprint-1
Customer	Software	USN-2	IBM Watson IoT platform, Workflows for IoT scenarios using Node-red		High	Sprint-2
Customer	MIT app	USN-3	To develop an application using MIT		High	Sprint-3
Customer	Web UI	USN-4	To make the user to interact with the software.	User can access the app for the services.	High	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement(Epic)	User Story Number	User Story /Task	Story Points	Priority	Team Member
Sprint-1	Simulation creation	UNS-1	Connect sensor and Arduino board and explain details to be covered in upcoming sprints	2	High	Karthikeyan K (Leader)
Sprint-1	Software	UNS-2	Design of software and use case of software	1	High	Sri vidya Niikethithan R
Sprint-2	MIT APP	UNS-3	Development of MIT APP and developer test of MIT application	3	High	Sri vidya Niikethithan R
Sprint-1	Design of Login interface	UNS-4	Creation of user Login to the MIT application	2	High	Kavin M
Sprint-3	Registration (Farmer - WebUser)	USN - 1	Registration of user to access the application	3	High	Vignesh P
Sprint - 2	Login	USN - 2	Testing of application by logging In	3	High	Srividya Niikethithan R , Karthikeyan K, Kavin M

						Vignesh P
Sprint - 4	Web UI	USN - 3	Development of Web UI for user interface for admin	3	High	Srividya Niikethithan R , Karthikeyan K, Kavin M Vignesh P
Sprint - 1	Gathering of specification needed	USN - 1	Getting the problem faces by customer and implement of feature	2	High	Kavin M
Sprint - 4	Login and integration application	USN - 2	Testing of application and integration of whole application	3	High	Srividya Niikethithan R , Karthikeyan K, Kavin M, Vignesh P
Sprint - 3	Web UI	USN - 3	Enhancement of portal of user Interface	3	Medium	Karthikeyan K, Vignesh P
Sprint - 1	Design of platform and connection of devices	USN - 1	Design of platform and creation of connection of devices	1	High	Vignesh P
Sprint - 4	Final integration of software	USN - 2	Finally bug blitz check of application	2	High	Srividya Niikethithan R , Karthikeyan K, Vignesh P

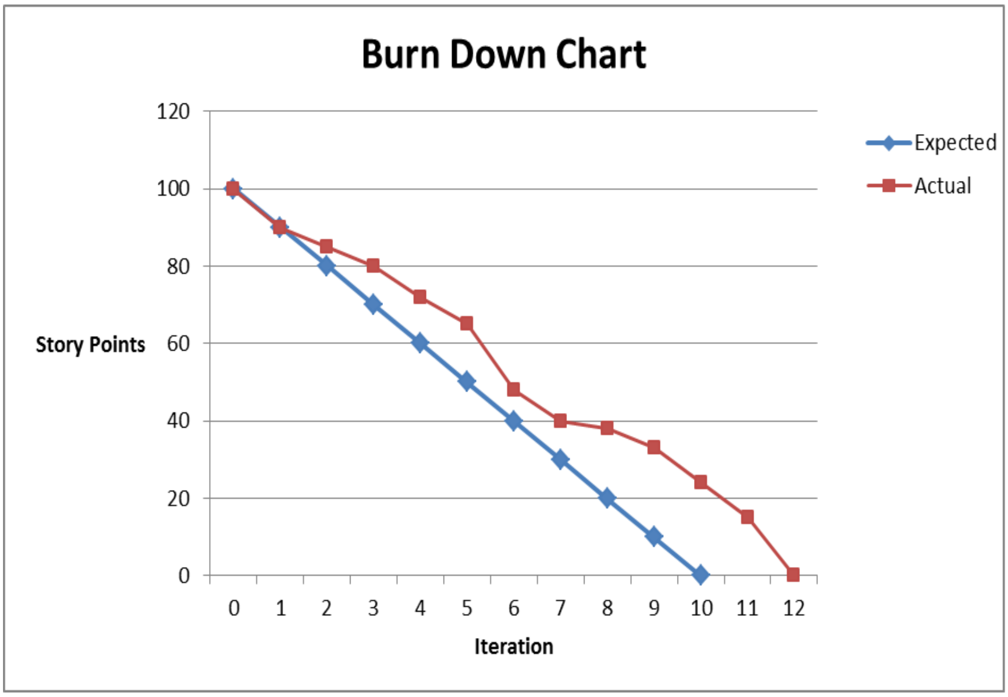
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	12	6 Days	20 Oct 2022	25 Oct 2022	20	25 Oct 2022
Sprint-2	6	6 Days	25 Oct 2022	01 Nov 2022	20	01 Nov 2022
Sprint-3	6	6 Days	03 Nov 2022	08 Nov 2022	20	06 Nov 2022
Sprint-4	6	6 Days	09 Nov 2022	14 Nov 2022	20	13 Nov 2022

Velocity:

AV for sprint 1=Sprint Duration /velocity =12/6=2AV for sprint 2= Sprint Duration/Velocity =6/6=1 AV for Sprint 3=Sprint Duration/Velocity=6/6=1 AV for Sprint 4=Sprint Duration/ Velocity=6/6=1

Burndown Chart:



6.2 Sprint Delivery Schedule

S.NO	ACTIVITY TITLE	ACTIVITY DESCRIPTION	DURATION
1	**Understanding the project**	Assign the team members after that create repository in the GitHub and then assign task to each member and guide them how to access the GitHub while submitting the assignments	1 week
2	**Staring The Project**	Team Members to Assign All the Tasks Based on Sprints and Work on It Accordingly.	1 week
3	**Completing Every Task**	Team Leader should ensure that whether every team member have completed the assigned task or not	1 week
4	**Stand Up Meetings**	Team Lead Must Have a Stand-Up Meeting with The Team and Work on The Updates and Requirement Session	1 week
5	**Deadline**	Ensure that team members are completing every task within the deadline	1 week
6	**Budget and Scope of project**	Analyze the overall budget which must be within certain limit it should be favorable to every person	1 week

7. CODING

```
// Sprint 1:  For Reference
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

# IBM
organization = "“jztdcw”"
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"
# Gpio
def mycommandCallback(cmd):
    print("Command Received: %s" % cmd.data["command"])
    status = cmd.data["command"]
    if status == "lighton":
        print("LED is ON")
    elif status == "lightoff":
        print("LED 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()
# CONNECCT
deviceCli.connect()
while True:
    temp = random.randint(0, 100)
    hum = random.randint(0, 100)
    data = {"temp": temp, "hum": hum}
```

```
    def myOnPublishCallback():
        print(
            "Published Temperature = %s C" % temp,
            "Humidity = %s %%" % hum,
            "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
deviceCli.disconnect()

// Sprint 3:  For Reference
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device
#Provide your IBM Watson Device Credentials
organization = "jztdcw"
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"
def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("Turn Motor ON")
    elif cmd.data['command']=='motoroff':
        print("Turn Motor OFF")
    elif cmd.data['command']=='lighton':
        print("Turn Light ON")
    elif cmd.data['command']=='lightoff':
        print("Turn Light OFF")

    if cmd.data['command']=='ACTIVATE IRRIGATION':
        print("TurnON")
    elif cmd.data['command']=='DEACTIVATE IRRIGATION':
        print("TurnOFF")
    elif cmd.data['command']=='HIGH TEMPERATURE':
```

```
        print("TurnON")
    elif cmd.data['command']=='LOW TEMPERATURE':
        print("TurnOFF")

    if cmd.data['command']=='BAD WEATHER':
        print("TurnON")
    elif cmd.data['command']=='GOOD WEATHER':
        print("TurnOFF")
    elif cmd.data['command']=='HUMIDITY HIGH':
        print("TurnON")
    elif cmd.data['command']=='HUMIDITY LOW':
        print("TurnOFF")

    if cmd.command == "setInterval":
        if 'interval' not in cmd.data:
        print("Error - command is missing required information: 'interval'")
    else:
        interval = cmd.data['interval']
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
        else:
             output=cmd.data['message']
    print(output)
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:
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()

// Code For Temperature
```

```
from random import *
from random import *
while True:
    temperature = randrange(0,100)
    humidity = randrange(0,100)
    if (temperature>=50):
        print("Alarm")
    else:
        print("No Alarm")

// Python Code
import random as rand
print("WELCOME SMART FARMER")
temperature = float(rand.uniform(15, 50))
if temperature > 22 and temperature < 40:
    humidity = int(rand.randint(45, 65))
elif temperature < 22:
    humidity = int(rand.randint(60, 70))
elif temperature > 40:
    humidity = int(rand.randint(25, 35))
moisture = int(rand.randint(00, 70))
print(
    "temperature:",
    temperature,
    "C",
    "\n",
    "humidity:",
    humidity,
    "\n",
    "moisture:",
    moisture,
)
if temperature > 35 or moisture < 20:
    print("Irrigation required")
    print("Activate irrigation ?")
    decision = input()
    if decision == "yes":
        print("Irrigation activated")
    else:
        print("Irrigation not activated")
else:
    print("Irrigation not required")
```

```
// Sprint 4:  For Reference

import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "jztdcw"
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %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 an
event of type "greeting" 10 times deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
    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():
```

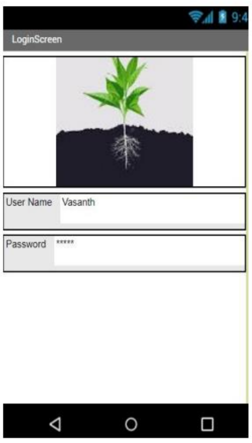
```
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, “Moisture =%s deg c” % Mois “to IBM Watson")
        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
    time.sleep(10)
   deviceCli.commandCallback = myCommandCallback
#Disconnect the device and application from the cloud
deviceCli.disconnect()
```

CHAPTER 8

TESTING

8.1:Test cases

Testing of the Smart Farming IOT Based Agriculture is done by covering all possible edges by using artificial environment and tested the test cases by using manual testing.



S.NO	Scenario	Input	Excepted output	Actual output
1	User login	User name and password	Login	Login success.
2.	Sign up	User should register by their details	Login to landing page	Login success
3.	Data visualisation	Data from IOT platform	Trigger from MIT	State of devices in field
4.	State of devices	Trigger from MIT App	Working of devices in field	Worked successfully

8.2 User Acceptance testing

User Acceptance Testing is carried out in a separate testing environment. A change, an update, or a new feature is requested and developed. Unit and integration tests were run. All seems to be in order.

These tests are conducted individually by the team member before releasing production version and all test cases seems to in tracked. The test cases results in positive result at the end of testing

CHAPTER 9

RESULT

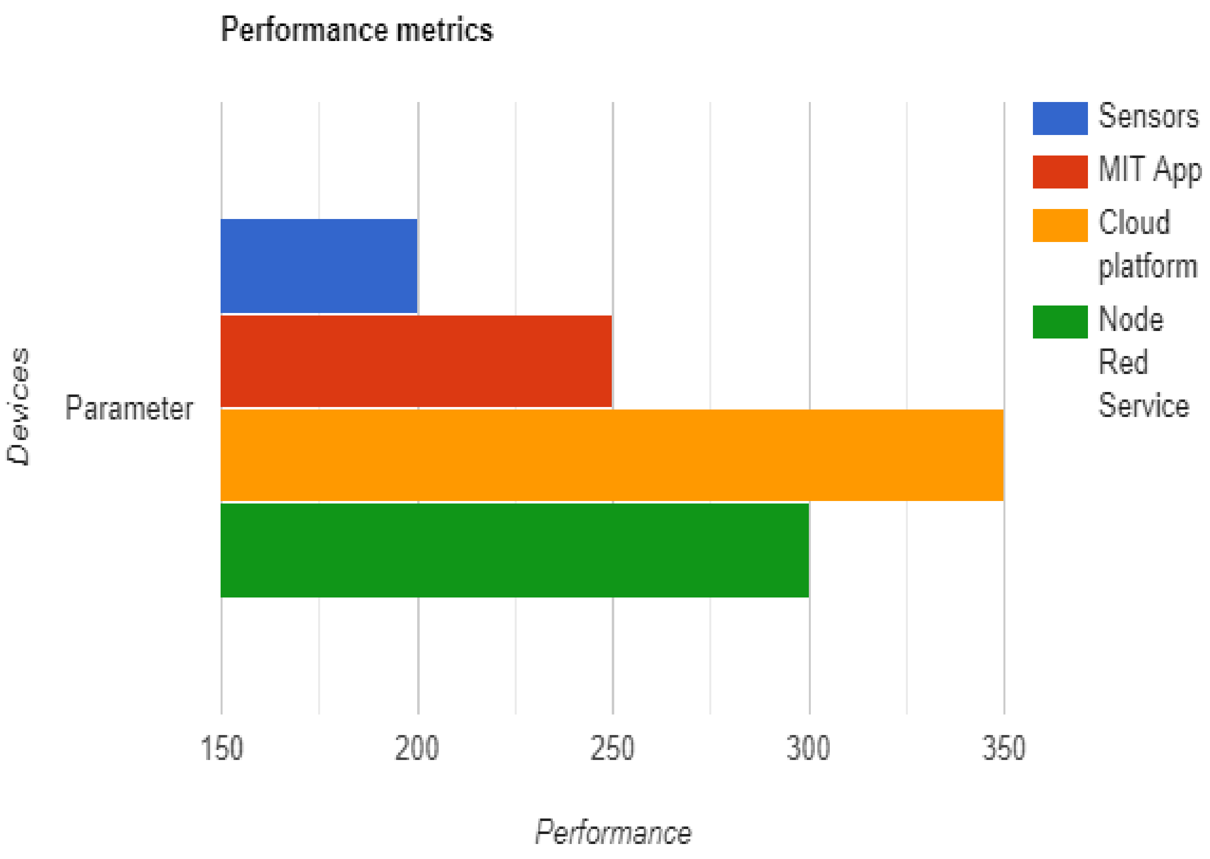
Smart Farming IOT Based Agriculture Stick for Live Monitoring of Temperature and Soil Moisture has been proposed using sensors, WIFI Module, and various other Hardware Devices along with the Watson IoT platform and Node Red Services are implemented. The stick has high efficiency and accuracy in fetching the live data of temperature, humidity, and soil moisture. The IoT-based Agriculture stick being developed through this paper will help farmers in increasing the agriculture yield and take efficient care of food production as the stick will always provide a helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with accurate results. With the help of these systems, various problems faced by farmers in daily life are being solved to a greater extent.

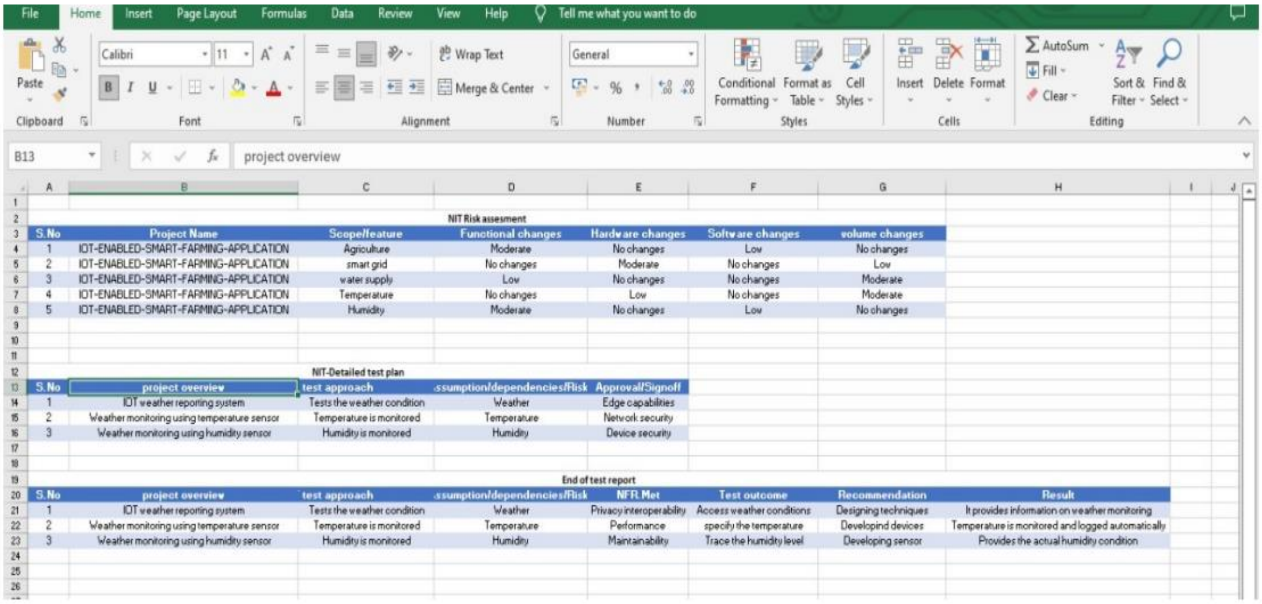
Therefore, this system avoids excessive irrigation, under irrigation, soil erosion, and reduces water wastage. The main advantage is that the action of the system can be changed depending on the situation (plants, climate, soil, etc.). Through this program, agriculture, agricultural fields, parks, gardens, golf courses can be measured. Therefore, this program is cheaper and more efficient compared to other types of automation systems. For larger applications, higher sensitivity can be performed in large areas of agricultural land. A soil moisture level monitoring system was developed and the project provided an opportunity to study existing systems, as well as their features and constraints. The proposed system can be used to turn on / off the water spray according to soil moisture levels thus making the irrigation process one of the most time-consuming agricultural activities. Agriculture is one of the biggest uses of water.

The main aim of this project is to implement the modern technology in required fields like agriculture. Using IoT technology in agriculture, this system makes agriculture monitoring easy. The benefits as mentioned like water saving and labour saving are required the maximum incurrent agricultural state of affairs. Consequently, using the sensor network in fields of agriculture makes clever irrigation. The information from IoT is sent to the client using cloud.

Consequently, any changes inside the crop may be identified effortlessly and early analysis is achieved as such.IoT will help to enhance smart farming. Using IoT the system can predict the soil moisture level and humidity so that the irrigation system can be monitored and controlled. IoT works in different domains of farming to improve time efficiency, water management, crop monitoring, soil management and control of insecticides and pesticides. This system also minimizes human efforts, simplifies techniques of farming and helps to gain smart farming. Besides the advantages provided by this system, smart farming can also help to grow the market for farmer with single touch and minimum effort.

9.1 Performance Metrics





NIT Risk assesment

S.No	Project Name	Scope/feature	Functional changes	Hardware changes	Software changes	volume changes
1	IOT-ENABLED-SMART-FARMING-APPLICATION	Agriculture	Moderate	No changes	Low	No changes
2	IOT-ENABLED-SMART-FARMING-APPLICATION	smart grid	No changes	Moderate	No changes	Low
3	IOT-ENABLED-SMART-FARMING-APPLICATION	water supply	Low	No changes	No changes	Moderate
4	IOT-ENABLED-SMART-FARMING-APPLICATION	Temperature	No changes	Low	No changes	Moderate
5	IOT-ENABLED-SMART-FARMING-APPLICATION	Humidity	Moderate	No changes	Low	No changes

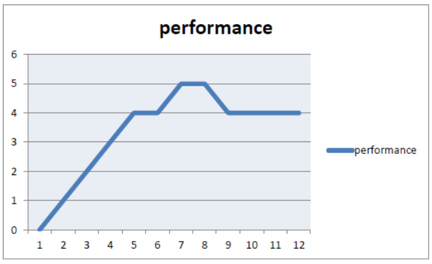
NIT-Detailed test plan

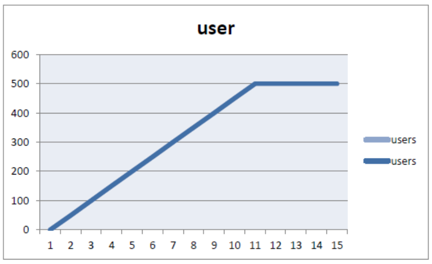
S.No	project overview	test approach	ssumption/dependencies/Risk	Approval/Signoff
1	IOT weather reporing system	Tests the weather condition	Weather	Edge capabilities
2	Weather monitoring using temperature sensor	Temperature is monitored	Temperature	Network security
3	Weather monitoring using humidity sensor	Humidity is monitored	Humidity	Device security

End of test report

S.No	project overview	test approach	ssumption/dependencies/Risk	NFR Met	Test outcome	Recommendation	Result
1	IOT weather reporing system	Tests the weather condition	Weather	Privacy interoperability	Access weather conditions	Designing techniques	It provides information on weather monitoring
2	Weather monitoring using temperature sensor	Temperature is monitored	Temperature	Performance	specify the temperature	Developind devices	Temperature is monitored and logged automatically
3	Weather monitoring using humidity sensor	Humidity is monitored	Humidity	Maintainability	Trace the humidity level	Developing sensor	Provides the actual humidity condition

Performance and User insight graph:





CHAPTER 10

ADVANTAGES AND DISADVANTAGES

10.1 Advantages

- **Efficiency:-** IOT-enabled agriculture allows farmers to monitor their products and conditions in real-time. They get insights fast, can predict issues before they happen, and make informed decisions on how to avoid them. Additionally, IoT solutions in agriculture introduce automation, for example, demand-based irrigation, fertilizing, and robot harvesting.
- **Expansion:-** By the time we have 9 billion people on the planet, 70% of them will live in urban areas. IoT-based greenhouses and hydroponic systems enable short food supply chains and should be able to feed the people. Smart closed-cycle agricultural systems allow growing food everywhere—in supermarkets, on skyscrapers' walls and rooftops, in shipping containers, and, of course, in the comfort of everyone's home.
- **Reduced resources:-** Plenty of IoT solutions are focused on optimizing the use of resources—water, energy, land. Precision farming using IoT relies on the data collected from diverse sensors in the field which helps farmers accurately allocate just enough resources within one plant.
- **Clean process:-** Not only do IOT-based systems for precision farming help producers save water and energy and, thus, make farming greener, but also significantly scale down on the use of pesticides and fertilizer. This approach allows getting a cleaner and more organic final product compared to traditional agricultural methods.
- **Agility:-** One of the benefits of using IoT in agriculture is the increased agility of the processes. Thanks to real-time monitoring and prediction systems, farmers can quickly respond to any significant change in weather, humidity, air quality as well as the health of each crop or soil in the field. In the conditions of extreme weather changes, new capabilities help agriculture professionals save the crops.
- **Improved product quality:-** Data-driven agriculture helps both grow more and better products. Using soil and crop sensors, aerial drone monitoring, and farm mapping, farmers better understand detailed dependencies between the conditions and the quality of the crops. Using connected systems, they can recreate the best conditions and increase the nutritional value of the products.

10.2 Disadvantages

- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

- The smart farming based equipments require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

- Security : As the IoT systems are interconnected and communicate over networks. The system offers little control despite any security measures, and it can be lead the various kinds of network attacks.

- Privacy: Even without the active participation on the user, the IoT system provides substantial personal data in maximum detail.

- Complexity: The designing, developing, and maintaining and enabling the large technology to IoT system is quite complicated.

CHAPTER 11

Conclusion

Survey results indicate that many studies have been performed to apply IoT for smart agriculture, aiming to enhance productivity, reduce human labour, and improve production efficiency. In addition, we pointed out the challenges we need to overcome to be able to accelerate the deployment of IoT in smart agriculture. However, there are still some challenges that need to be addressed for IoT solutions to be affordable for the majority of farmers, including small- and medium-scale farm owners. In addition, security technologies need to be continuously improved, but in our opinion, the application of IoT solutions for smart agriculture is inevitable and will enhance productivity, provide clean and green foods, support food traceability, reduce human labour, and improve production efficiency.

Future Scope:

Monitoring of climate conditions:

Probably the most popular smart agriculture gadgets are weather stations, combining various smart farming sensors. Located across the field, they collect various data from the environment and send it to the cloud. The provided measurements can be used to map the climate conditions, choose the appropriate crops, and take the required measures to improve their capacity.

Greenhouse automation:

Typically, farmers use manual intervention to control the greenhouse environment. The use of IoT sensors enables them to get accurate real-time information on greenhouse conditions such as lighting, temperature, soil condition, and humidity.

In addition to sourcing environmental data, weather stations can automatically adjust the conditions to match the given parameters.

Crop management:

One more type of IoT product in agriculture and another element of precision farming are crop management devices. Just like weather stations, they should be placed in the field to collect data specific to crop farming; from temperature and precipitation to leaf water potential and overall crop health.

Thus, you can monitor your crop growth and any anomalies to effectively prevent any diseases or infestations that can harm your yield.

Precision farming:

Also known as precision agriculture, precision farming is all about efficiency and making accurate data-driven decisions. It’s also one of the most widespread and effective applications of IoT in agriculture.

By using IoT sensors, farmers can collect a vast array of metrics on every facet of the field microclimate and ecosystem: lighting, temperature, soil condition, humidity, CO_2 levels, and pest infections. This data enables farmers to estimate optimal amounts of water, fertilizers, and pesticides that their crops need, reduce expenses, and raise better and healthier crops.

Drones in Agriculture:

In precision agriculture, drones have a range of uses from soil and crop field analysis to planting and pesticide spraying. Drones can be used with different imaging technologies like hyperspectral, multispectral, thermal, etc. that can provide the farmers with time and site-specific information regarding crop health, fungal infections, growth bottlenecks, etc.

Drones can also identify drier regions in a field and measures can then be taken to irrigate such regions with better techniques. Precision agriculture provides farmers with such concrete information that enables them to make informed decisions and utilize their resources more efficiently.

Appendix:

// Sprint 1: For Reference

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

# IBM
organization = "“jztdcw”"
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"
# Gpio
def mycommandCallback(cmd):
    print("Command Received: %s" % cmd.data["command"])
    status = cmd.data["command"]
    if status == "lighton":
        print("LED is ON")
    elif status == "lightoff":
        print("LED 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()
# CONNECCT
deviceCli.connect()
while True:
    temp = random.randint(0, 100)
    hum = random.randint(0, 100)
    data = {"temp": temp, "hum": hum}
```

```
    def myOnPublishCallback():
        print(
            "Published Temperature = %s C" % temp,
            "Humidity = %s %%" % hum,
            "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
deviceCli.disconnect()
```

// Sprint 3: For Reference

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device
#Provide your IBM Watson Device Credentials
organization = "jztdcw"
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"
def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("Turn Motor ON")
    elif cmd.data['command']=='motoroff':
        print("Turn Motor OFF")
    elif cmd.data['command']=='lighton':
        print("Turn Light ON")
    elif cmd.data['command']=='lightoff':
        print("Turn Light OFF")

    if cmd.data['command']=='ACTIVATE IRRIGATION':
        print("TurnON")
    elif cmd.data['command']=='DEACTIVATE IRRIGATION':
```

```
        print("TurnOFF")
    elif cmd.data['command']=='HIGH TEMPERATURE':
        print("TurnON")
    elif cmd.data['command']=='LOW TEMPERATURE':
        print("TurnOFF")

    if cmd.data['command']=='BAD WEATHER':
        print("TurnON")
    elif cmd.data['command']=='GOOD WEATHER':
        print("TurnOFF")
    elif cmd.data['command']=='HUMIDITY HIGH':
        print("TurnON")
    elif cmd.data['command']=='HUMIDITY LOW':
        print("TurnOFF")

    if cmd.command == "setInterval":
        if 'interval' not in cmd.data:
        print("Error - command is missing required information: 'interval'")
    else:
        interval = cmd.data['interval']
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information:
'message'")
        else:
            output=cmd.data['message']
    print(output)
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:
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
```

```
deviceCli.disconnect()
```

// Code For Temperature

```
from random import *
from random import *
while True:
    temperature = randrange(0,100)
    humidity = randrange(0,100)
    if (temperature>=50):
        print("Alarm")
    else:
        print("No Alarm")
```

// Python Code

```
import random as rand
print("WELCOME SMART FARMER")
temperature = float(rand.uniform(15, 50))
if temperature > 22 and temperature < 40:
    humidity = int(rand.randint(45, 65))
elif temperature < 22:
    humidity = int(rand.randint(60, 70))
elif temperature > 40:
    humidity = int(rand.randint(25, 35))
moisture = int(rand.randint(00, 70))
print(
    "temperature:",
    temperature,
    "C",
    "\n",
    "humidity:",
    humidity,
    "\n",
    "moisture:",
    moisture,
)
if temperature > 35 or moisture < 20:
    print("Irrigation required")
    print("Activate irrigation ?")
    decision = input()
    if decision == "yes":
        print("Irrigation activated")
```

```
    else:
        print("Irrigation not activated")
else:
    print("Irrigation not required")
```

// Sprint 4: For Reference

```
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "jztdcw"
deviceType = "NodeMCU"
deviceId = " node-mcu-4321 "
authMethod = "use-token-auth"
authToken = "987654321"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %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 an event of type "greeting" 10 times deviceCli.connect()

while True:
    #Get Sensor Data from DHT11
```

```
    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()
```

GitHub Link and Project Demo Link:
https://github.com/IBM-EPBL/IBM-Project-18928-1659691377