IBM - NALAIYATHIRAN

THIAGARAJAR COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SMART FARMER – IOT ENABLED SMART FARMING APPLICATION

TEAM-ID: PNT2022TMID21246

TEAM DETAILS

TEAM-ID:
PNT2022TMID21246

TEAM MEMBERS

REG. NO.	IBM REG. NO.	NAME
19C001	917719C001	ABIRAMIPRIYA J
19C040	917719C040	KALAISELVI S
19C042	917719C042	KAVIYA D
19C048	917719C048	LAVANYA R

ABSTRACT

The history of agriculture began thousands of years ago. After gathering wild grains beginning at least 105,000 years ago, nascent farmers began to plant them around 11,500 years ago. Agriculture is the practice of cultivating plants and livestock. Agriculture was the key development in the rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that enabled people to live in cities. Modern agronomy, plant breeding, agrochemicals such as pesticides and fertilizers, and technological developments have sharply increased crop yields, but cause ecological and environmental damage. Agriculture is the backbone of Indian Economy. The improvement of agriculture sector provides enormous opportunities for common folks, business people. So, in order to improve the farming methods, we use technology (Iot applications) rather than modern technology. This helps in providing precise condition of the farms which include temperature, humidity, range, soil conditions. This information when known to famers beforehand or to owners in their remote location helps them to improve the field and the farming conditions. The improved farming method thus result in increase in output

Farming is a complex, unpredictable and individual business.

Content

1. INTRODUCTION

Fig.1.1. IoT based smart farming

1.1 Project Overview

The history of agriculture began thousands of years ago. After gathering wild grains beginning at least 105,000 years ago, nascent farmers began to plant them around 11,500 years ago. Agriculture is the practice of cultivating plants and livestock. Agriculture was the key development in the rise of sedentary human civilization, whereby farming of domesticated species created food surpluses that enabled people to live in cities.

Modern agronomy, plant breeding, agrochemicals such as pesticides and fertilizers, and technological developments have sharply increased crop yields, but cause ecological and environmental damage.

1.2 Purpose

Agriculture is the practice of cultivating natural resources to sustain human life and provide economic gain. It combines the creativity, imagination, and skill involved in planting crops and raising animals with modern production methods and new technologies.

Agriculture is also a business that provides the global economy with commodities: basic goods used in commerce, such as grain, livestock, dairy, fibre, and raw materials for fuel. Even though many modern approaches take place to make an effective agriculture practice. There are still problems that are faced by farmers in their daily life such as, unpredictable climate changes, adapting new technology, and monitoring each work about their agricultural land. Farming is a complex, unpredictable and individual business.

2. LITERATURE SURVEY

2.1 Existing problem

S.NO.	PAPER TITLE	AUTHOR NAME	PUBLICATION YEAR	RESULTS
1.	Machine learning applications	MWP Maduranga, Ruvan Abeysekera		To improve productivity of agriculture through intelligent farm management, the data analyzing must

	in IoT based agriculture and smart farming		2020	be well analyzed and processed. High-performance computing capability in ML opens up new opportunities for data-intensive science as the amount of data collected increases; In this article we review existing approaches have been made to the smart agriculture and farming based on IoT and ML separately. Also, we propose novel concepts that how can ML-IoT can be blended in such applications.
2.	A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming	Muhammad Shoaib Farooq, Shamyla Riaz, Adnan Abid, Kamran Abid, Muhammad Azhar Naeem	2019	A sensor performs multiple tasks like soil sensing, temperature sensing, weather sensing, light sensing, and moisture sensing. Similarly, devices perform many control functions like, node discovery, device identification and naming services etc. All these functions are performed by any device or sensor which is controlled through a microcontroller.
3.	Smart farming: IoT based smart sensors agriculture stick for	Anand Nayyar, Vikram Puri	2016	The aim/objective of this paper is to propose a Novel Smart IoT based Agriculture Stick assisting farmers in getting Live Data (Temperature, Soil Moisture) for efficient environment monitoring which will enable them to do smart farming and increase their

	live temperature and moisture monitoring using Arduino, cloud computing & solar technology			overall yield and quality of products. The product being proposed is tested on Live Agriculture Fields giving high accuracy over 98% in data feeds.
4.	IoT based intelligent irrigation support system for smart farming applications	Neha Kailash Nawandar, Vishal Satpute	2019	A crop irrigation management system with sensor data fetch, transfer and operate functionalities is proposed to meet the expectations. The system comprises of sensing, data processing and actuator sections, with a network of ambient temperature and humidity at a height and soil moisture sensor placed at the root zone of the subject. Results show that there is tolerable error in the reconstructed data and 62.5% and 67.5% compression is achieved for ambient temperature, humidity and soil moisture respectively.
5.	IOT based smart farming	C Mageshkumar, KR Sugunamuki	2020	This project includes the various features like soil moisture sensor, temperature sensor, humidity sensor

				for facilitate the irrigation in proper way. Various sensor nodes are deployed at different locations in the farm to automate the irrigation anytime anywhere. This project will be more helpful for the farmer's welfare.
6.	Survey, comparison and research challenges of IoT application protocols for smart farming	Dimitrios Glaroudis, Athanasios Iossifides, Periklis Chatzimisios	2020	The unprecedented capability of data collection and management offered by IoT is based on several factors of the underlying communication network architecture and technology, one of the most important being the application-level protocol that is used among IoT nodes, gateways, and application servers. Furthermore, it provides a comparison among them, in terms of well-accepted key performance indicators and comments on their suitability in the framework of smart farming as well as the corresponding challenges that must be faced towards their efficient implementation.

Table 2.1. Existing solutions

2.2 References

- https://ieeexplore.ieee.org/Xplore/home.jsp
- https://link.springer.com/

2.3 Problem Statement Definition

Our problem statement is to reduce manual works of the farmer by monitoring and controlling the agricultural lands.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

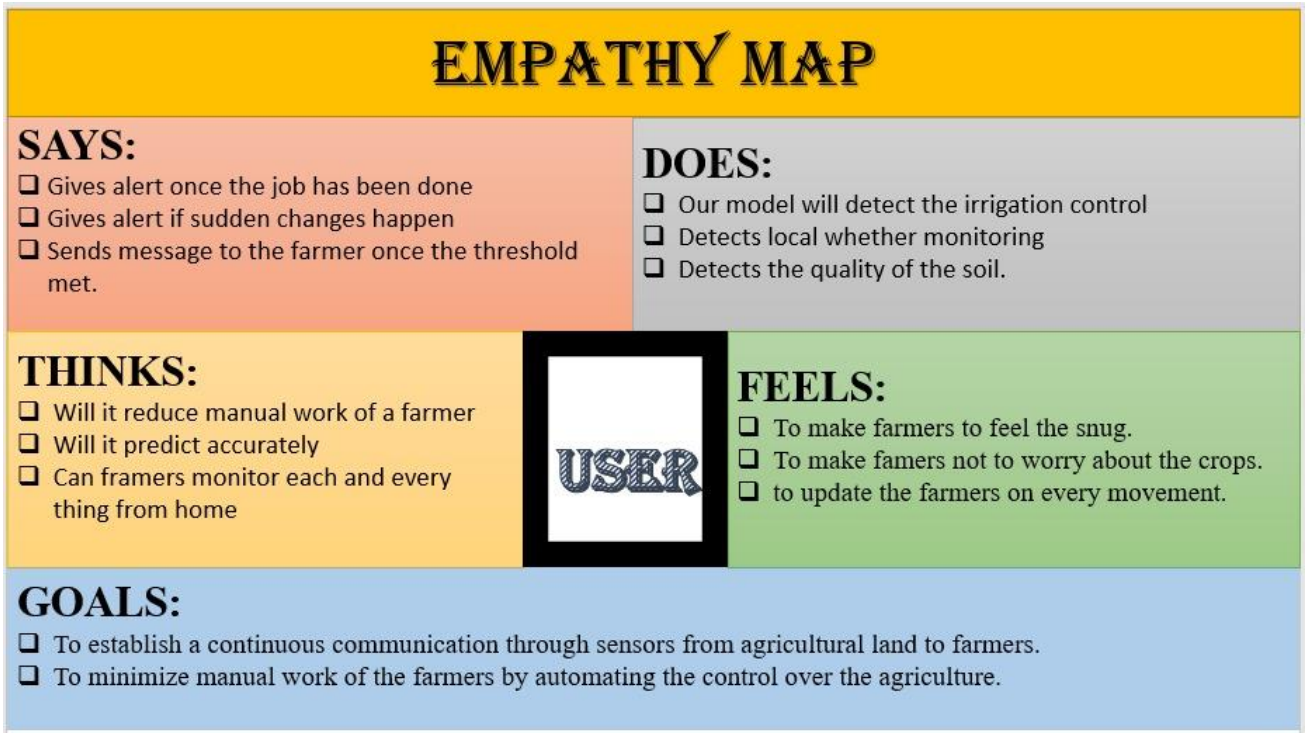


Fig 3.1. Empathy map

3.2 Ideation & Brainstorming

Step-1: Team Gathering, Collaboration and Selectthe Problem Statement

We have followed the first step of brainstorming; we have discussed as a team todecide a problem statement.

Fig 3.2. Brainstorming

As per the guideline the following is done

- Team gathering
- Collaboration
- Deciding the problem statement

Step-2: Brainstorm, Idea Listing and Grouping

TEAM MEMBER	MEMBERS OPNION	IDEAS

KALAI SELVI S	MAY GIVE TRY	Collecting various sensor valuesand reporting to the farmer through website
LAVANYA R	GOOD	Collecting values from sensors and gives alarm through buzzerswhen the threshold is met.
KAVIYA D	BEST	Collecting various sensor values and reporting to the farmer through application
ABIRAMIPRIYA J	COSTLY	Automatic irrigation control (on and off) and other actions should be done automatically.

Table 3.1. Ideas of all the teammates

By grouping ideas,

- We have planned to use moisture sensor, humidity sensor for detection.
- The values when surpasses the threshold value will send a notification to farmer.
- The notification is also sent in case of emergencies.

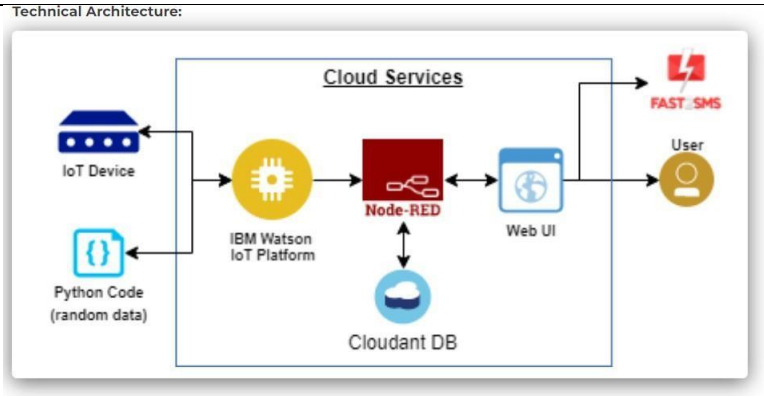
Step-3: Idea Prioritization:

Fig 3.3. Prioritization matrix

1) Using sensors like humidity, moisture sensor in order to detect the nature of crops.
2) The sensors are connected to IOT application in order notify the farmers.
3) Use other sensors if required for the crops, like tea plantations.
4) Using automation without the knowledge of famers or human resource.

3.3 Proposed Solution

S.NO.	PARAMETER	DESCRIPTION
1.	**Problem Statement**	Designing an IoT enabled smart farming application that is used for automating agriculture and sending notifications to farmers when the threshold level is met.
2.	**Idea / Solution description**	Our solution for smart farming system includes • Our system includes a humidity sensor and moisture sensor to detect the current soil conditions and to know whether there are any deviations from the pre-determined threshold. • If a particular threshold is met in the water level, then the servo motor is automatically ON or OFF. • Notifications regarding the soil conditions and the moisture level will be sent to the farmer using blink app.
3.	**Novelty / Uniqueness**	**Preceding system's objectives:** ➢ In the prior systems, the main objective is to switch on or off the servo motor based on the pre-determined threshold. ➢ The farmer will not be able to know about what is happening in the field. **Proposed system's objective:** ➢ We will be designing a system in which the notifications regarding the water level and the soil conditions will be sent to the farmer directly using blink app.

4.	**Social Impact / Customer Satisfaction**	**Social Impact:** ➢ If there is no irrigation control, then the amount of water usage will increase and the water will be wasted for unnecessary purposes. Some crops will need less amount of water for their growth. In such cases if the irrigation control is not in the hands of the farmer, then the health of the crops will be spoilt. ➢ If the soil is not monitored continuously, the farmer will not get profit in the yield. **Customer Satisfaction:** The main objectives of this proposed solution are ❖ **User-friendly** – Easy access for the farmer with the help of notifications. ❖ **Flexible** – Check status of the water level and soil conditions under all the conditions.
5.	**Business Model (Revenue Model)**	 Fig 3.4. Technical architecture

		DESCRIPTION OF THE TECHNICAL ARCHITECTURE: • Sensors values can be viewed by the farmers using their mobile phones. • Notifies the farmer when the random values cross the threshold value.
6.	**Scalability of the Solution**	❖ The solution can be easily expandable to larger fields by deploying sensors wherever needed. The values would be read and based on the threshold, warnings and notifications could be given to the farmer. ❖ By using GSM modules, the farmer gets notifications quickly and the automation can also be done without any manual interruption.

Table 3.2. Proposed solution

3.4 Problem Solution fit

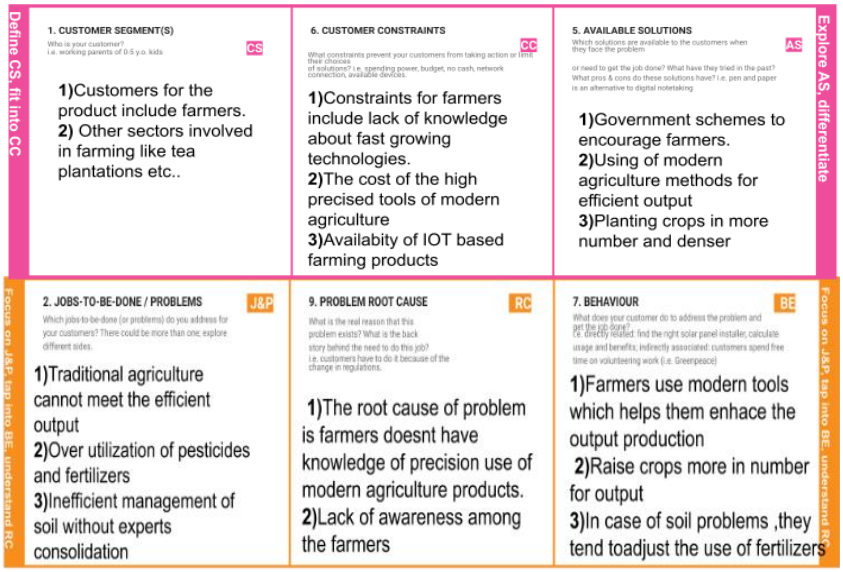


Fig 3.5. Problem fit

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)
FR – 1	Timely Updated with proper network connection	The notifications shown to the farmers with respect to the information gathered from various sensors must be an updated version with respect to time. The farmers will get information simultaneously after getting updated.
FR – 2	User Understandable	The information displayed to the farmers must be understandable in terms of the words that are being used.
FR – 3	Distinct visibility	The information should be quite large for the users to read clearly. The sensor values and the words used must be of appropriate size, so that it will be clear to the farmers.

Table 4.1. Functional requirements

4.2 Non-Functional requirements

Following is the Non-Functional Requirements of the proposed solution

FR No.	Non-Functional Requirements	Description
NFR – 1	**Reliability**	The notifications of the sensor values should display correct and errorless information.

NFR – 2	**Security**	The information should be made secure so that it could not be manipulated by any other person.
NFR – 3	**Usability**	The information must be able to update whenever andwherever required
NFR – 4	**Performance**	The system should be able to update itself properly attimes of fatal conditions and provide necessary driving measures.
NFR – 5	**Availability**	The system must be made available 24/7 in order to provide uninterrupted service to the farmer.
NFR – 6	**Scalability**	The system should be compatible with the developed specifications and be open for future upgradations

Table 4.2. Non – functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

Data flow process:

- The user logs in into the application using credentials used while registering, choose indashboard to access information about crops or experts' advice.
- The sensors placed in the soil send its data about the condition of the soil.
- Sensors send information about condition of humidity in air.
- Sensors send information about the condition of co2 level etc.
- The threshold value is present for the condition of crop, soils etc.

- The decision is on whether to display daily updates or alert as notification depends on the condition of thefarm.
- Notification is sent on both the occasions depending on the situation.
- If the condition of the farm far exceeds the threshold value, it alerts the user to take action.
- If not, the daily updates are displaced along with the suggestions if necessary.

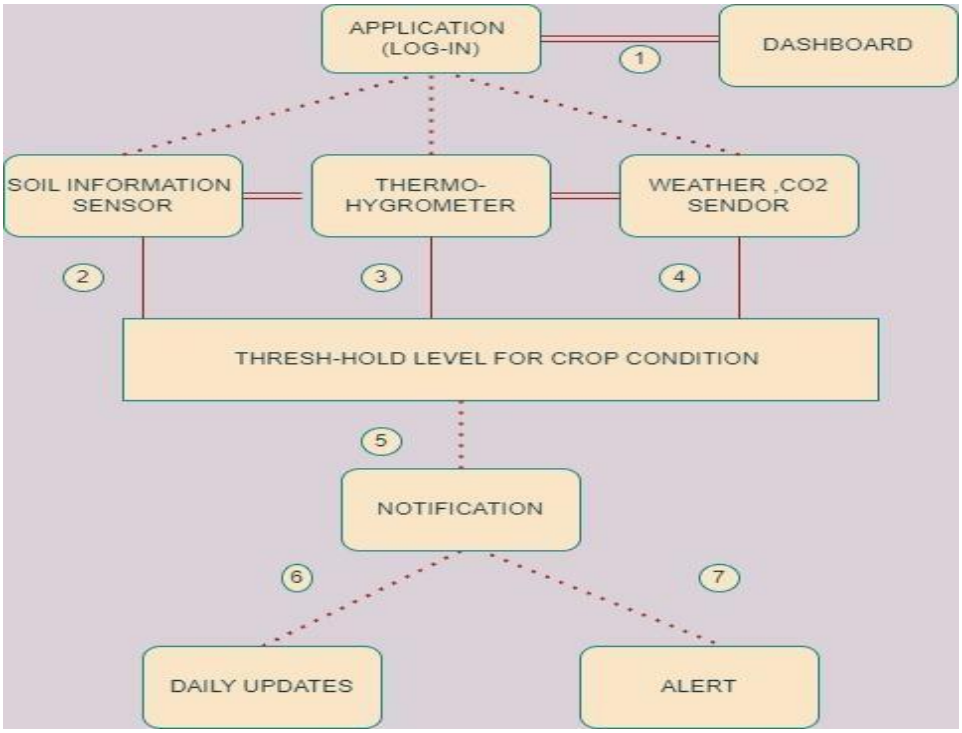


Fig 5.1. Dataflow diagram

5.2 Solution & Technical Architecture

Solution Architecture:

- The smart fire management system includes a Gas sensor, Flamesensor and temperature sensors to detect any changes in the environment.

- Based on the temperature readings and if any Gases are present the exhaust fans are powered ON.
- If any flame is detected the sprinklers will be switched on automatically.
- Emergency alerts are notified to the authorities and Fire station.

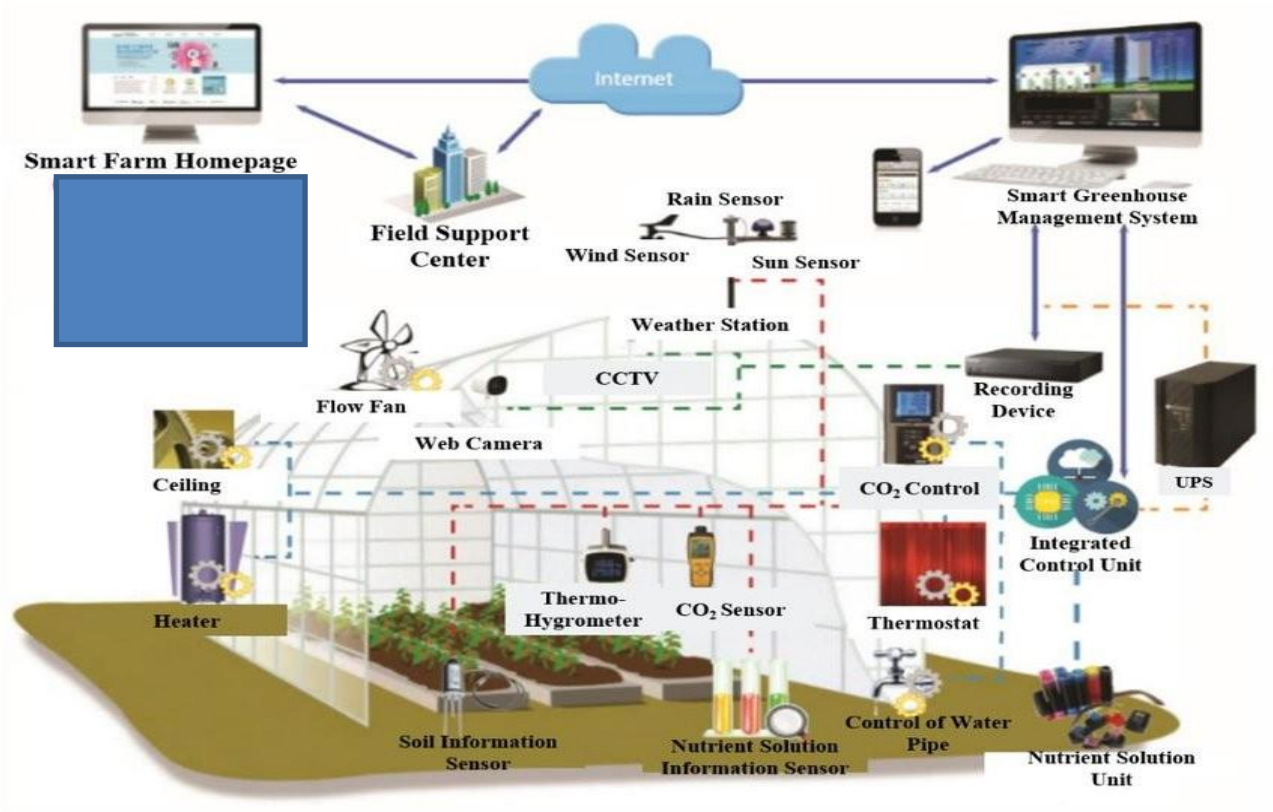


Fig 5.2. Solution architecture

Technical stack:

Components	Description	Technology
User interface	User interface with the equipment using mobile app, web UI	HTML, CSS, JavaScript /AngularJS/ React JS
Appl cat on logic 1	Logic for detecting the status of the sensors	python

Appl cat on logic 2	Logic for detecting the level of water	Arduino IDE
Appl cat on logic 3	Logic for controlling process of irrigation system	Raspberry pi
Cloud data base	Database Service on Cloud	IBM DB2, IBM Cloudant, IBM Watson, Node red service
File storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem

Table 5.1. Components and technologies

Characteristics	Description	Technology
Throughout	High Efficiency is achieved	Using IOT
Scalability	To accommodate future changes in use and occupancy	Update can be made easily using mobile app

Table 5.2. Application Characteristics

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Farmer)	Registration	USN-1	As a user, I can register for the	I can access my account / dashboard	High	Sprint-1

			application by entering my email, password, and confirming my password.			
		USN-2	As a user, I can register the type of crops in the farm	I can get necessary details about crop	Medium	Sprint-2
		USN-3	As a user, I can obtain real time updates of the crops and soil	I can know about constraints in ph of soil and health of crop	High	Sprint-1
		USN-4	As a user, I can receive emergency notification in case of emergency	I can act efficiently in case of emergency	High	Sprint-1
		USN-5	As a user, I can receive experts advise in time of need	Efficient output of crop	High	Sprint-1
	Login	USN-6	As a user, I can use the applicationby entering mail and password	I can log into application	High	Spirint-2
	Dashboard	USN-6	As a user, I can choose among the experts for discussion and information about crops registered	I can select among the given expert and crops	Medium	Spirint-2

Customer (Application user)	Generating Data	USN-7	As a user, I am able to get information about the crops and get experts advice regarding the crops condition	Better information about crop and experts from application	High	Spirint-1
	Notification	USN-8	As a user,I can get notification updates of crops condition and notification incase of emergency	I can receive notification	High	Spirint-1
Customer Care Executive	Efficient care of crops	USN-9	As an executive, I can help the farmers use the application to nourish the crops better	Efficient crop output	High	Spirint-2
Administrator	Administering the cropdata	USN-10	As an admin, I can get through the interface and administer thedata functionality	Easy Administration of data when data is updated	High	Spirint-2
	Notification	USN-11	As an admin, send timely updates of crops and alert in case of notification	Timely notifications	High	Spirint-2

Table 5.3. User stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	5	High
Sprint-1		USN-3	As a user, I can register for the application through Gmail	5	Medium
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	5	High
Sprint-2	Dashboard	USN-5	As a user, I can see the status of the temperature in the dashboard.		High

Sprint-2	Dashboard User Interface	USN - 11	Administrator designing the user interface	10	Medium
Sprint-3		USN-6	As a user I can see the status of the water level in the irrigation system.	10	High
Sprint-3		USN-7	As a user, I can log out my account in settings.	10	Medium
Sprint-4		USN-8	As a user, I can see my daily updates in account settings.	10	Medium
Sprint-4	Mobile application / web application	USN-9	Solve issues brought up by client	5	Medium
Sprint-4		USN-10	Roll out updates and bug fixes	5	High

Table 6.1. Estimation chart

Project Tracker, Velocity & Burndown Chart:

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	29 Oct 2022

Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Table 6.2. Burndown chart

6.2 Sprint Delivery Schedule

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on theselected project & gathering information byreferring the, technical papers, research publications etc.	19 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the userPains & Gains, Prepare listof problem statements	19 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session andprioritize the top 3 ideas based on the feasibility & importance.	19 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solutiondocument, which includes the novelty, feasibility of idea, business model, social impact,scalability of solution, etc.	19 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fitdocument.	19 SEPTEMBER 2022

Solution Architecture	Prepare solution architecturedocument.	19 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	3 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	3 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	3 OCTOBER 2022
Technology Stack	Prepare the technology architecture diagram.	3 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	1 NOVEMBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	19 NOVEMBER 2022

Table 6.3. Sprint delivery schedule

6.3 Reports from JIRA

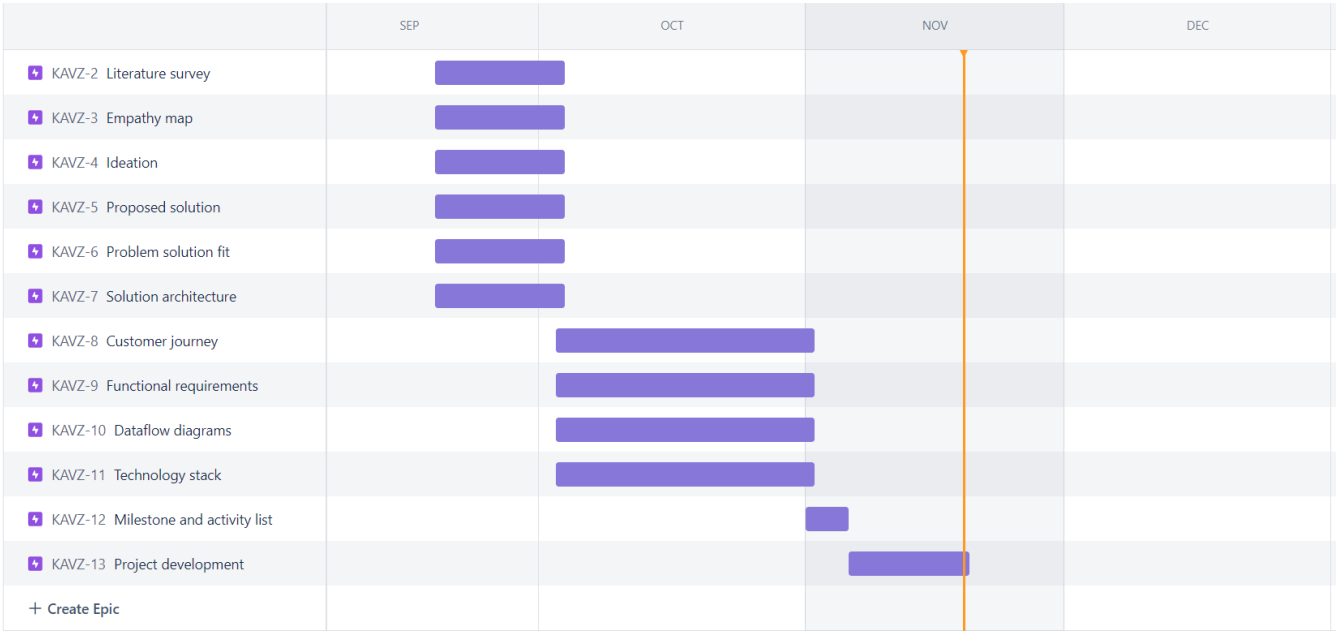


Fig 6.1. JIRA report

7.CODING AND SOLUTION:

The python code used in the project is as follows:

```
import time
import random
#import ibmiotf.application
import ibmiotf.device
import sys
config={
    "org":"5dxn3m",
    "type" :"abcd",
    "id":"123",
```

```
    "auth-method":"token",
    "auth-token":"123456789"
}
client= ibmiotf.device.Client (config)
client.connect()

def myCommandCallback (cmd):
    a=cmd.data
    if len(a["command"])==0:
        pass
    else:
        print(a["command"])
def pub (data):
    client.publishEvent (event="status", msgFormat="json",data=data, qos=0)
    print("Published data Successfully: %s",data)
while True:
    s=random.randint(0,100)
    h=random.randint(0,100)
    t=random.randint(0,100)
    data={"sm":s,"hum":h,"temp":t}
    pub(data)
    time.sleep(5)
client.commandCallback = myCommandCallback
client.disconnect()
```

ON TESTING RESULTS:

Connected successfully: d:5dxn3m:abcd:123

Published data Successfully: %s {'sm': 46, 'hum': 13, 'temp': 92}

Published data Successfully: %s {'sm': 2, 'hum': 17, 'temp': 52}

Published data Successfully: %s {'sm': 98, 'hum': 3, 'temp': 54}

Published data Successfully: %s {'sm': 86, 'hum': 100, 'temp': 17}

Published data Successfully: %s {'sm': 40, 'hum': 37, 'temp': 85}

Published data Successfully: %s {'sm': 92, 'hum': 54, 'temp': 96}

Published data Successfully: %s {'sm': 92, 'hum': 92, 'temp': 80}

Published data Successfully: %s {'sm': 66, 'hum': 31, 'temp': 67}

Published data Successfully: %s {'sm': 57, 'hum': 0, 'temp': 43}

Published data Successfully: %s {'sm': 69, 'hum': 1, 'temp': 48}

Published data Successfully: %s {'sm': 45, 'hum': 56, 'temp': 25}

Published data Successfully: %s {'sm': 41, 'hum': 40, 'temp': 31}

Published data Successfully: %s {'sm': 94, 'hum': 77, 'temp': 69}

Published data Successfully: %s {'sm': 80, 'hum': 65, 'temp': 59}

Published data Successfully: %s {'sm': 57, 'hum': 78, 'temp': 22}

Published data Successfully: %s {'sm': 31, 'hum': 97, 'temp': 26}

Published data Successfully: %s {'sm': 80, 'hum': 57, 'temp': 54}

Published data Successfully: %s {'sm': 65, 'hum': 49, 'temp': 22}

Published data Successfully: %s {'sm': 71, 'hum': 32, 'temp': 81}

Published data Successfully: %s {'sm': 83, 'hum': 53, 'temp': 44}

Published data Successfully: %s {'sm': 55, 'hum': 25, 'temp': 32}

Published data Successfully: %s {'sm': 0, 'hum': 76, 'temp': 55}

Published data Successfully: %s {'sm': 27, 'hum': 65, 'temp': 3}

Published data Successfully: %s {'sm': 30, 'hum': 76, 'temp': 94}

Published data Successfully: %s {'sm': 35, 'hum': 89, 'temp': 3}

Published data Successfully: %s {'sm': 87, 'hum': 97, 'temp': 54}

Published data Successfully: %s {'sm': 73, 'hum': 71, 'temp': 41}

Published data Successfully: %s {'sm': 66, 'hum': 95, 'temp': 100}

Published data Successfully: %s {'sm': 16, 'hum': 70, 'temp': 72}

Published data Successfully: %s {'sm': 34, 'hum': 32, 'temp': 94}

Published data Successfully: %s {'sm': 46, 'hum': 81, 'temp': 28}

Published data Successfully: %s {'sm': 52, 'hum': 80, 'temp': 29}

FEATURES:

FEATURE 1: The first feature is the use of web application inoder to help farmer to have remote control over the field.

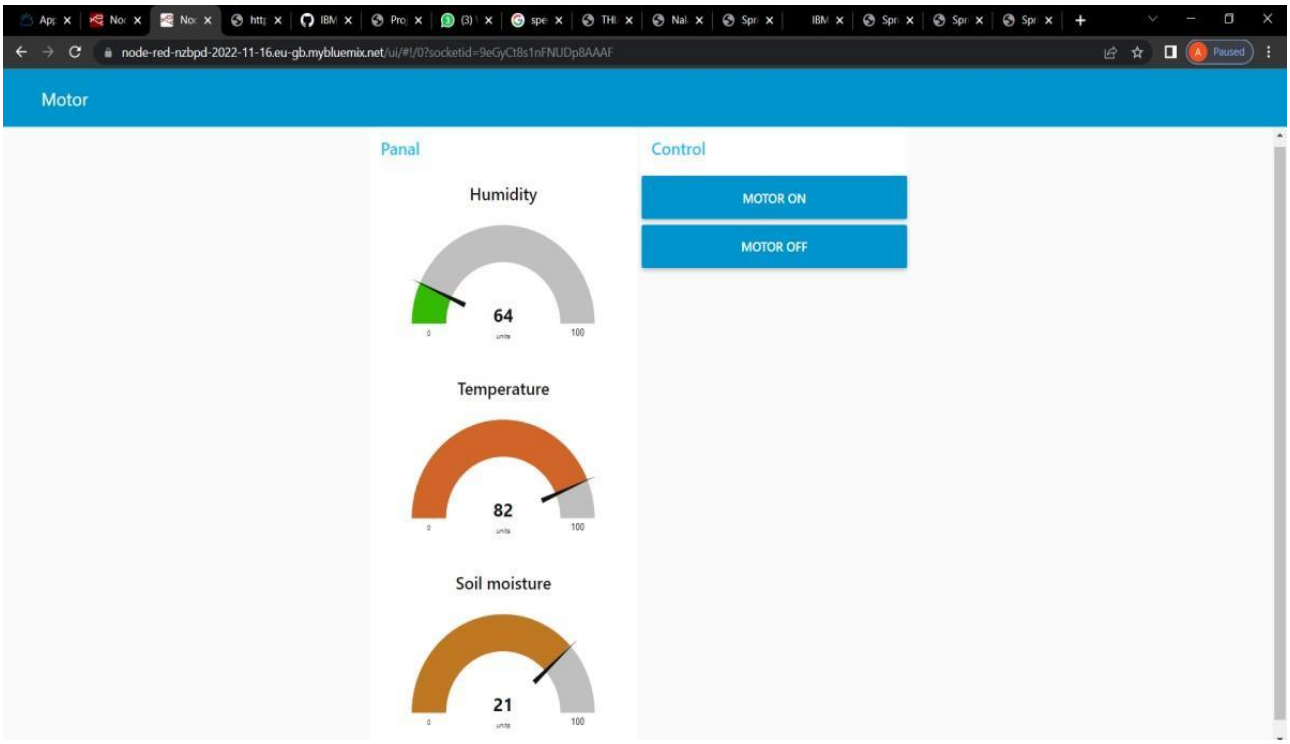


Fig. 7.1. Visual results in node red dashboard

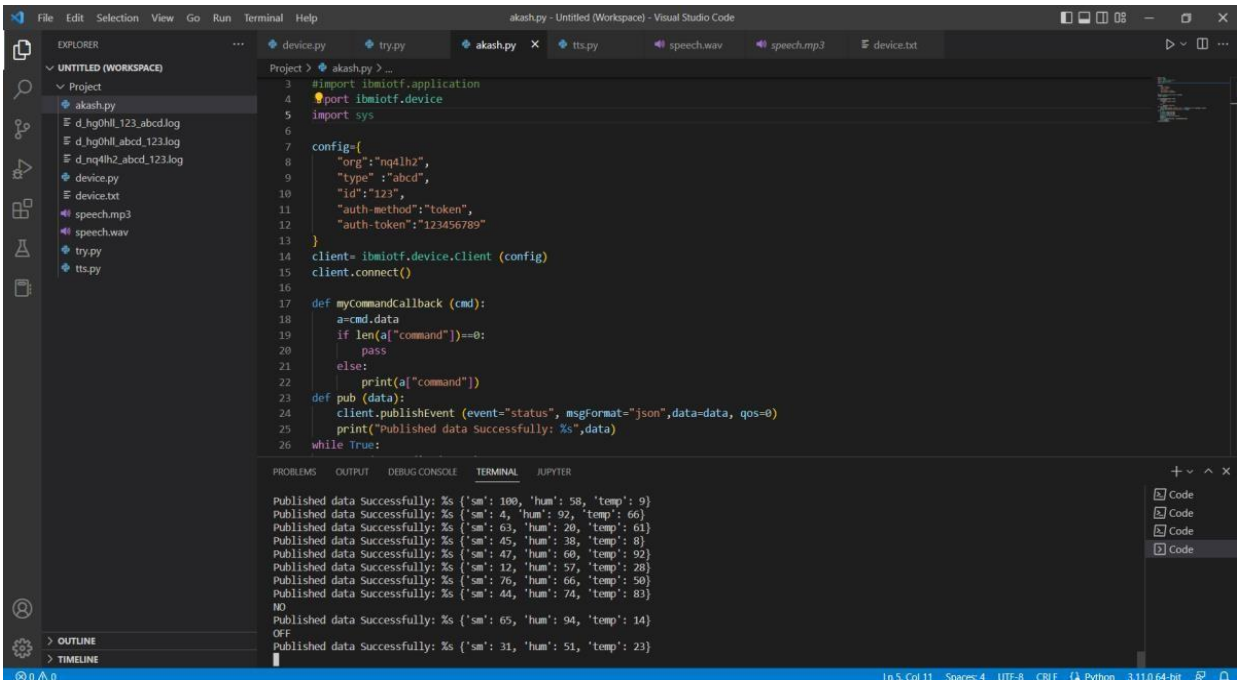


Fig. 7.2. Coding results

FEATURE 2:

The other feature is to have the sensor information available anywhere when the farmer wishes to view it. (i.e) MOBILE APP

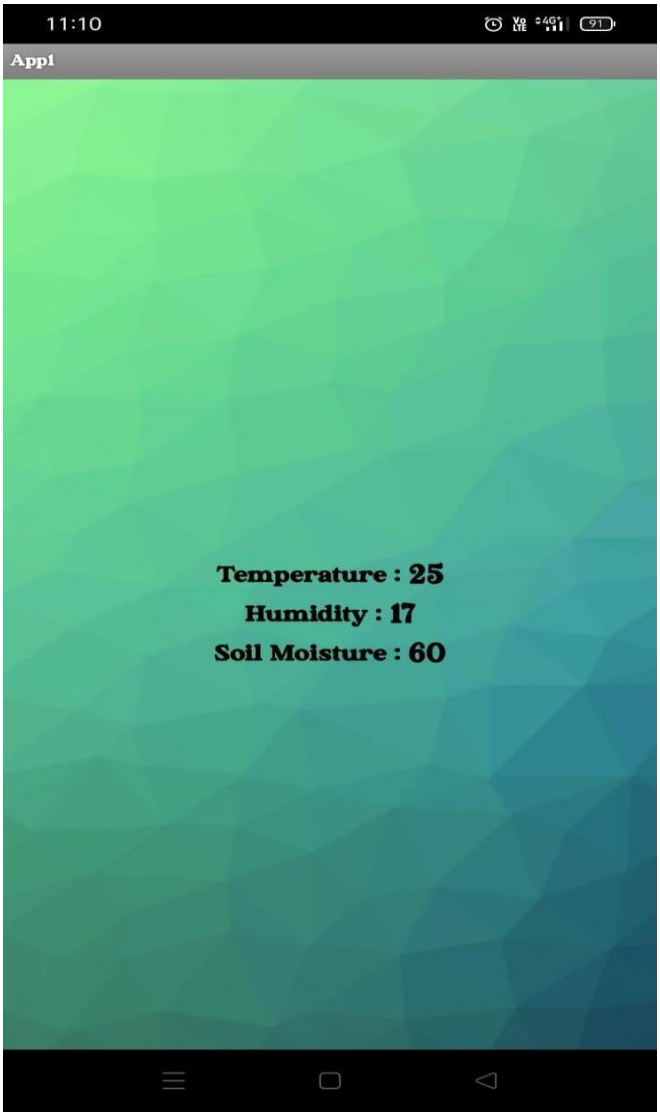


Fig. 7.3. Results in MIT App

8.TESTING AND USE CASES:

The python code generate the following ,when executed thus resulting as follows

TEST CASES:

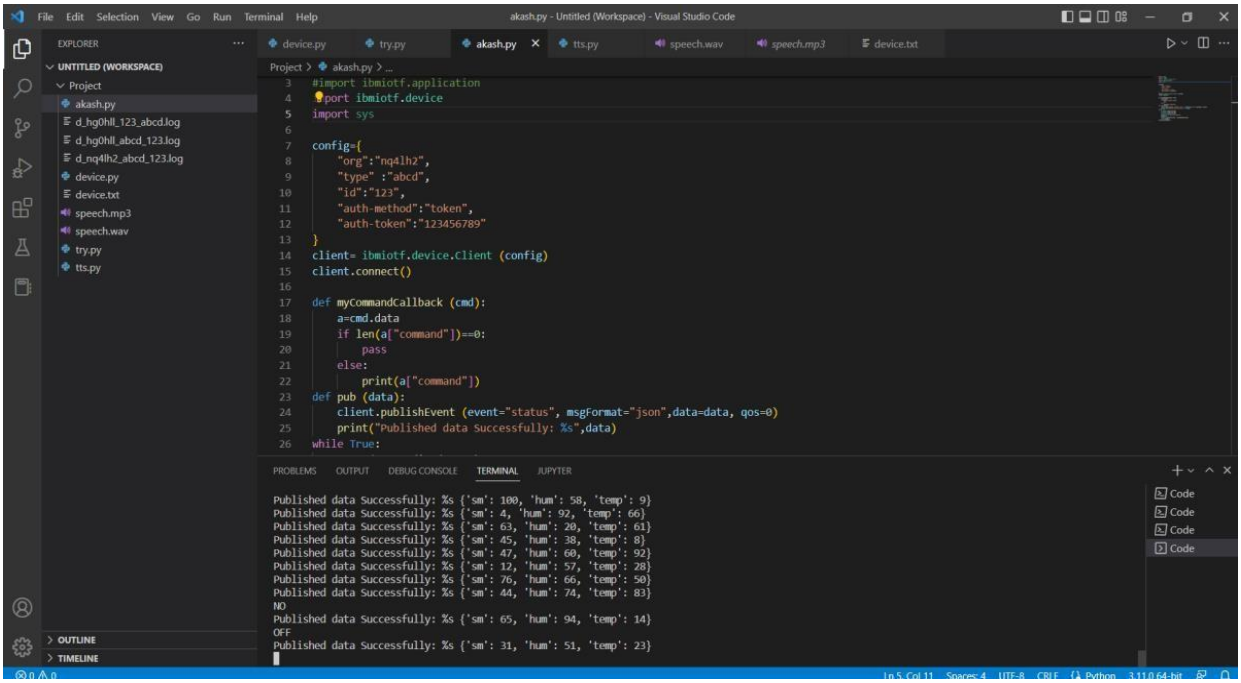


Fig 8.1. Python code

MOTOR OFF: Published data Successfully: %s {'sm': 90, 'hum': 41, 'temp:26}

Published data Successfully: %s {'sm': 18, 'hum': 4, 'temp': 86}

MOTOR ON: Published data Successfully: %s {'sm': 17, 'hum': 5, 'temp': 50}

Published data Successfully: %s {'sm': 29, 'hum': 47, 'temp': 32}

USE CASES:

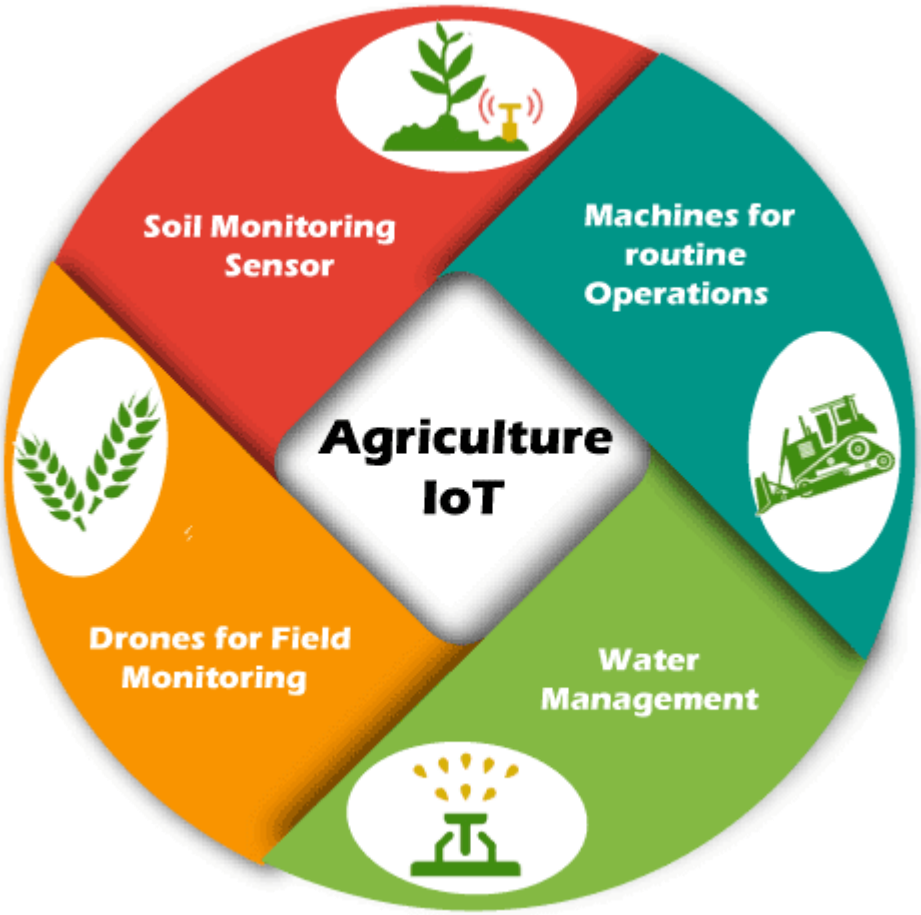


Fig 8.2. Use cases of smart farming

The use case of the device is for farmers. It provides immense help in the following ways:

1) Remote monitoring
2) Alert with precise timing (APP)
3) Make alterations based on the condition proposed by the sensors
4) Improve the efficiency of the farming
5) Cost effective
6) Increase in output
7) Reduction of labour

9.RESULT:

The end result of the problem is a SMART FARMING APPLICATION (IOT DEVICE),the application which has its connection with Iot sensors.

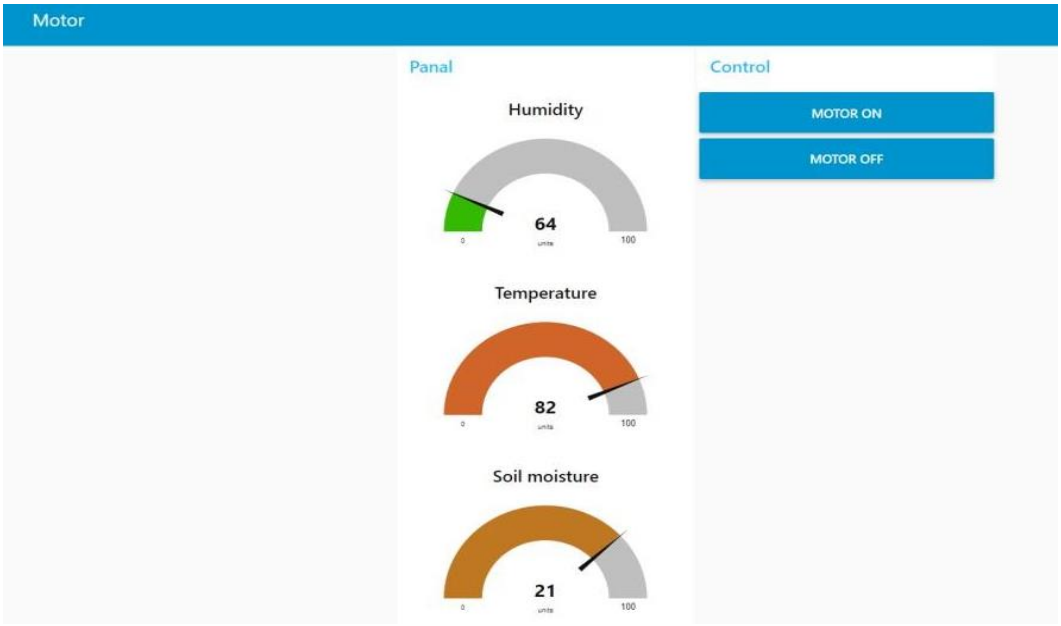


Fig 9.1. Results of smart farming application

The performance metric in order to measure the device performance is done by following:

1)SERVICE VALUE: The service provided by the device and the availability of the service. The service value of the end product is EXCELLENT as the app always shows the farmer, condition of the land and crops

2)ACTIVE USERS: The active users of the product are the farmers or owners who wish to have remote knowledge of the condition in field.

3)VALUE OFFERED: The value offered to the world is the product of the above two values for the device.

PERFROMANCE = SERVICE VALUE * ACTIVE USERS

Hence the device performance is measured by the service or amount of time the device is up and the active users (farmers or owners) using the app as well as web application.

10.ADVANTAGES AND DISADVANTAGES:

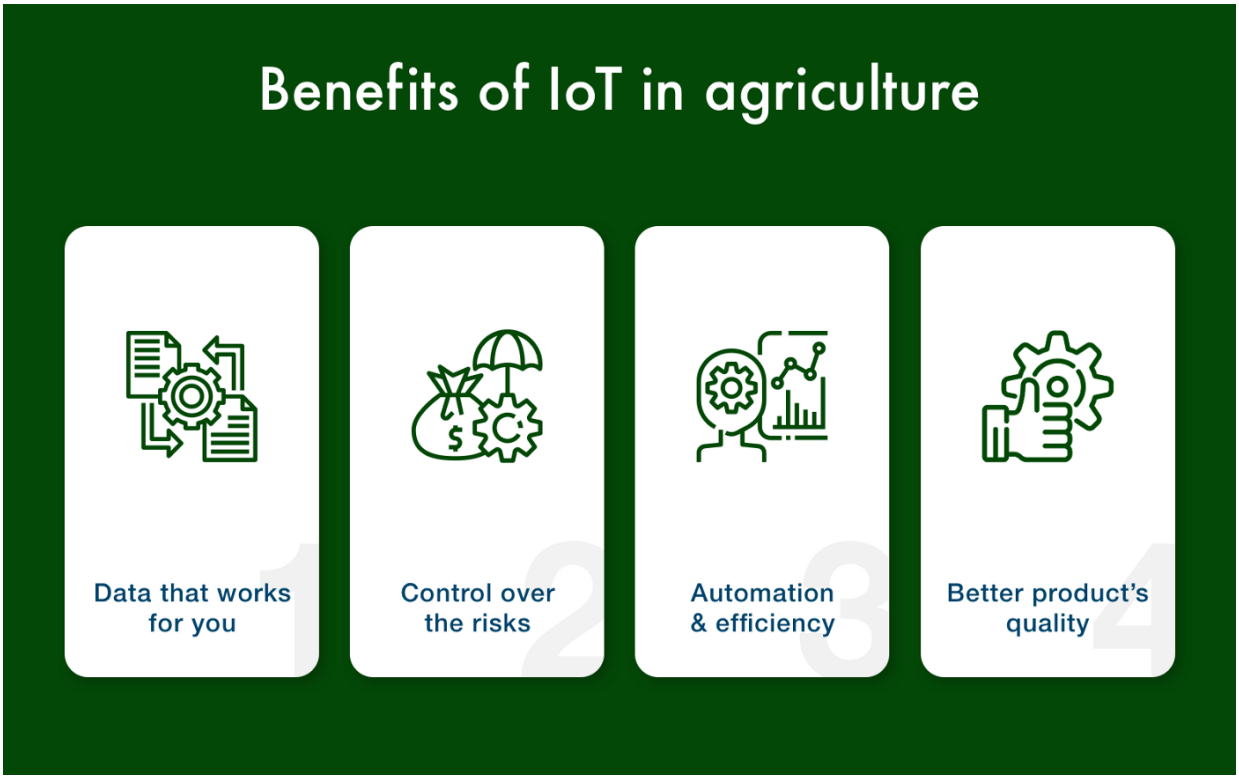


Fig. 10.1. Agriculture advantages

The main advantage of the product is its functionality able to solve the problem faced by the farmers.

1) The smart iot based farming application helps farmers to know about the precise condition in the farm

2) To be able to control the field remotely.

3) To make necessary further improvements in the field by knowing the precise condition incase of humidity,temperature etc..

4)To increase the output

Some disadvantages include:

1)Inorder to use to product farmers should have basic awareness about technology

2)Farmers awareness about device should to spread inorder to increase the active users

11.CONCLUSION:

The solution to reduce the labour work and to control some functionalities remotely, the best option available is to be able to use iot devices to control over the field.

The method used to solve the problem is to be able to have control through a web application connected through node red ,and have information access through the app.The sensors available in the field provide the information on field condition.Thus the farmers will be able to improve the efficiency of farming .In the owners point of view ,they might be able to know about the condition of thier land at any point of time.

12.FUTURE SCOPE:

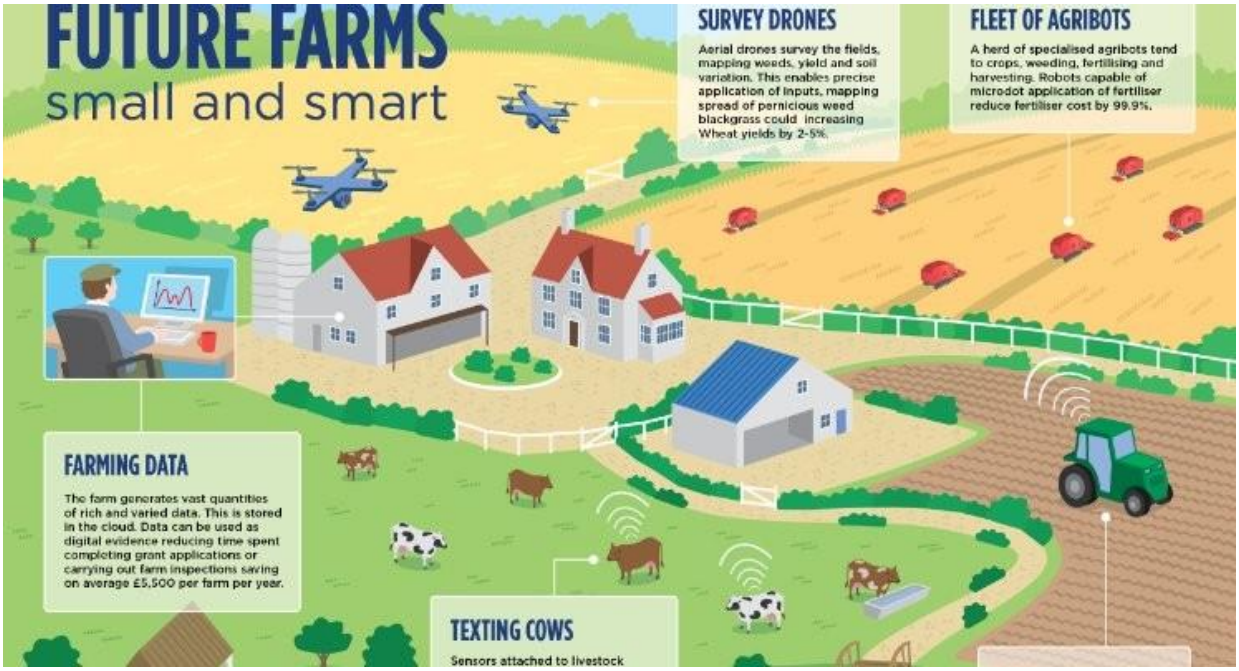


Fig. 12.1. Future scope of smart farming

The future scope of the product is to improve the information provided by the app. The improvement also includes the application which can be monitored or controlled remotely.

Other ways to improve the active users, which include the increase in number of active users, the service value of the device. It can be used to improve the business scope in the agriculture sector.

If the performance metrics is very high, then it will create a new market area which results in the increase of business opportunities in the IoT field.

13.APPENDIX:

SOURCE CODE:

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



config={
    "org":"5dxn3m",
    "type" :"abcd",
    "id":"123",
    "auth-method":"token",
    "auth-token":"123456789"
}
client= ibmiotf.device.Client (config)
client.connect()


def myCommandCallback (cmd):
    a=cmd.data
    if len(a["command"])==0:
        pass
    else:
        print(a["command"])
def pub (data):
    client.publishEvent (event="status", msgFormat="json",data=data,
qos=0)
    print("Published data Successfully: %s",data)
while True:
    s=random.randint(0,100)
    h=random.randint(0,100)
    t=random.randint(0,100)
    data={"sm":s,"hum":h,"temp":t}
    pub(data)
```

```
    time.sleep(5)
    client.commandCallback = myCommandCallback
client.disconnect()
```

PROJECT DEMO LINK:

https://drive.google.com/drive/folders/1-ZIt7EyEpPTw4u98vrlkN1Gjm4SqrF4n