

SMART FARMER - IoT BASED SMART FARMING APPLICATION

PROJECT REPORT

TEAM ID : PNT2022TMID31713

TEAM LEAD : SAMRITHA S

TEAM MEMBERS : 1. AAKASH J
2. DHEVAKI V
3. JANANI S
4. GOWTHAM S

CONTENTS

1. INTRODUCTION

- 1.1 Project Overview
- 1.2 Purpose

2.LITERATURE SURVEY

- 2.1 Existing Problem
- 2.2 References
- 2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- 3.1 Empathy map canvas
- 3.2 Ideation & Brainstorming
- 3.3 Proposed Solution
- 3.4 Problem Solution Fit

4.REQUIREMENT ANALYSIS

- 4.1 Functional Requirements
- 4.2 Non functional Requirements

5. PROJECT DESIGN

- 5.1 Data flow diagrams
- 5.2 Solution & Technical Architecture
- 5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

- 6.1 Sprint Planning & Estimation
- 6.2 Sprint Delivery Schedule
- 6.3 Reports from JIRA

7. CODING & SOLUTIONING

- 7.1 Feature 1
- 7.2 Feature 2
- 7.3 Database Scheme

8. TESTING

- 8.1 Test Cases
- 8.2 User Acceptance Testing

9. RESULTS

- 9.1 Performance Metrics

10.ADVANTAGES & DISADVANTAGES

11.CONCLUSION

12.FUTURE SCOPE

13.APPENDIX

- Source Code
- Github & Project Demo Link

1. Introduction

1.1 Project Overview

The title for this project is Smart Farmer IoT Based Smart Farming Application. The main goal of this project is to assist farmers in automating their farms by granting them access to a Web App.

1.2 Purpose

The Web App allows them to remotely control equipment like water motors as well as other gadgets without even being physically present in the field while also monitoring field parameters like temperature, soil moisture, and humidity.

2. Literature Survey

2.1 Existing Problem

Mr.Raja ,a 55-year-old city resident who owns an agricultural land in his hometown, is 55 years old. He wants to use a smartphone application to remotely monitor the field's characteristics and make decisions about whether or not to irrigate the crop.

2.2 Reference

- Zhang, L., Dabipi, I. K. And Brown, W. L, "Internet of Things Applications for Agriculture".
- Mat I., Mohd Kassim M. R., Harun A. N. and Yusoff I. M. 2018 *2018 IEEE Conference on Open Systems (ICOS) Smart Agriculture Using Internet of Things* 54-59
- S Navulur and M. N. Giri Prasad, "Agricultural management through wireless sensors and Internet of Things", *Int. J. Elect. Comput. Eng.*, vol. 7, pp. 3492-3499, 2017.

2.3 Problem Definition

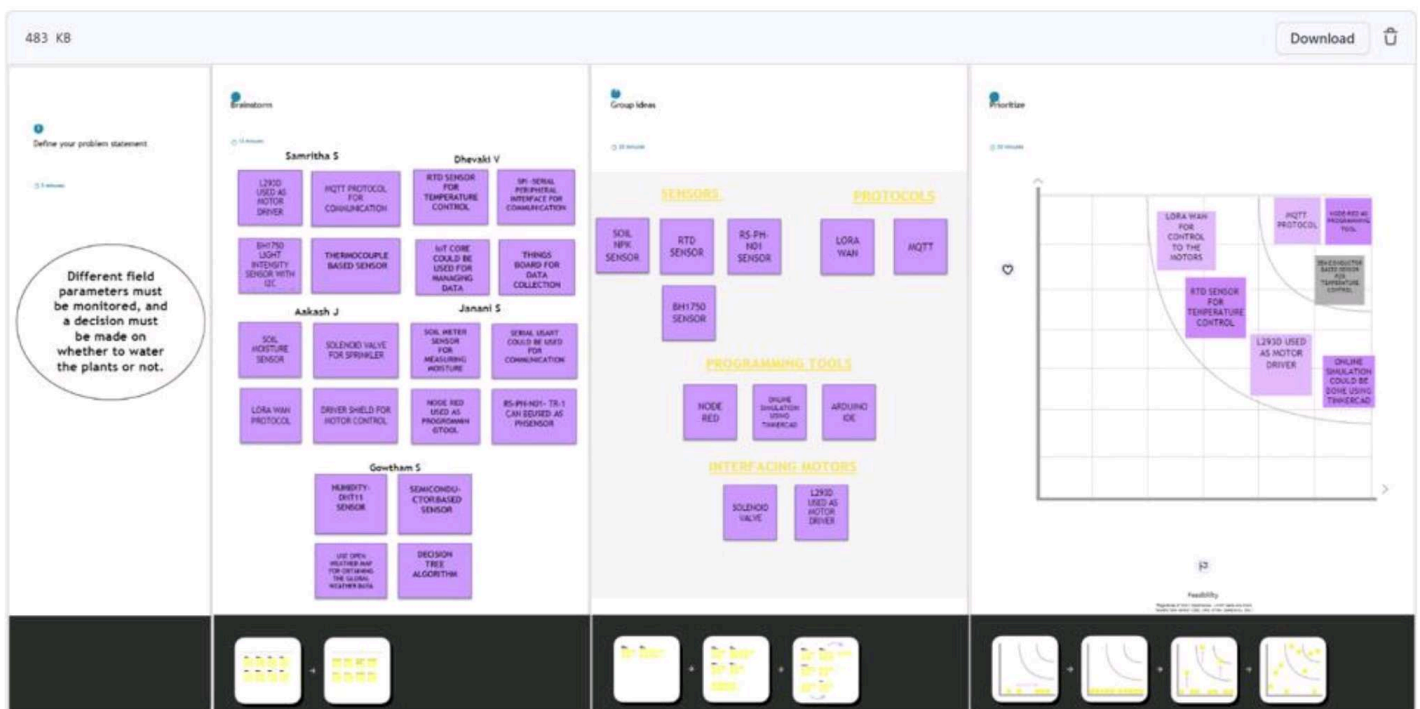
In agricultural aspects, if the plant is not provided with sufficient water, the production of the crop will be affected to a great extent. Providing correct amount of water is a challenge for the farmers. When the weather condition is uncertain, it is difficult to decide whether to water the crop or not.

3. Ideation & Proposed Solution

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming



3.3 Proposed Solution

We bring IoT services to the farmer in order to enhance and facilitate his working circumstances. These services make it possible for the farmer to operate remotely using the internet and cloud services. Through a smartphone app, user may monitor the field's characteristics and manage the farm's equipment.

3.4 Problem Solution Fit

Define CS, fit into CC	<p>1. CUSTOMER SEGMENT(S) Who is your customer?</p> <p>CS</p> <p>A farmer who raises crops is the target market for this product. Our intention is to assist them by remotely monitoring field conditions. This product prevents the demise of agriculture.</p>	<p>6. CUSTOMER</p> <p>What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.</p> <p>C</p> <p>It's tough to use many sensors at once. For success, you must have limitless or constant internet access.</p>	<p>5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face the problem?</p> <p>or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper</p> <p>AS</p> <p>Using IoT, the watering process is automated. To automate the watering operation, field characteristics and meteorological data were gathered and analysed. Efficiency is limited over small distances, and data storage is challenging.</p>	Explore AS, differentiate
Focus on J&P, tap into BE, understand RC	<p>2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.</p> <p>J&P</p> <p>This product's function is to employ sensors to collect different field characteristics and then process them through a centralised processing system. IoT uses the cloud to send and store data. Farmers utilise the Weather API to aid in decision-making. Through mobile applications, farmers may make decisions.</p>	<p>9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind</p> <p>R</p> <p>Farmers found it challenging to pursue agriculture because of the frequently changing and unpredictable weather and environment. Considering these elements is crucial when determining whether to water your plants. When the farmer is not on the field, it is impossible to supervise the field, which might cause crop damage.</p>	<p>7. BEHAVIOUR What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)</p> <p>B</p> <p>To counteract the consequences of extra water from heavy rain, use a suitable drainage system. the use of pest-resistant hybrid plants.</p>	Focus on J&P, tap into BE, understand RC
	<p>3. TRIGGERS What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.</p> <p>TR</p> <p>It is difficult for farmers to supply enough irrigation. Reduced yields and lower profits are consequences of inadequate water supplies for farmers. Weather forecasting is difficult for farmers.</p> <p>4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.</p> <p>EM</p> <p>BEFORE: Poor weather predicting skills; irrational choices; poor return. AFTER: Reliable data, an informed choice, and a high yield</p>	<p>10. YOUR SOLUTION If you are working on an existing business, write down your current solution first. Fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</p> <p>SL</p> <p>Our device gathers information from several sensor kinds and transmits the values to our primary server. The Weather API is also used to get weather information. The farmer uses a smartphone application to make the final decision about irrigation of the crop.</p>	<p>8. CHANNELS of BEHAVIOUR 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7</p> <p>8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development</p> <p>CH</p> <p>ONLINE: Giving the farmer access to information on the pH and moisture content of the soil by way of the internet. The user will receive online help for utilising the product</p> <p>OFFLINE: Education camps will be held to spread awareness of the value and benefits of automation and IoT in the advancement of agriculture.</p>	

4. Requirement Analysis

4.1 Functional Requirement

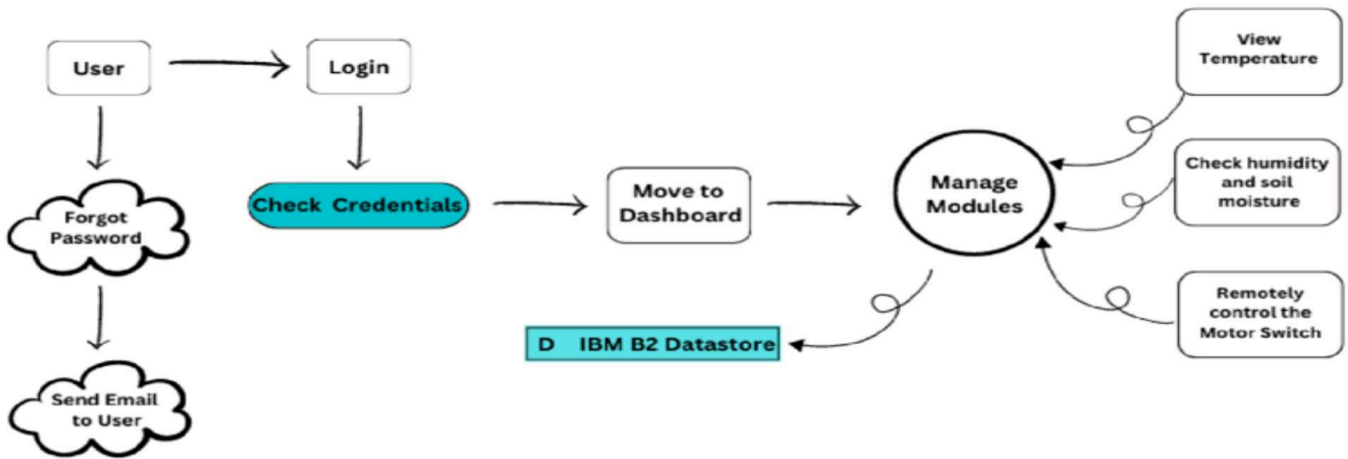
FR No.	Functional Requirement (Epic)	Requirement for a supporting story or task
FR-1	User Registration	Utilizing a Form for Registration signing up with Gmail
FR-2	User Confirmation	Email confirmation required Reassurance through OTP
FR-3	Sensor Function for framing System	Take temperature and humidity readings. Monitoring the Soil Measurement Look into crop diseases.
FR-4	Manage Modules	Manage User Roles controlling user permission
FR-5	Check whether details	temperature information Humidity information
FR-6	Data Management	Organize the information about the weather Control the agricultural conditions data. control the live stock circumstances data

4.2 Non-Functional Requirements

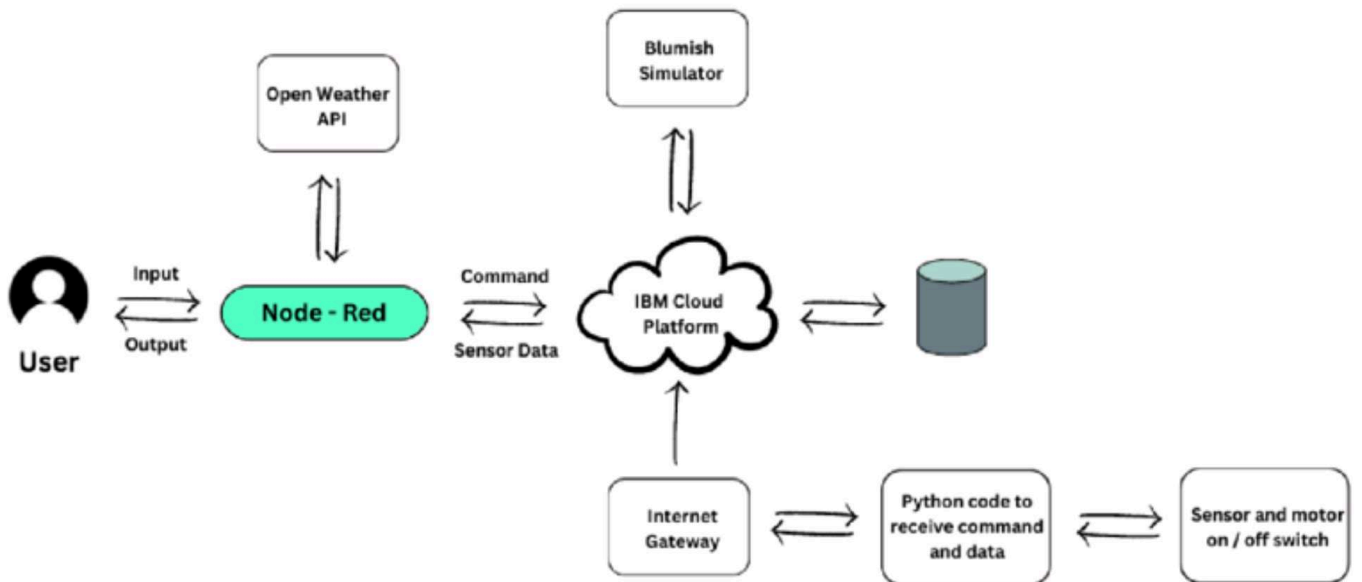
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	<ul style="list-style-type: none">✓ User friendly guidelines for users to avail the features.✓ Most simplistic user interface for ease of use.
NFR-2	Security	<ul style="list-style-type: none">✓ All the details about the user are protected from unauthorized access.✓ Detection and identification of any misfunctions of sensors.
NFR-3	Reliability	<ul style="list-style-type: none">✓ Implementing Mesh IoT Networks✓ Building a Multi-layered defence 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.

5. Project Design

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	I can sign up for the application as a user by providing my email address, a password, and a password confirmation.	I can access my account / dashboard	High	Sprint-1
		USN-2	Once I've signed up for the application, I, as a user, will receive a confirmation email.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	I may sign up for the application as a user using Gmail.		Medium	Sprint-1
	Login	USN-4	I may access the application as a user by providing my email address and password.		High	Sprint-1
Customer (Web user)	Dashboard	USN-5	As a user, you may visit the dashboard, where you can verify the access roles before moving on to the manage modules.	In this system of smart agricultural applications, I can see the dashboard.	High	Sprint 2
		USN-6	The motor switch is remote-accessible by the user.	In the app for smart farming	High	Sprint 3
Administrator			Once a user views the manage modules, they are described together with the manage system admins, manage user roles, etc.			Sprint 2

6.Problem Planning & Scheduling

6.1 Sprint Planning & Estimation & 6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	UserStory/Task	Story Points	Priority	Team Members
Sprint-1	Software	USN-1	Creating account in IBM cloud ,IBM Watson IoT and Node-Red. Adding device in the IBM Watson IoT platform.	2	High	Aakash J, Samritha S, Dhevaki V, Janani S, Gowtham S
Sprint-2	Program	USN-2	Developing the Python code	2	High	AakashJ, Dhevaki V

Sprint-3	Web Application	USN-3	Develop an application for the Smart farmer project using Node-RED	2	High	Aakash J
Sprint-3	MITApp Inventor	USN-3	Develop an application for the Smart farmer project using MIT App Inventor	2	High	Samritha S, Aakash J
Sprint-4	WebUI	USN-4	To make the user to interact with the software	2	High	Samritha S, Aakash J

PNT2022TMID31713
Software project

PLANNING

- Roadmap
- Backlog
- Board

DEVELOPMENT

Code

- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
[Learn more](#)

Projects / PNT2022TMID31713 / Add epic / PNT2022TMI-7

develope python code

Attach Add a child issue Link issue ...

Description

Add a description...

Attachments (1)



Activity

Show: All Comments History

Newest first 17

SS Add a comment...

Pro tip: press **M** to comment

1

Done Done

Details

Assignee SS Samritha Sudarsan

Labels None

Sprint None

Story point estimate None

Reporter SS Samritha Sudarsan

Created 13 minutes ago
Updated 12 minutes ago
Resolved 12 minutes ago

Configure

PNT2022TMID31713
Software project

PLANNING

- Roadmap
- Backlog
- Board

DEVELOPMENT

Code

- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
[Learn more](#)

Projects / PNT2022TMID31713 / Add epic / PNT2022TMI-9

connect to mit application

Attach Add a child issue Link issue ...

Description

Add a description...

Attachments (1)



Activity

Show: All Comments History

Newest first 17

SS Add a comment...

Pro tip: press **M** to comment

1

Done Done

Details

Assignee SS Samritha Sudarsan

Labels None

Sprint None

Story point estimate None

Reporter SS Samritha Sudarsan

Created 12 minutes ago
Updated 11 minutes ago
Resolved 11 minutes ago

Configure

PNT2022TMID31713
Software project

PLANNING

- Roadmap
- Backlog
- Board

DEVELOPMENT

Code

- Project pages
- Add shortcut
- Project settings

You're in a team-managed project
[Learn more](#)

Projects / PNT2022TMID31713 / Add epic / PNT2022TMI-8

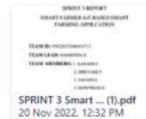
create a web application

Attach Add a child issue Link issue ...

Description

Add a description...

Attachments (1)



Activity

Show: All Comments History

Newest first 17

SS Add a comment...

Pro tip: press **M** to comment

1

Done Done

Details

Assignee Unassigned
Assign to me

Labels None

Sprint None

Story point estimate None

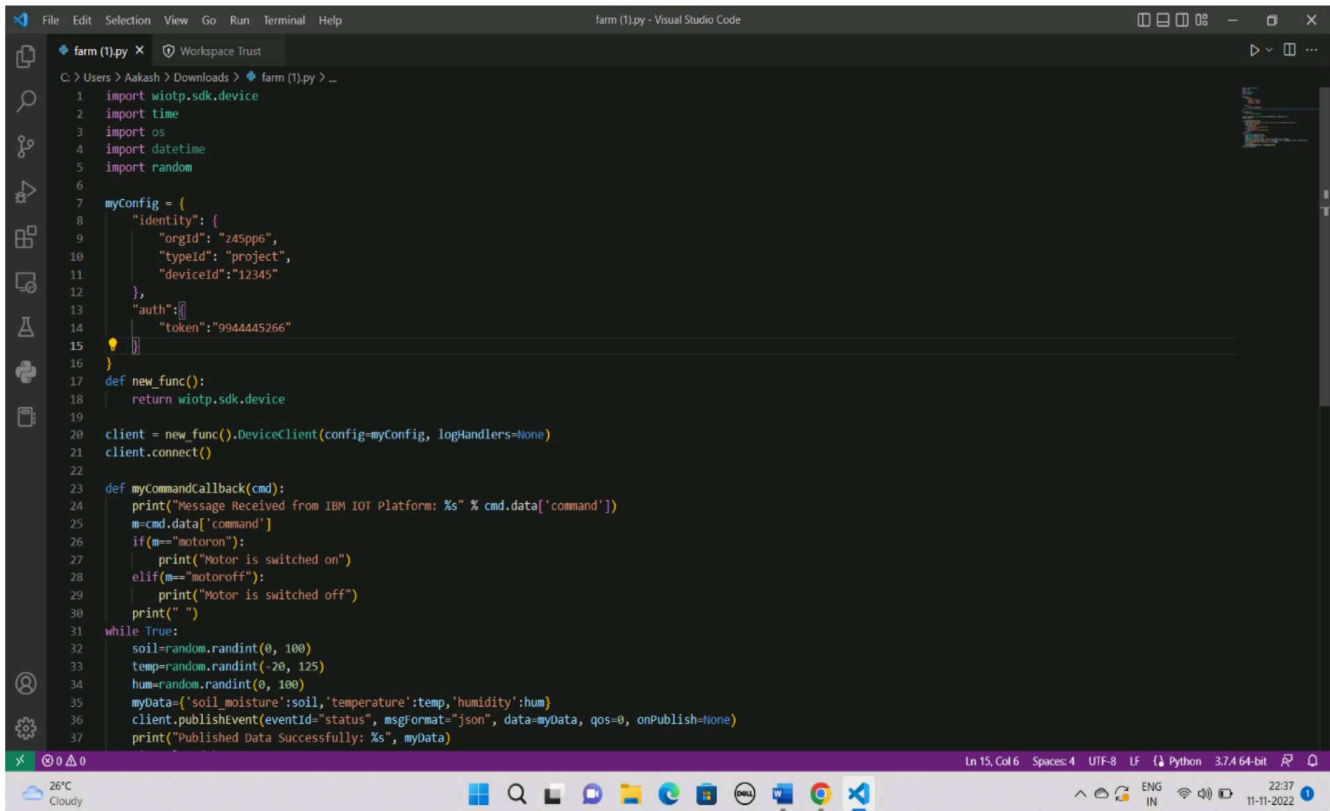
Reporter SS Samritha Sudarsan

Created 14 minutes ago
Updated 12 minutes ago
Resolved 12 minutes ago

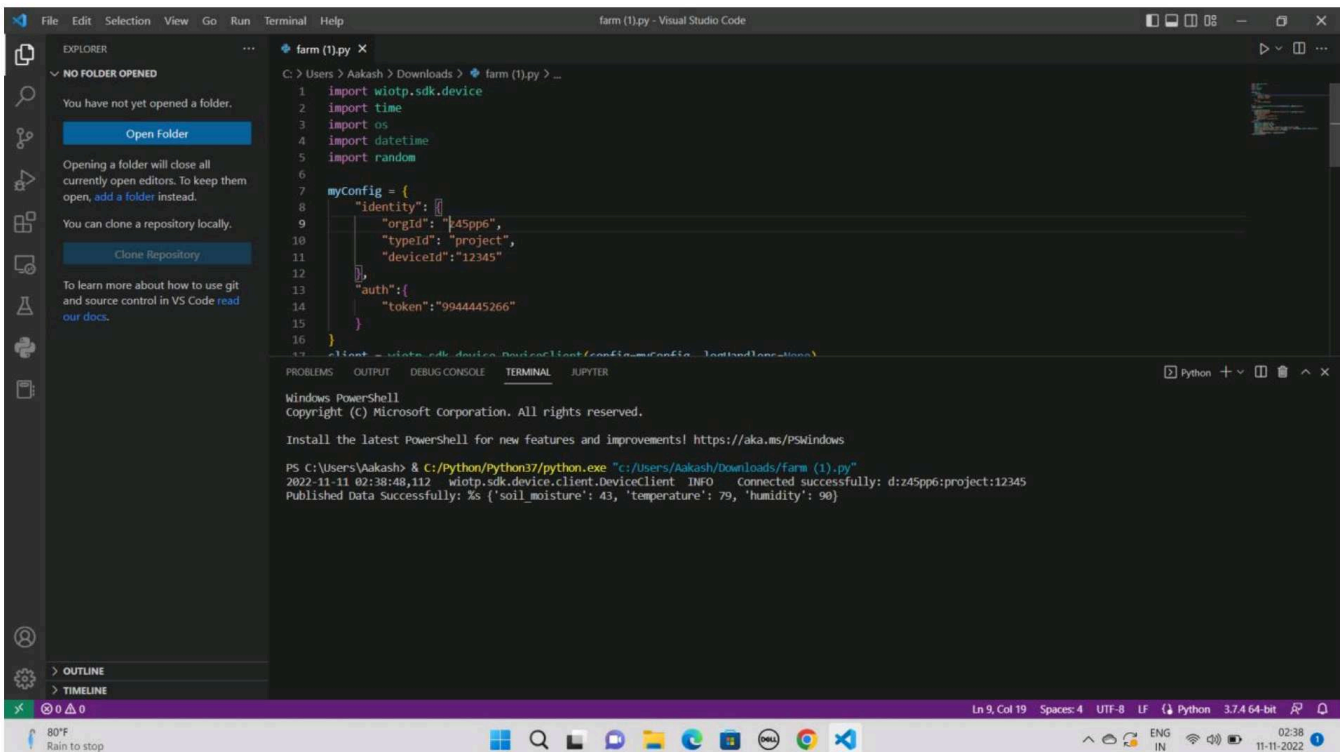
Configure

7. Coding & Solutioning

7.1 Feature 1 & 7.2 Feature 2



```
1 import wiotp.sdk.device
2 import time
3 import os
4 import datetime
5 import random
6
7 myConfig = {
8     "identity": {
9         "orgId": "z45pp6",
10        "typeId": "project",
11        "deviceId": "12345"
12    },
13    "auth": {
14        "token": "9944445266"
15    }
16 }
17
18 def new_func():
19     return wiotp.sdk.device
20
21 client = new_func().DeviceClient(config=myConfig, logHandlers=None)
22 client.connect()
23
24 def myCommandCallback(cmd):
25     print("Message Received from IBM IoT Platform: %s" % cmd.data['command'])
26     m=cmd.data['command']
27     if(m=="motoron"):
28         print("Motor is switched on")
29     elif(m=="motoroff"):
30         print("Motor is switched off")
31     print(" ")
32 while True:
33     soil=random.randint(0, 100)
34     temp=random.randint(-20, 125)
35     hum=random.randint(0, 100)
36     myData={"soil_moisture":soil,"temperature":temp,"humidity":hum}
37     client.publishEvent(eventId="status", msgformat="json", data=myData, qos=0, onPublish=None)
38     print("Published Data Successfully: %s", myData)
```



```
PS C:\Users\Aakash> & C:/Python/Python37/python.exe "c:/Users/Aakash/Downloads/farm (1).py"
2022-11-11 02:38:48,112 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:z45pp6:project:12345
Published Data Successfully: %s {'soil_moisture': 43, 'temperature': 79, 'humidity': 90}
```

```
File Edit Selection View Go Run Terminal Help
farm (1).py - Visual Studio Code
farm (1).py x Workspace Trust
C:\Users\Aakash> Downloads > farm (1).py > ...
import wiotp.sdk.device

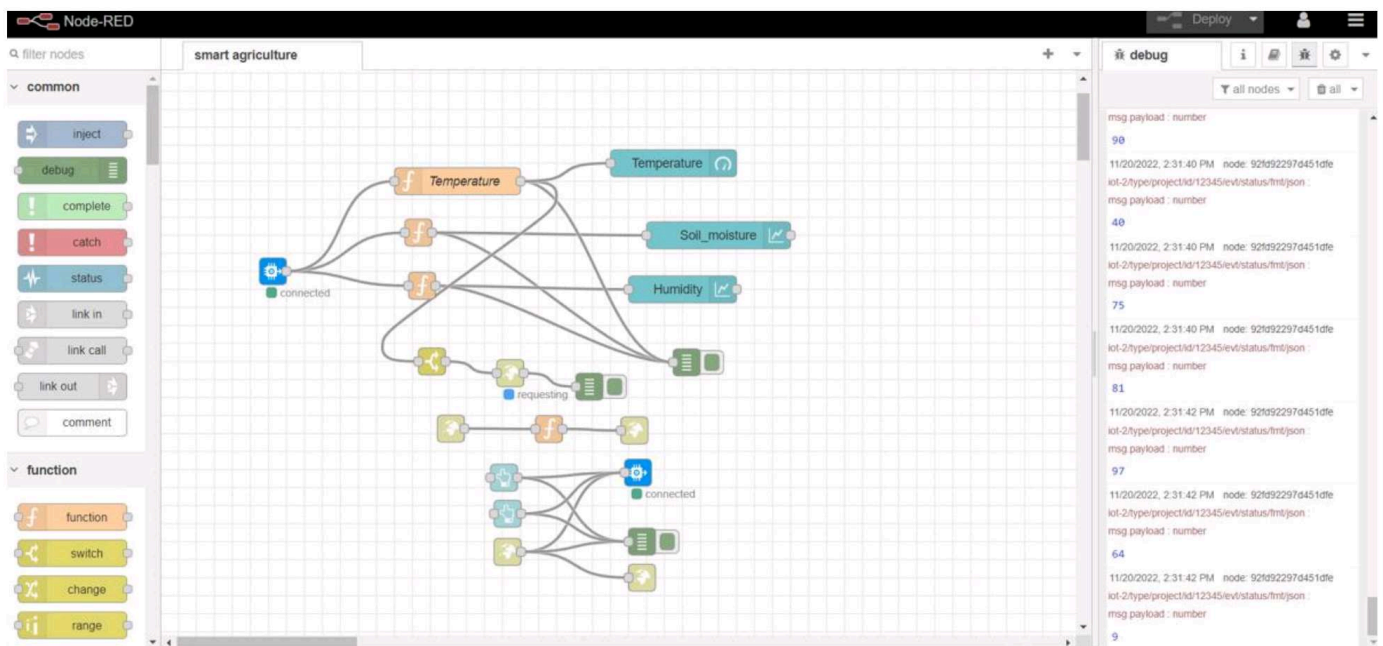
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Code
[Running] python -u "c:\Users\Aakash\Downloads\farm (1).py"
2022-11-11 23:19:42,002 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:z45pp6:project:12345
Published Data Successfully: %s ('soil_moisture': 83, 'temperature': 6, 'humidity': 99)
Published Data Successfully: %s ('soil_moisture': 60, 'temperature': 18, 'humidity': 25)
Published Data Successfully: %s ('soil_moisture': 63, 'temperature': 43, 'humidity': 89)
Published Data Successfully: %s ('soil_moisture': 74, 'temperature': 102, 'humidity': 92)
Published Data Successfully: %s ('soil_moisture': 52, 'temperature': 111, 'humidity': 55)
Published Data Successfully: %s ('soil_moisture': 77, 'temperature': 45, 'humidity': 81)
Published Data Successfully: %s ('soil_moisture': 9, 'temperature': 121, 'humidity': 96)
Published Data Successfully: %s ('soil_moisture': 22, 'temperature': 12, 'humidity': 69)
[Done] exited with code=1 in 15.933 seconds

[Running] python -u "c:\Users\Aakash\Downloads\farm (1).py"
2022-11-11 23:19:59,366 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:z45pp6:project:12345
Published Data Successfully: %s ('soil_moisture': 61, 'temperature': 113, 'humidity': 6)
Published Data Successfully: %s ('soil_moisture': 54, 'temperature': -17, 'humidity': 67)
Published Data Successfully: %s ('soil_moisture': 7, 'temperature': 38, 'humidity': 27)
Published Data Successfully: %s ('soil_moisture': 40, 'temperature': 60, 'humidity': 31)
Published Data Successfully: %s ('soil_moisture': 46, 'temperature': 12, 'humidity': 19)
Published Data Successfully: %s ('soil_moisture': 63, 'temperature': 8, 'humidity': 8)
Published Data Successfully: %s ('soil_moisture': 59, 'temperature': 9, 'humidity': 31)

Ln 22, Col 1 Spaces: 4 UTF-8 LF Python 3.7.4 64-bit
26°C Cloudy
```

8. Testing

8.1 Test Cases:



8.2 User Acceptance Testing

Screen1

Smart Agriculture

Temperature99

Humidity14

Moisturenot found

MOTOR ONMOTOR OFF

Receiving Data

Screen1

Smart Agriculture

Temperature69

Humidity54

Moisture45

MOTOR ONMOTOR OFF

Screen1

Smart Agriculture

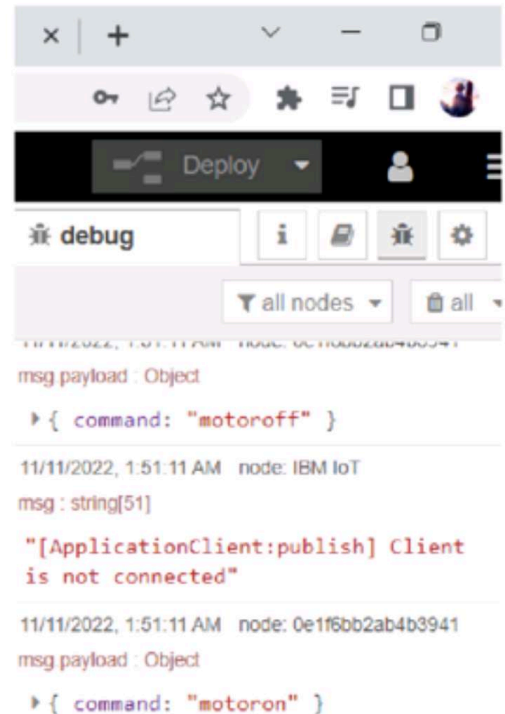
Temperature 69

Humidity 54

Moisture 45

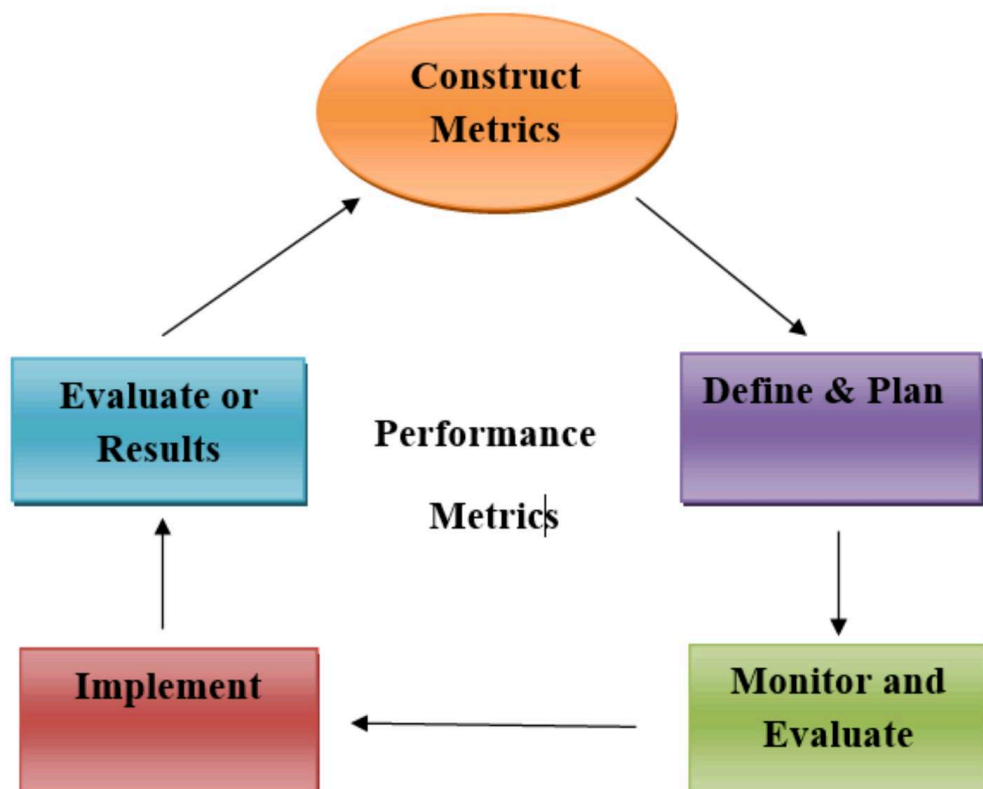
MOTOR ON

MOTOR OFF



9. Results

9.1 Performance Metrics



10. Advantages:

- Farms can be monitored and controlled remotely
- Increase inconvenience to farmers
- Less cost
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the technology.
- Less Accuracy

11. Conclusion

Thus, the project's goal of putting in place an IoT system to assist farmers in managing and monitoring their fields has been accomplished.

12. Future Scope

Project Summary

IoT-based smart agriculture systems can keep an eye on the weather and soil moisture to help crops grow and produce well. Using other systems like Open Weather API, the farmer may also obtain data about the current weather prediction. The farmer is given a smartphone app so he can keep track of the parameters for temperature, humidity, and soil moisture as well as weather predicting information. He may use the mobile application to regulate the motors to irrigate his crop based on all the factors. Farmers may irrigate their crops even when they are far away from them by utilizing a smartphone application to manage the motors. Here, we're utilizing an online IoT simulator to obtain readings for the soil moisture, humidity, and temperature.

Technical Requirements:

- IoT Simulator

Software Requirements:

- Python
- Node-Red
- IBM Watson IoT Platform
- Fast to SMS

Project Deliverables:

A Web App for farmers where he can:

- monitor temperature, humidity and soil moisture details.
- control motor for watering the crop.

Project Team:

Samritha.S, Aakash.J, Dhevali.V, Janani.S, Gowtham.S

13.Appendix

Source Code:

```
import wiotp.sdk.device
import time
import os
import datetime
import random

myConfig = {
    "identity": {
        "orgId": "z45pp6",
        "typeId": "project",
        "deviceId":"12345"
    },
    "auth":{
        "token":"99444445266"
    }
}

def new_func():
    return wiotp.sdk.device

client = new_func().DeviceClient(config=myConfig, logHandlers=None)
client.connect()

def myCommandCallback(cmd):
    print("Message Received from IBM IOT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="motoron"):
        print("Motor is switched on")
    elif(m=="motoroff"):
        print("Motor is switched off")
    print(" ")
while True:
    soil=random.randint(0, 100)
    temp=random.randint(-20, 125)
    hum=random.randint(0, 100)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published Data Successfully: %s", myData)
    time.sleep(2)
    client.myCommandCallback = myCommandCallback
client.disconnect()
```

GitHub & Project Demo Link:

GitHub: <https://github.com/IBM-EPBL/IBM-Project-11541-1659333960>

Project Demo Link: <https://drive.google.com/file/d/1ffC2hyLKk8Ho0GBJGjePgrb098pzqTRw/view?usp=sharing>