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

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.

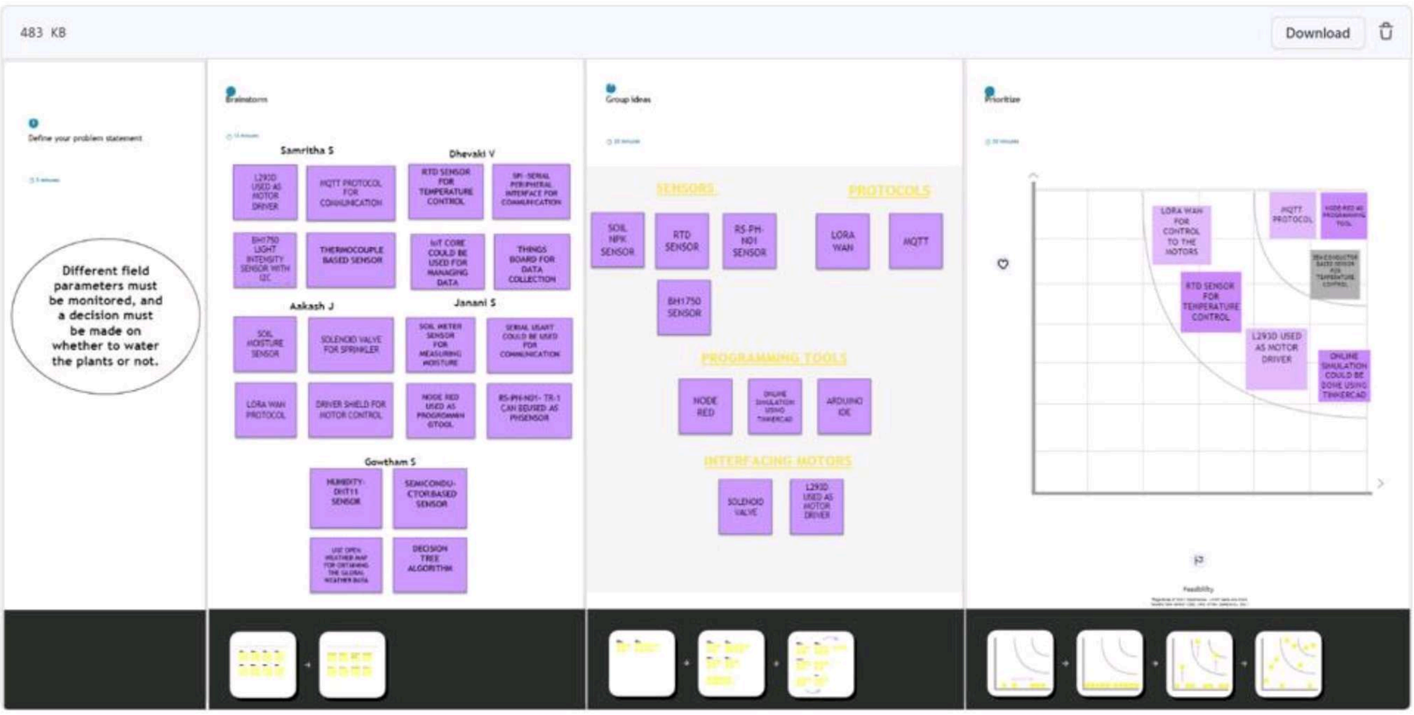
2.3 Problem Definition

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

4. Requirement Analysis

4.1 Functional Requirement

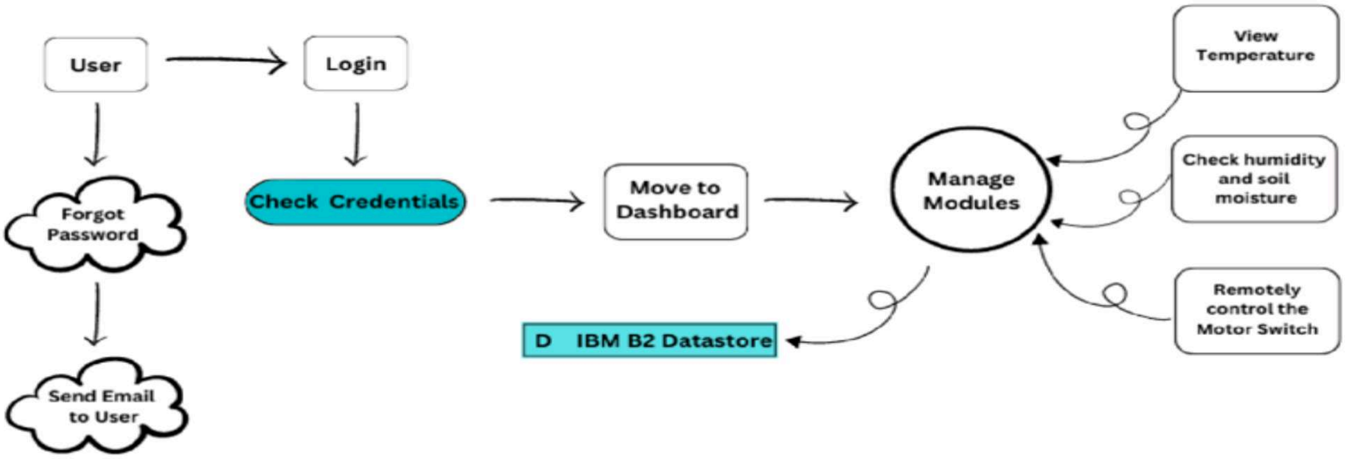
FR No.	Functional Requirement (Epic)	Requirement for a supporting story or task
FR-1	User Registration	Utilizing a Form for Registration signing up with Gmail
FR-2	User Confirmation	Email confirmation required Reassurance through OTP
FR-3	Sensor Function for framing System	Take temperature and humidity 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

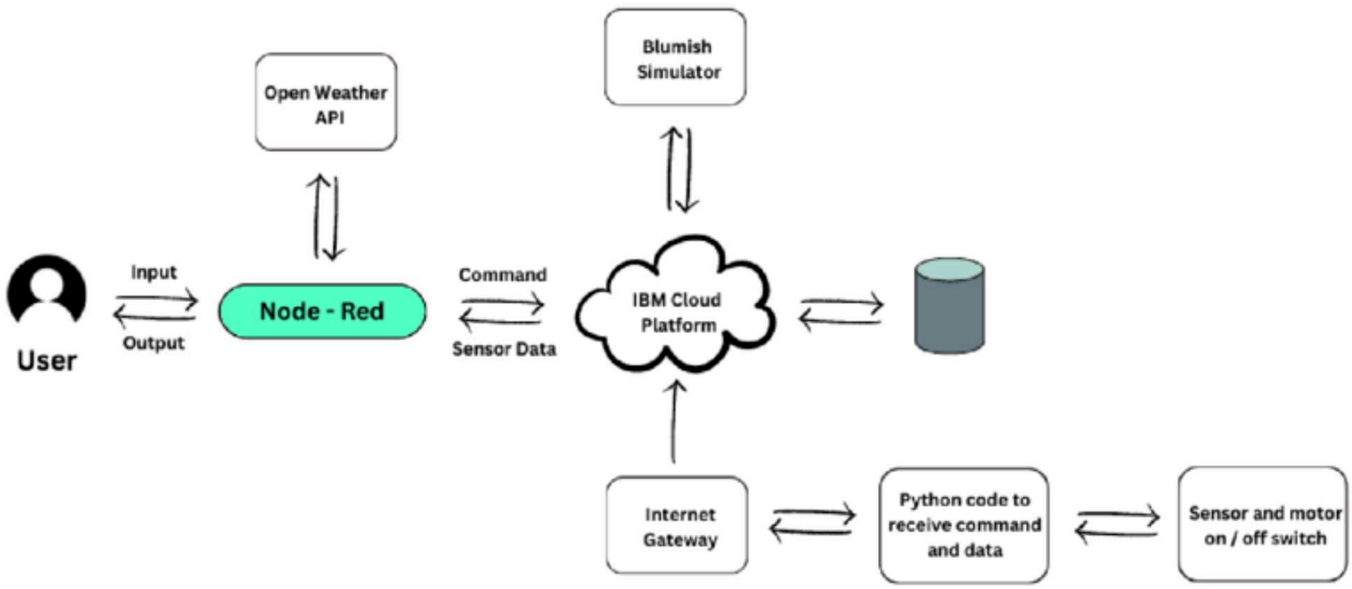
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	✓ User friendly guidelines for users to avail the features. ✓ Most simplistic user interface for ease of use.
NFR-2	**Security**	✓ All the details about the user are protected from unauthorized access. ✓ Detection and identification of any misfunctions of sensors.
NFR-3	**Reliability**	✓ Implementing Mesh IoT Networks ✓ Building a Multi-layered 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	FunctionalRequirement (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

Project Tracker, Velocity &Burn down Chart:(4Marks)

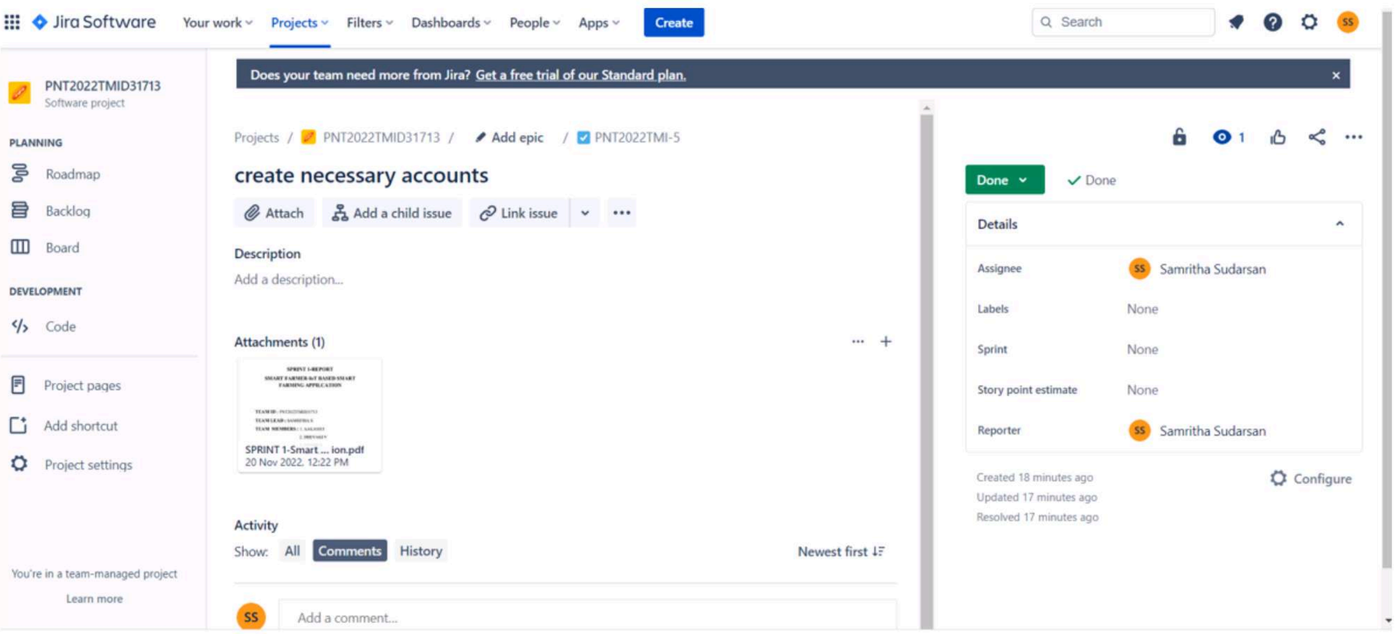
Sprint	Total Story Points	Duration	Sprint StartDate	Sprint EndDate(Planned)	Story PointsCompleted(as onPlanned EndDate)	Sprint Release Date(Actual)
Sprint-1	20	7 Days	03 Nov 2022	09Nov2022	20	29Oct 2022
Sprint-2	20	9 Days	31Oct 2022	09Nov2022		05Oct 2022
Sprint-3	20	6 Days	06Nov2022	13Nov2022		12Oct 2022
Sprint-4	20	6 Days	09Nov2022	15Nov2022		15Oct 2022

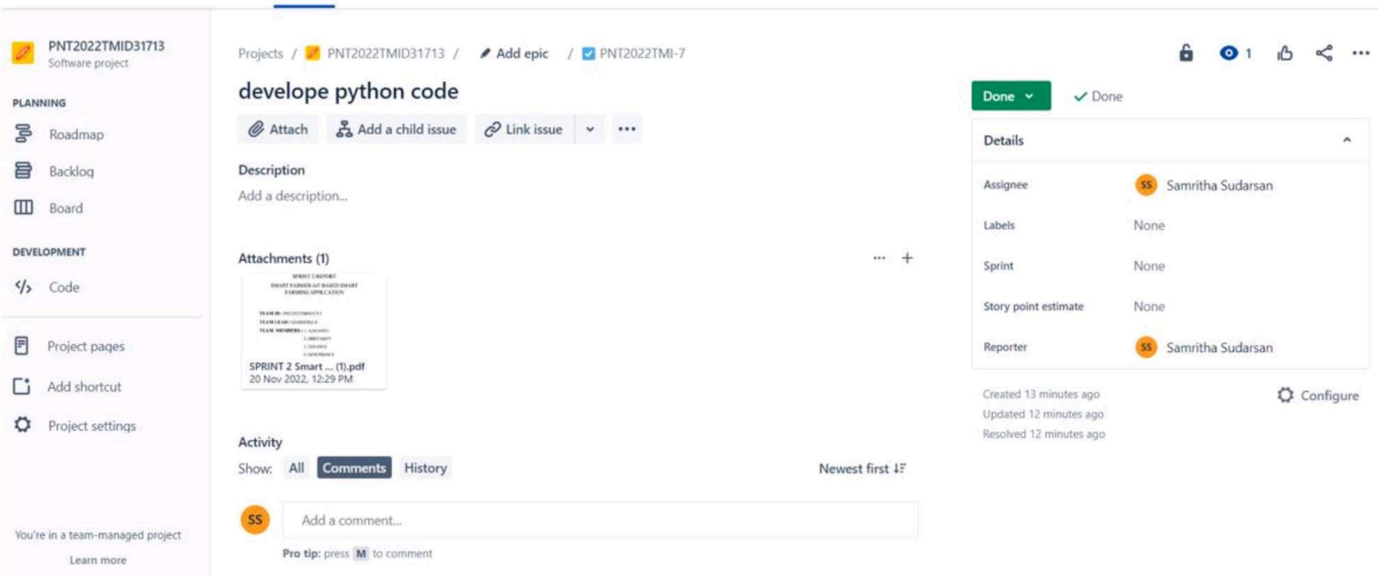
Velocity:

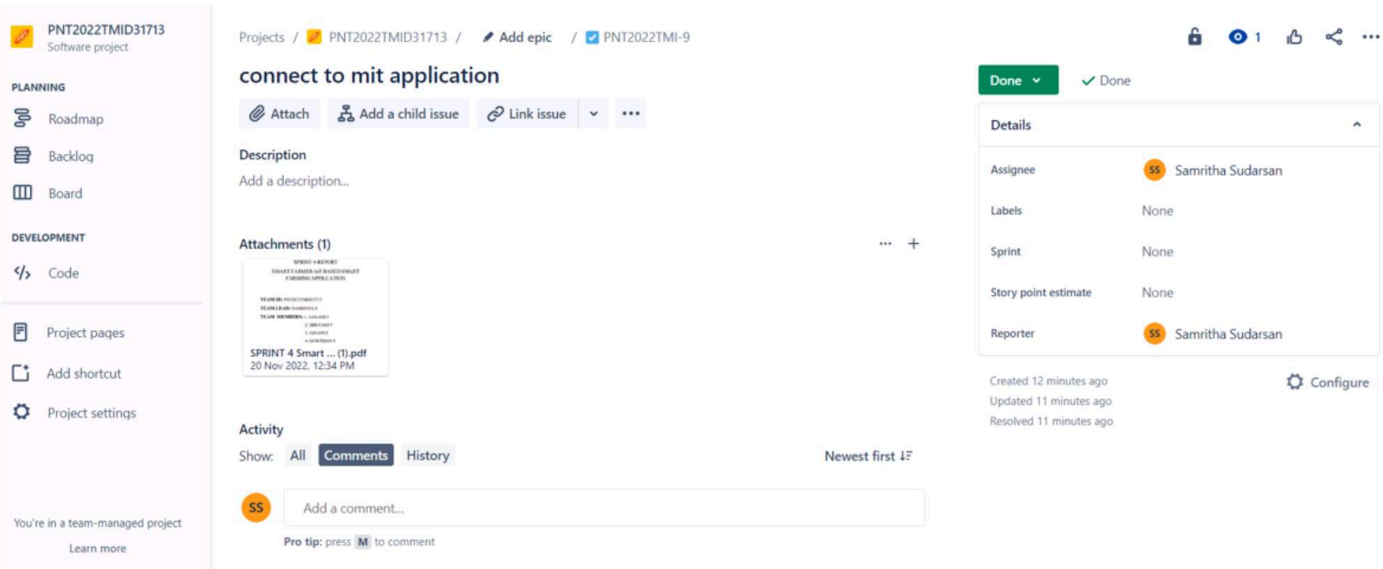
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity(AV) per iteration unit(story points per day)

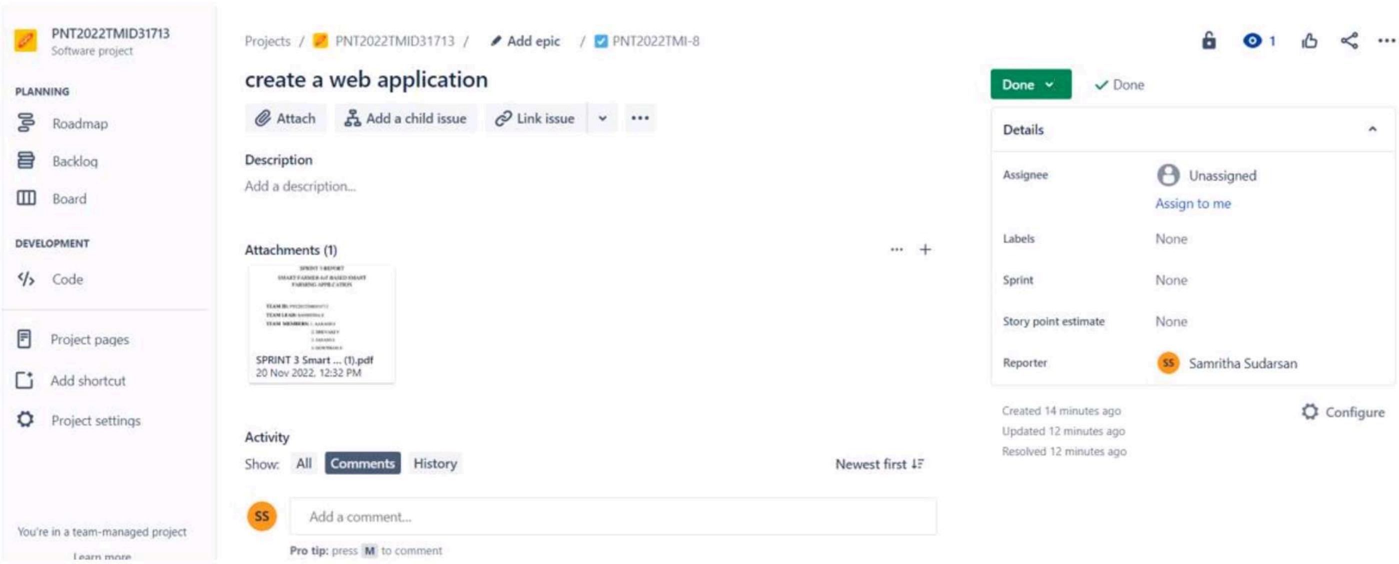
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports from JIRA



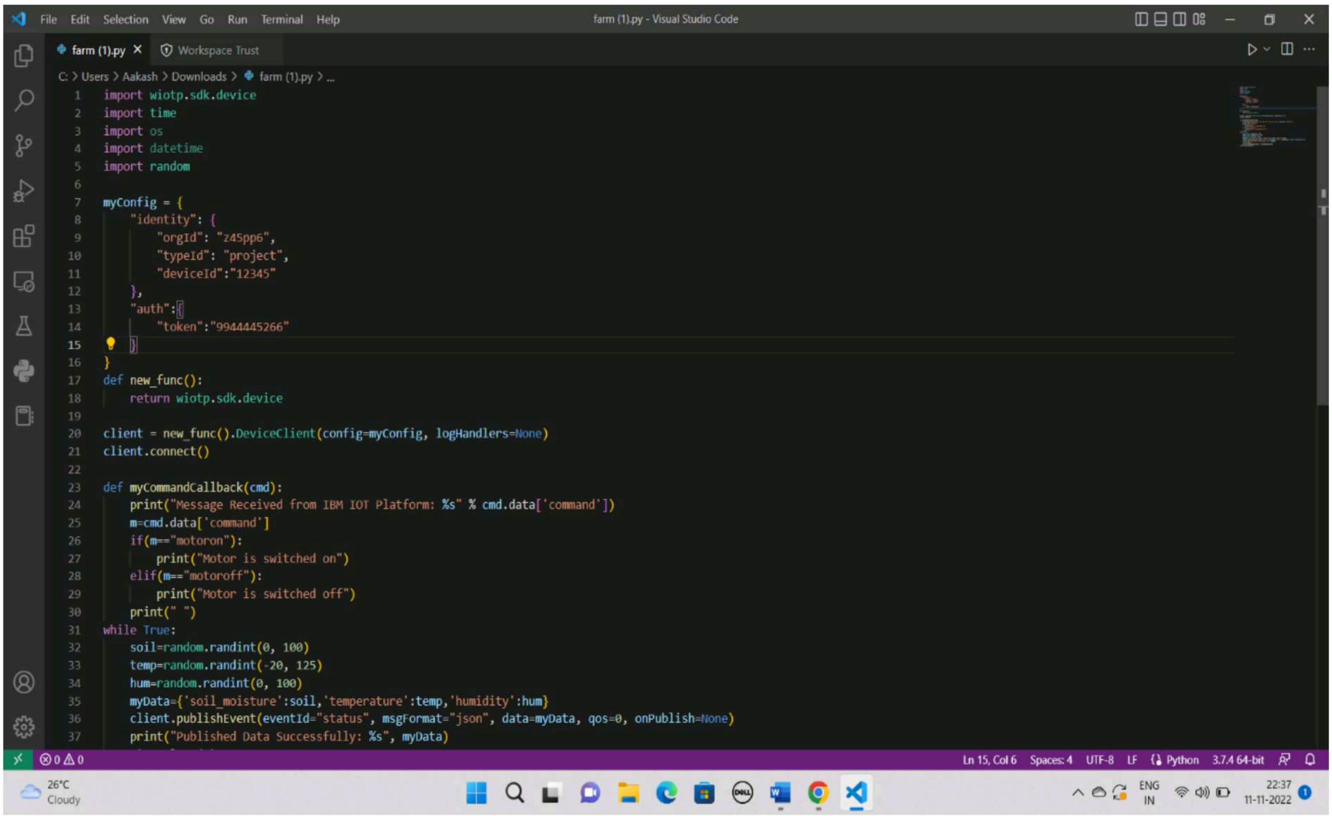

PNT2022TMID31713
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / PNT2022TMID31713 / Add epic / PNT2022TMI-7
develope python code
Attach
Add a child issue
Link issue
Description
Add a description...
Attachments (1)
SPRINT 2 Smart ... (1).pdf
20 Nov 2022, 12:29 PM
Activity
Show: All Comments History
Newest first
SS
Add a comment...
Pro tip: press M to comment
Done
Done
Details
Assignee Samritha Sudarsan
Labels None
Sprint None
Story point estimate None
Reporter Samritha Sudarsan
Created 13 minutes ago
Updated 12 minutes ago
Resolved 12 minutes ago
Configure

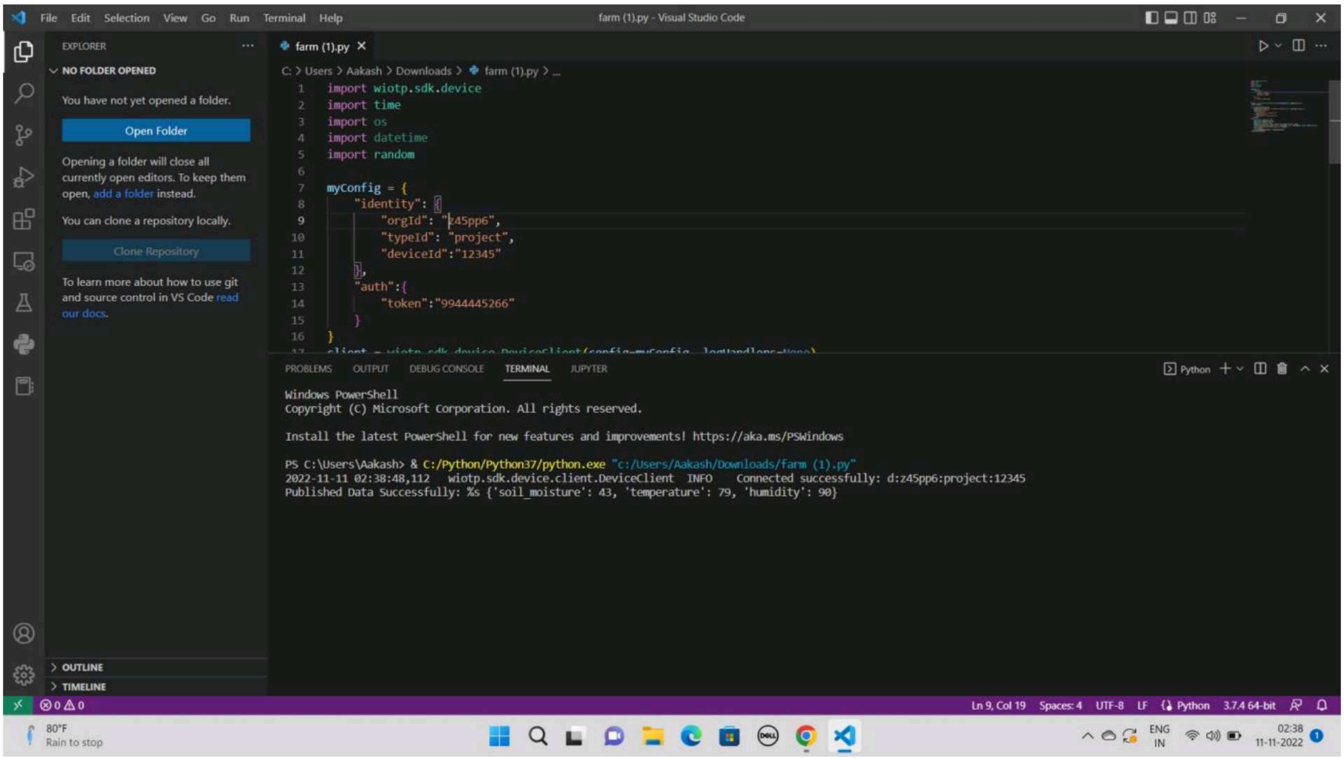

PNT2022TMID31713
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Projects / PNT2022TMID31713 / Add epic / PNT2022TMI-9
connect to mit application
Attach
Add a child issue
Link issue
Description
Add a description...
Attachments (1)
SPRINT 4 Smart ... (1).pdf
20 Nov 2022, 12:34 PM
Activity
Show: All Comments History
Newest first
SS
Add a comment...
Pro tip: press M to comment
Done
Done
Details
Assignee Samritha Sudarsan
Labels None
Sprint None
Story point estimate None
Reporter Samritha Sudarsan
Created 12 minutes ago
Updated 11 minutes ago
Resolved 11 minutes ago
Configure

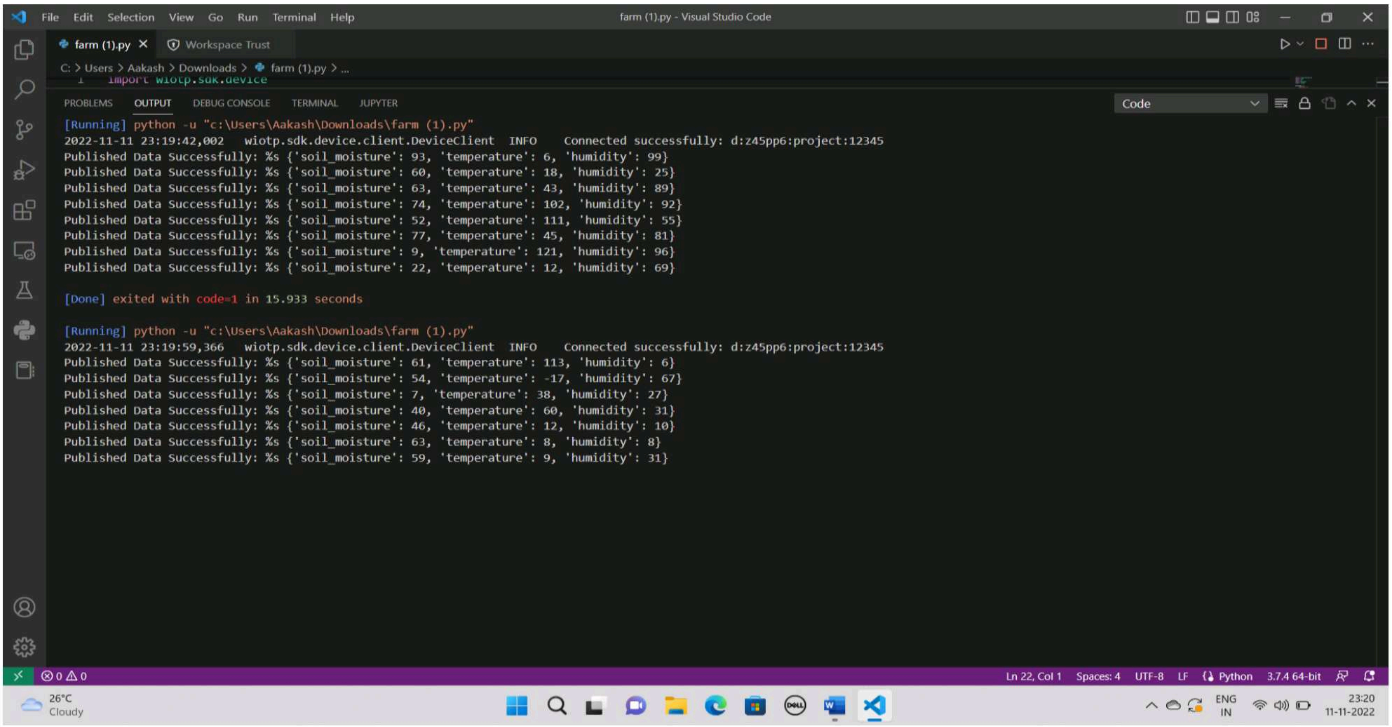

PNT2022TMID31713
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Projects / PNT2022TMID31713 / Add epic / PNT2022TMI-8
create a web application
Attach
Add a child issue
Link issue
Description
Add a description...
Attachments (1)
SPRINT 3 Smart ... (1).pdf
20 Nov 2022, 12:32 PM
Activity
Show: All Comments History
Newest first
SS
Add a comment...
Pro tip: press M to comment
Done
Done
Details
Assignee Unassigned
Assign to me
Labels None
Sprint None
Story point estimate None
Reporter 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

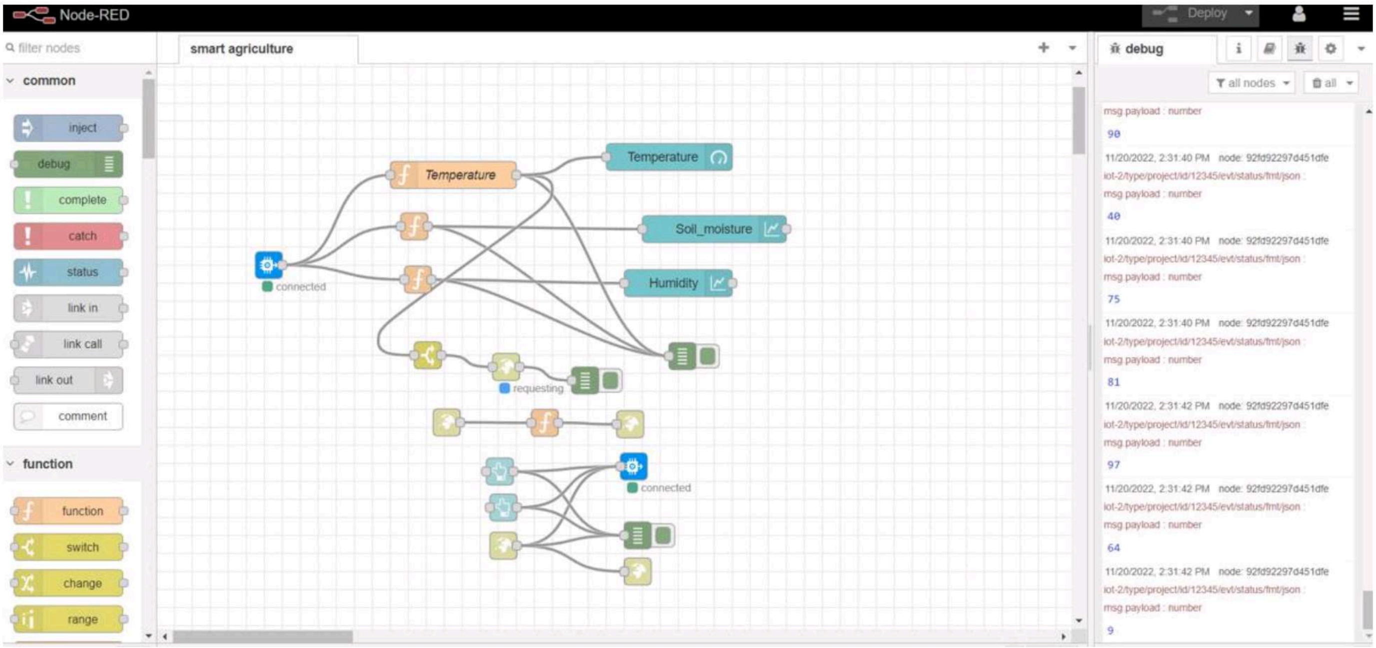






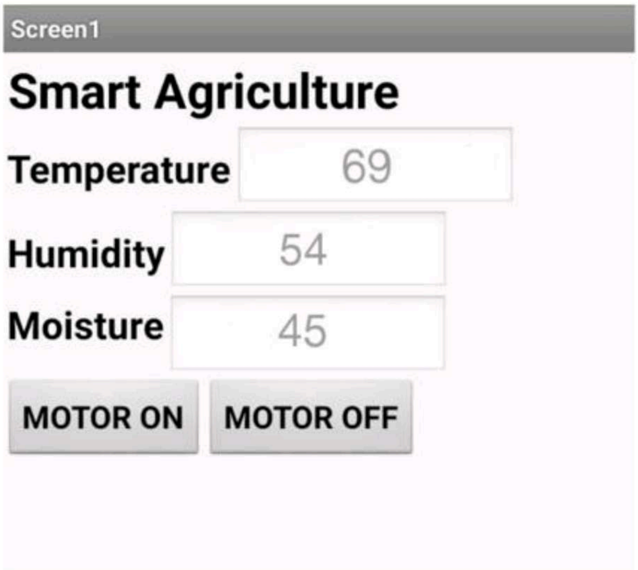
8. Testing

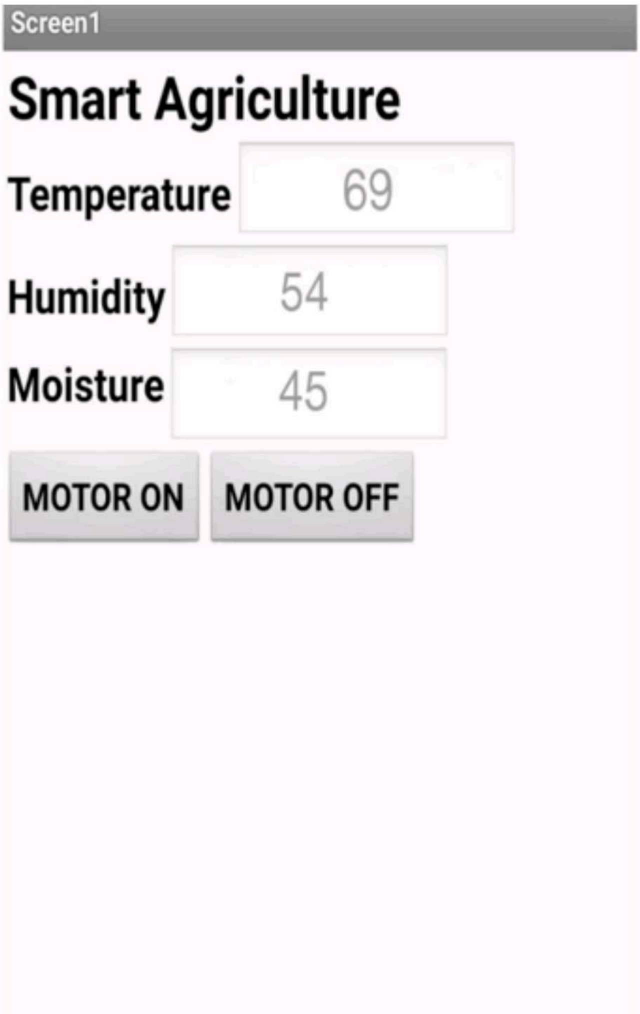
8.1 Test Cases:

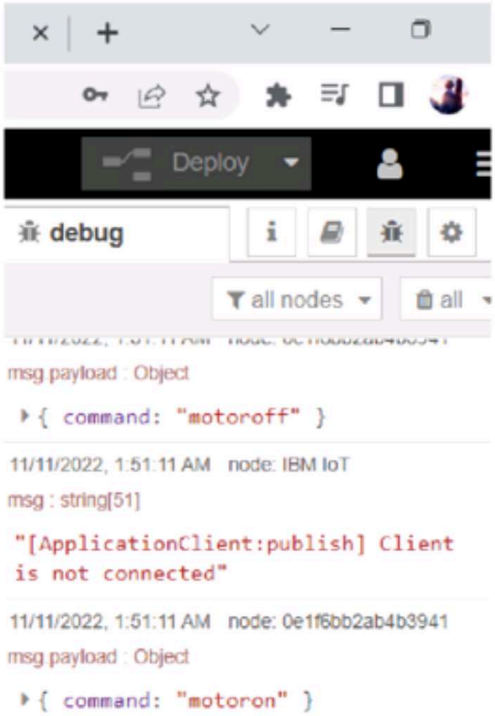


8.2 User Acceptance Testing

Receiving Data

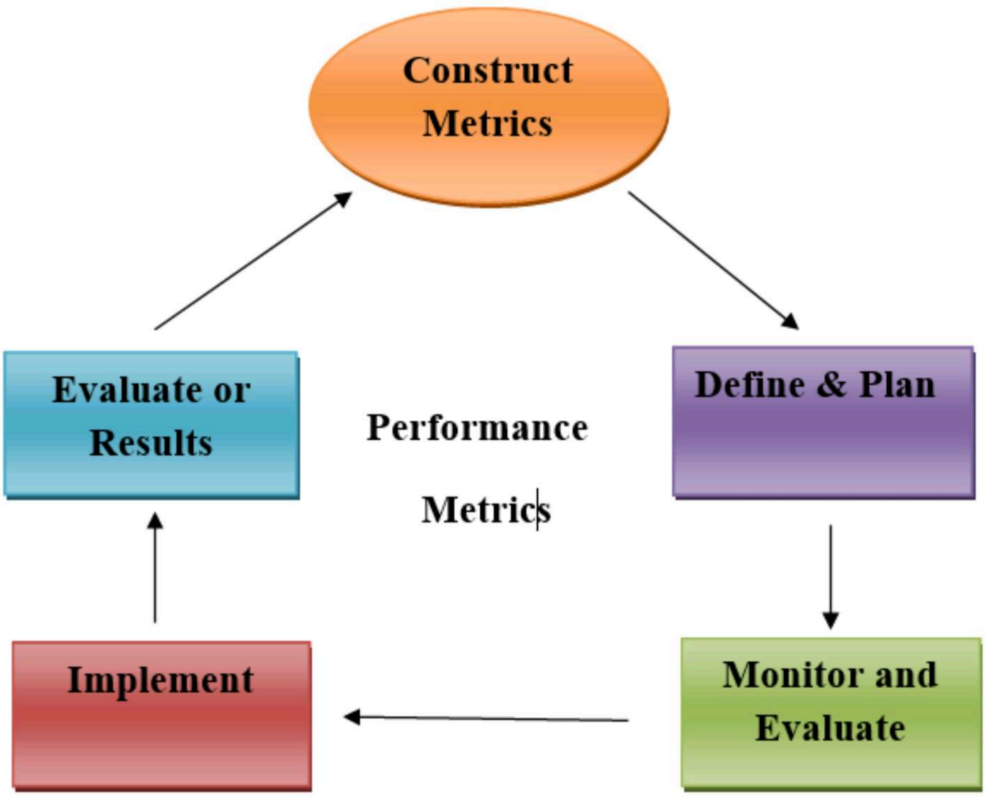






9. Results

9.1 Performance Metrics



10. Advantages:

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

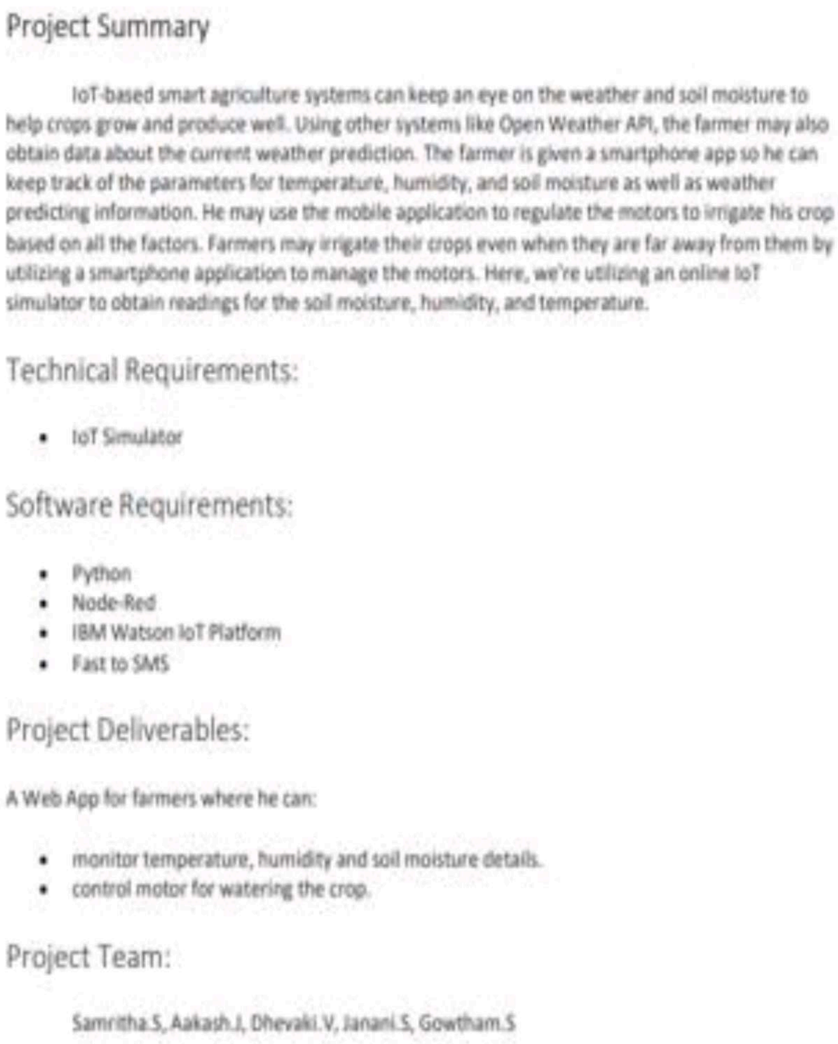
Disadvantages:

- 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, Dhevaki.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":"9944445266"
   }
}
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.disconnnect()
```

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