PROJECT REPORT

Project Name: SMARTFARMER- IOT ENABLED SMART FARMING APPLICATION

Team ID: PNT2022TMID34190

Team:

Sharmila Devi.N- TeamLead

Uthra C.R

Safrin.I

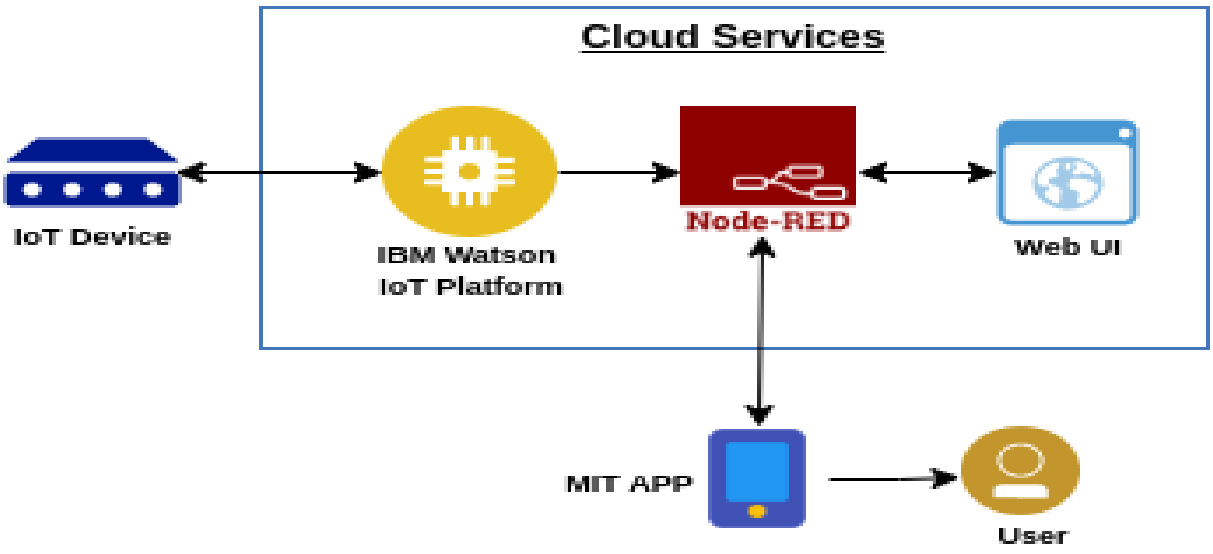
Sherlin Beaula S.B

Rammiya R

1. INTRODUCTION

Project Overview

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.



PURPOSE

The smart agriculture model main aim **to avoid water wastage in the irrigation process**. It is low cost and efficient system Is shown below. It includes NodeMCU, Arduino Nano, sensors like soil moisture and Dht11, solenoid valves, relays.

2.LITERATURE SURVEY

2.2 Existing problem

The challenges of a smart agriculture system include the integration of these sensors and tying the sensor data to the analytics driving automation and response activities. When integrated, the use of data analytics can reduce the overall cost of agriculture and contribute to higher production from the same amount of area through precise control of water, fertilizer and light. Smart methods allow for farming on smaller and more distributed lands through remote monitoring, whether indoor or outdoor.

To successfully deploy a smart agriculture system, consider setting up a communications network that can integrate a limited number of sensors across a large area of farmland. This will require third-party network provisioning or setting up a private network consisting of access points and uplinks to a private backhaul network, which channels all the data traffic to centralized monitoring software or an analytics head-end system

• It is not a secure system.

• There is no motion detection for protection of agriculture field.

• Automation is not available.

References

[1] ISSN No:-2456-2165 Volume 4, Issue 2 Feb – 2019: "Solars' Energy: - A safe and reliable, eco-friendly and sustainable Clean Energy Option for Future India: - A Review."

[2] Universal Paper of advanced science and science and exploration technology. [2] GRD Journals- Global Research and Development Journal for Engineering | Volume 4 | Issue 3 | February (2019) ISSN: 2455-5703 “Design and Implementation of an Advanced Security System for Farm Protection from Wild Animals”.

[3] International Journal of Innovationsin Engineering and Science, Impact Factor Value 4.046 e-ISSN: 2456-3463 Vol.4, No. 5, 2019 “Solar Powered Smart Fencing System for Agriculture Protection using GSM & Wireless Camera”.

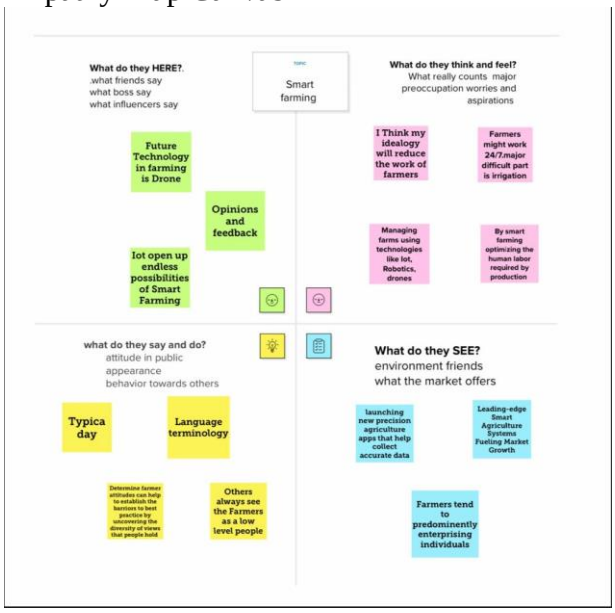
[4] International Journal of Management, Technology And Engineering ISSN NO : 2249-7455 Volume 8, Issue VII, JULY/2018”Protecting Crops From Birds, Using Sound Technology In Agriculture” [5] American Journal of Engineering Research (AJER)2018 eISSN: 2320-0847 p-ISSN : 2320- 0936 Volume-7, Issue-7, pp-326-330 “Moisture Sensing Automatic Plant Watering System Using Arduino Uno”.

Problem Statement Definition

The soil moisture sensor measures wetness content in the soil. The Arduino UNO microcontroller used to receive input from a various sensors and it can be controlled automatically. When soil moisture sensor goes low the water pump will be on and it exceeds defined levels of the water motor will turn off automatically. We can constantly monitor the growth of a crop using ultrasonic sensor. PIR sensor detects the motion or unusual movement in the agricultural land. This device his very helpful to the former to monitor and control environmental parameters at their field. The farmers did not go to theirfield, they can remotely monitor and control using cloud.

3. IDEATION & PROPOSED SOLUTION

Empathy Map Canvas



Ideation and Brainstorming

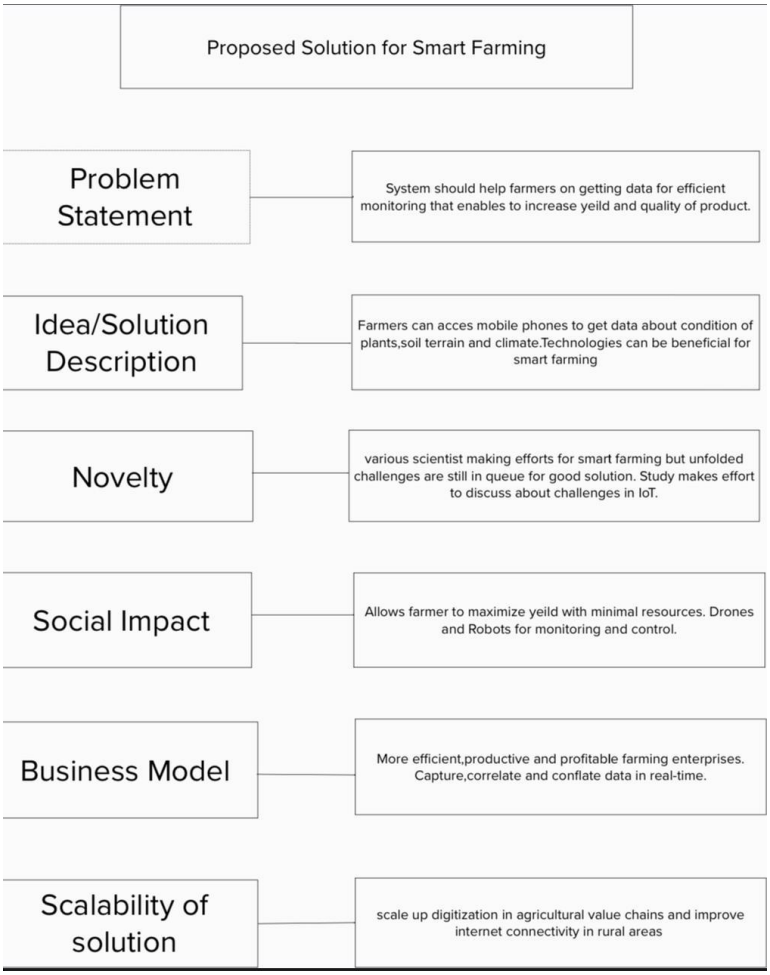
Introduction on Internet of Things (IoT), application of IoT in agricultural field to improve the yield and quality by reducing the cost is provided. The sensors which are used in the architecture are discussed briefly and the process of transmission of data from the agriculture field to the central system is explained. The proposed system advantages are included. In addition, open research issues, challenges, and future of IoT in agricultural field are highlighted. The concept is basically developed on an idea, where there are numerous things or objects - such as Arduino, sensors, GSM models, LCD display, etc., that are connected with the Internet. Each of the objects has a different address and is able to interact with other items. The things or objects co-operate with each other to reach a common goal.

We are going to construct a smart agricultural monitoring system which can collect crucial agricultural data and send it to an IoT platform called Thingspeak in real time where the data can be logged and analyzed. The logged data on Thingspeak is in graphical format, a botanist or a reasonably knowledged farmer can analyze the data (from anywhere in the world) to make sensible changes in the supplied resources (to crops) to obtain high quality yield

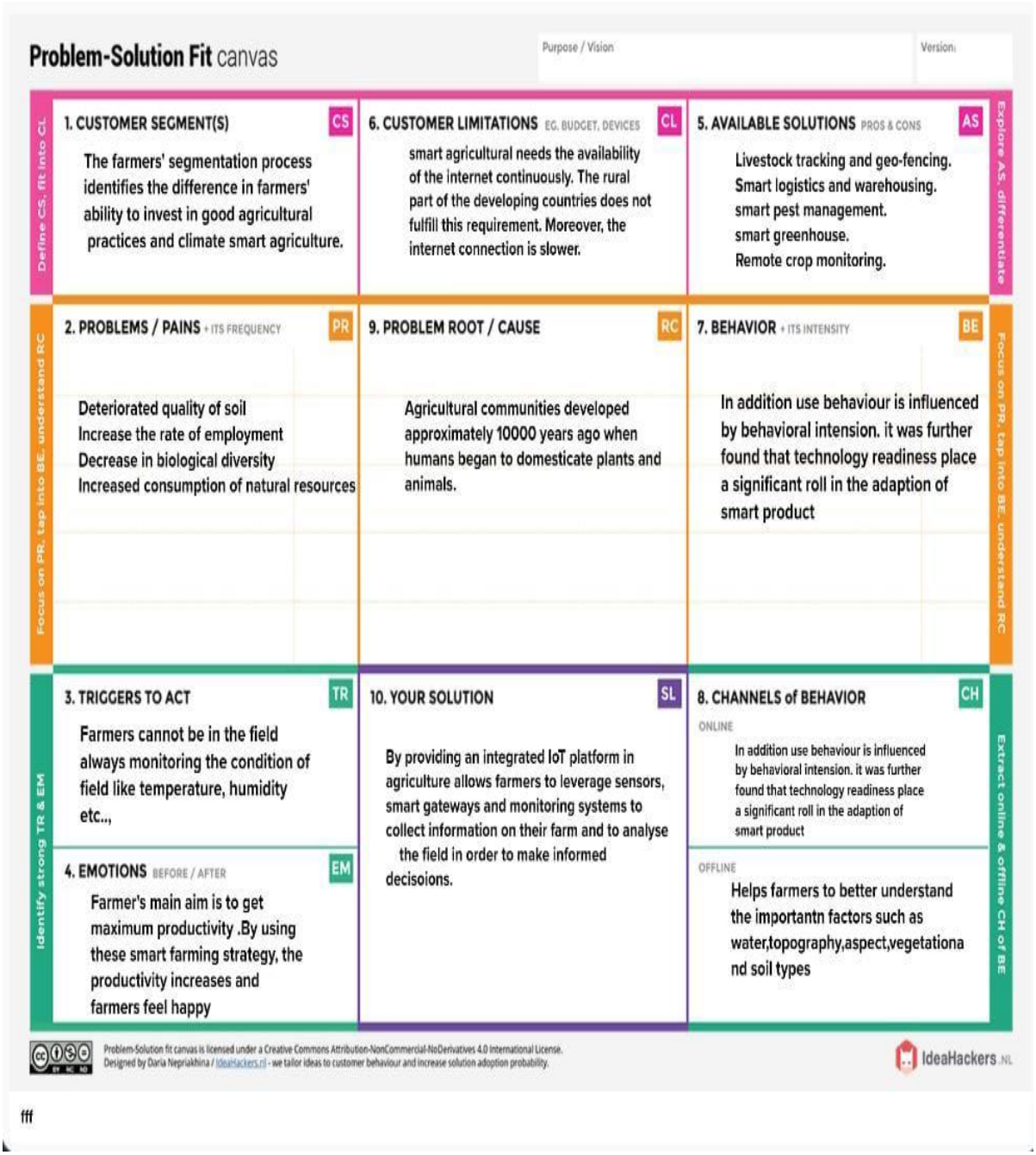
Smart agriculture monitoring system or simply smart farming is an emerging technology concept where data from several agricultural fields ranging from small to large scale and its surrounding are collected using smart electronic sensors. The collected data are analyzed by experts and local farmers to draw short term and long-term conclusion on weather pattern, soil fertility, current quality of crops, amount of water that will be required for next week to amonth etc.

We can take smart farming a step further by automating several parts of farming, for example smart irrigation and water management. We can apply predictive algorithms on microcontrollers or SoC to calculate the amount of water that will be required today for a particular agriculture field. Say, if there was rain yesterday and the quantity of water required today is going to be less. Similarly, if humidity was high the evaporation of water at upper ground level is going to be less, so water required will be less than normal, thus reducing water usage.

Proposed Solution



Problem solution fit



fff

4. Requirement Analysis

Functional Requirement

FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR NO.	FUNCTIONAL REQUIREMENTS (Epic)	SUB-REQUIREMENT (STORY/SUB-TASK)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linked IN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Certification Requirements	Regulation Rules Profession wide.
FR-4	Authorization	Healthcare provider user group.
FR-5	Business rules	Decision-making marketing.
FR-6	External interfaces	Wide Area Network (WAN) screen layouts.

NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution

FR NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
NFR-1	Usability	Use of fertilizers, Irrigation, and scheduled planting operation.
NFR-2	Security	Crops could be protected from these diseases use pesticides and biocontrol agents.
NFR-3	Reliability	Reducing deforestation, conserving natural resources and curbing soil erosion.
NRF-4	Performance	Agricultural productivity depends on the quality of the seeds with which farmers show their fields.
NFR-5	Availability	Farming methods require growers' appropriate plant protection strategy and training
NFR-6	Scalability	The application of sensors and automated irrigation practices can help monitor agricultural land.

5. PRODUCT DESIGN

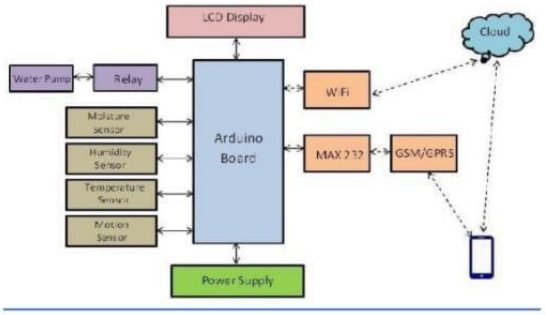
Data flow diagrams

Project Design Phase-II
Data Flow Diagram & User Stories

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.

Data flow diagram for Smart agriculture:



The feature of this system is that it can monitor moisture, Humidity, Pressure of the soil by means of Arduino board which is incorporated by IoT and in any case of discrepancy it will send a sms notification to application developed for the same to the farmer's mobile phone we can log on to them by means of providing account information to it

- Arduino Board is connected with different sensors in order to monitor temperature, humidity, pressure etc.
- Power supply is given to the board to provide sufficient power for the system.
- A relay is connected to board which in turn connected to water pump for water supply.
- A LCD display is also connected in order to display the information monitored by sensors.
- By wifi or 3G /4G services we can get information about our land through sms notification.

User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Mobile user)	Download the database	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account/ dashboard	High
	Register	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	I can receive confirmation email & click confirm	High
	Login	USN-3	As a user, I will receive confirmation email once I have registered for the application	I can register & access the dashboard with Facebook Login	Low
	Upload the image	USN-4	As a user, I must upload the image to identify the problem and works on it		Medium
Customer (Web user)	The functional requirements are same as mobile user	Same as mobile user	Same as mobile user	Same as mobile user	High when compared to mobile users

6.Guidelines:

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.

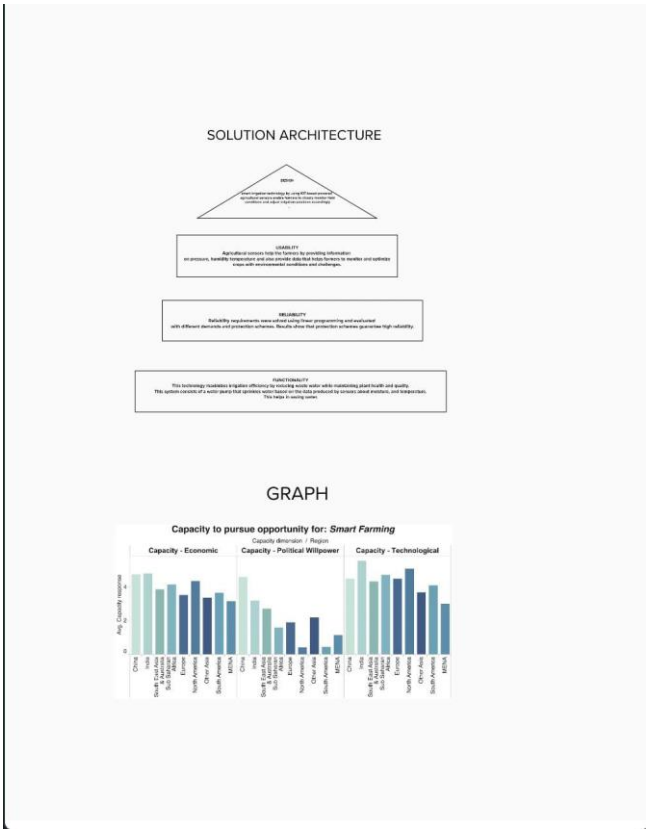
• The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.

• Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.

• NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.

• All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could plan through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.

Solution and Technical Architecture



• The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.

• Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.

• NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.

• All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could decide through an app, weather to

water the crop or not depending upon the sensor values. By using the app, they can remotely operate the motor switch.

PROJECT PLANNING AND SCHEDULING

Project Planning Phase
Sprint Delivery Plan

Date	21 October 2022
Team ID	PNT2022TMID39109
Project Name	Smart farming
Maximum Marks	8 Marks

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	Registration (Farmer Mobile User)	UNS-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Sharmila Devi N(Leader)
Sprint-1	Login	UNS-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Uthra C R(Member 2)
Sprint-2	User Interface	UNS-3	As a user, I can register for the application through Facebook	3	Low	Safrin I(Member 3)
Sprint-1	Data Visualization	UNS-4	As a user, I can register for the application through GMAIL	2	Medium	Sherlin Beaula S B (Member 4)
Sprint-3	Registration (Farmer - Web User)	USN - 1	As a user, I can log into the application by entering email and password	3	High	Rammiya R(Member 5)
Sprint - 2	Login	USN - 2	As a registered user, I need to easily login log into my registered account via the web page in minimum time	3	High	Sharmila Devi N(Leader)
Sprint - 4	Web UI	USN - 3	As a user, I need to have a friendly user interface to easily view and access the resources	3	Medium	Uthra C R(Member 2)
Sprint - 1	Registration (Chemical Manufacturer - Web user)	USN - 1	As a new user, I want to first register using my organization email and create a password for the account.	2	High	Safrin I(Member 3)

6. CODING AND SOLUTIONING

Feature

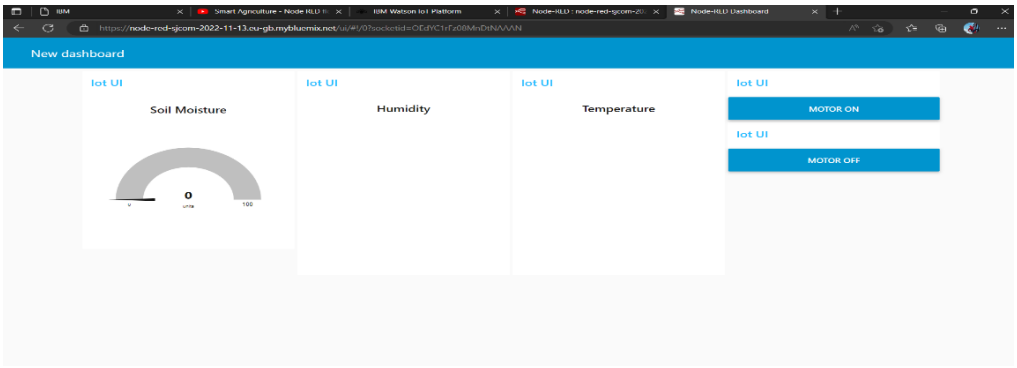
```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig ={
    "identity": {
         "orgId": "0hzydu",
         "typeId": "NodeMCU",
         "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
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 switchedon")
    elif (m=="motoroff"):
        print ("Motor is switchedOFF")
    print (" ")
while True:
    moist =random.randint (0,100)
```

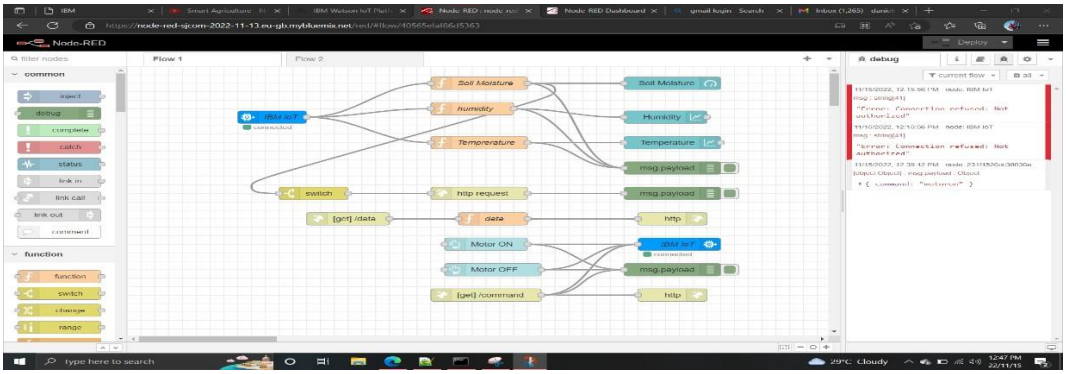
```
    temp=random.randint (-20, 125)
    hum=random.randint (0, 100)
    myData={'moisture':moist,'temperature':temp,'humidity':hum}
    client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0 ,
onPublish=None)
    print ("Published data Successfully: %s",myData)
    time.sleep (2)
    client.commandCallback =myCommandCallback
client.disconnect ()
```

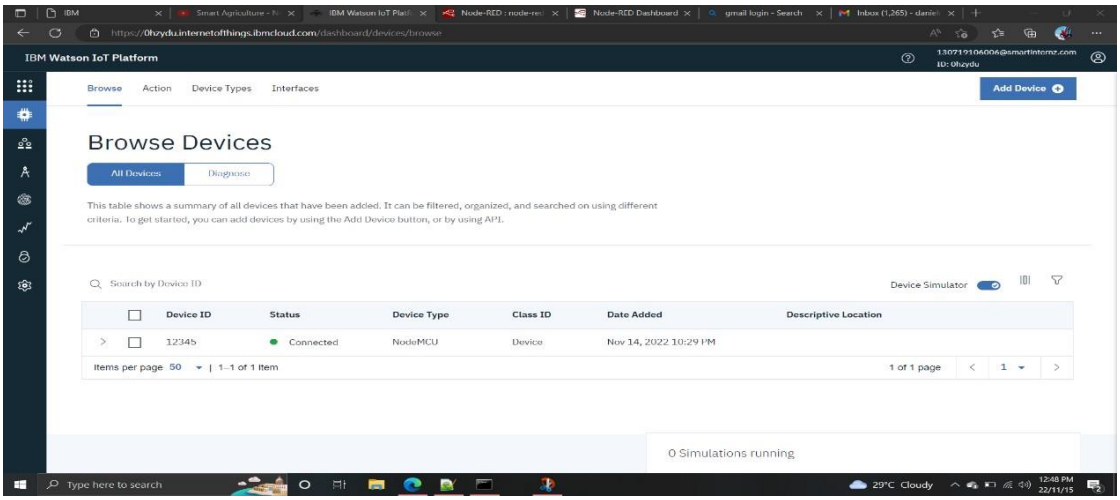
7. TESTING

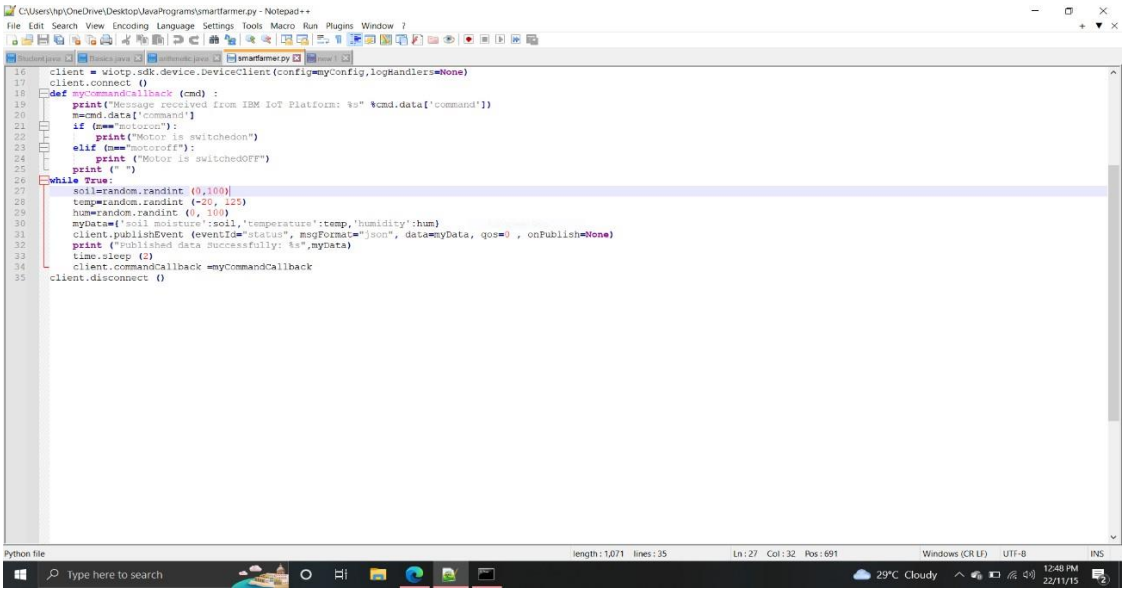
Test case

Web application using Node Red

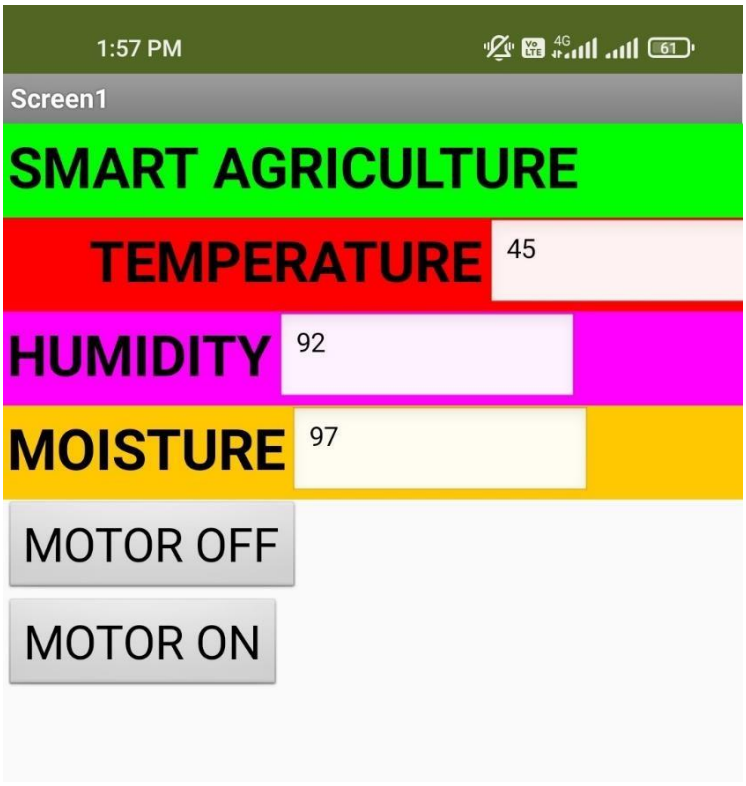





IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Browse Devices
All Devices
Diagnose
This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.
Search by Device ID
Device Simulator
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
12345
Connected
NodeMCU
Device
Nov 14, 2022 10:29 PM
Items per page 50
1–1 of 1 item
1 of 1 page
0 Simulations running
Type here to search
29°C Cloudy
12:48 PM
22/11/15

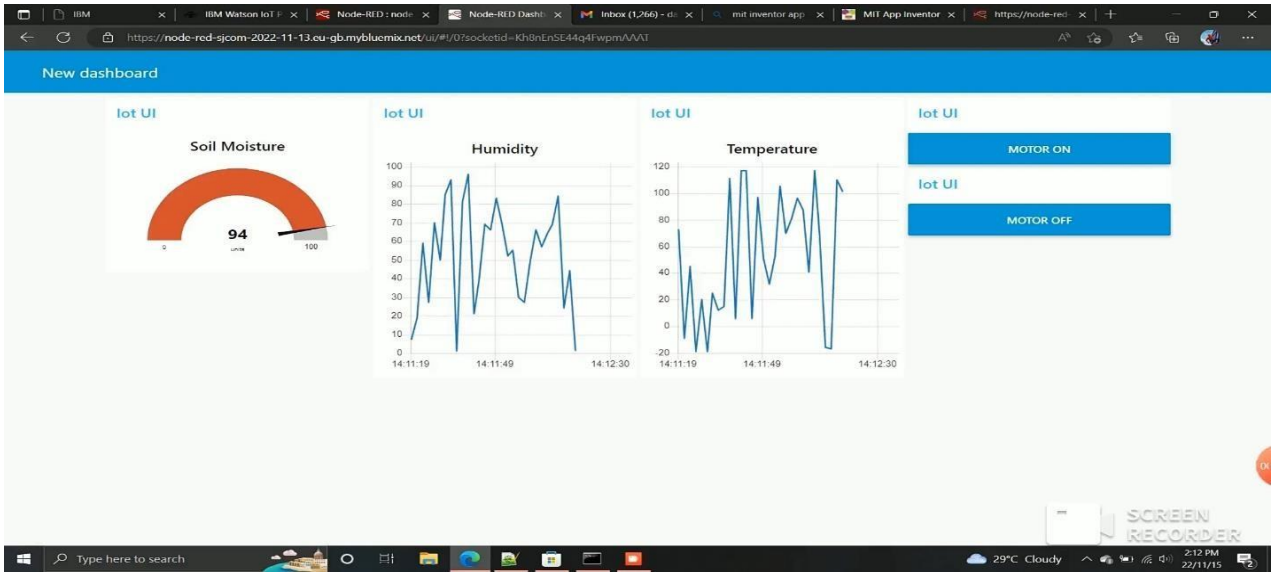

C:\Users\hp\OneDrive\Desktop\JavaPrograms\smartfarmer.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
smartfarmer.py
client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect ()
def myCommandCallback (cmd) :
print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
m=cmd.data['command']
if (m=="motoron"):
print("Motor is switchedon")
elif (m=="motoroff"):
print ("Motor is switchedOFF")
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.commandCallback =myCommandCallback
client.disconnect ()
Python file
length : 1,071 lines : 35
Ln : 27 Col : 32 Pos : 691
Windows (CR LF)
UTF-8
INS
Type here to search
29°C Cloudy
12:48 PM
22/11/15

User Acceptance Testing



8. RESULTS

Performance Metrics



9. Advantages and disadvantages

Advantages:

- A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.

- For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.

- Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.

- Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.

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 equipment 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.

10. CONCLUSION

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted dependent on the local environmental variables. This technology also detects animal invasions, which are a major cause of crop loss. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source. It helps in deciding the farmer to whether to do irrigation or not to do. Continuous internet connectivity is required for continuous monitoring of data from sensors. This also can be overcome by using GSM unit as an alternative of mobile app. By GSM, SMS can be sent to farmers phone.

11. Future scope

In the current project we have implemented the project that can protect and maintain the the crop. In this project the farmer monitor and control the field remotely. In future we can add or update few more things to this project

. • We can create few more models of the same project ,so that the farmer can have information of a entire.

• We can update the this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. It reduces the power line cost. It will be a one time investment. We can add solar fencing technology to this project.

• We can use GSM technology to this project so that the farmers can get the information directly to his home through SMS. This helps the farmer to get information if there is a internet issues.

• We can add camera feature so that the farmer can monitor his field in real time. This helps in avoiding thefts.

12. Appendix

Source Code

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig ={
    "identity": {
        "orgId": "0hzydu",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
       "token": "12345678"
    }
}
client = wiotp.sdk.device.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 switchedon")
    elif (m=="motoroff"):
        print ("Motor is switchedOFF")
    print (" ")
while True:
    moist =random.randint (0,100)
    temp=random.randint (-20, 125)
    hum=random.randint (0, 100)
```

```
    myData={'moisture':moist,'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.commandCallback =myCommandCallback
client.disconnect ()
```

Github link: **https://github.com/IBM-EPBL/IBM-Project-40278-1660627026**

Project Demo link:

https://drive.google.com/file/d/1stY808fD0f3P1-YCVmaI3MSWAZcWpXGs/view?usp=drivesdk