PROJECT REPORT

ON

Smart Farmer - IoT Enabled Smart Farming Application

Submitted by,

Team ID : PNT2022TMID42646

INDEX

6.3 Reports from JIRA

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. RESULTS

8.1 Performance Metrics

9. ADVANTAGES & DISADVANTAGES

10. CONCLUSION

11. FUTURE SCOPE

12. APPENDIX

1.Introduction:

1.1. Overview:

In Agriculture, yield depends on many factors such as seeds quality, soil type, moisture, temperature and other climatic factors. As a result, production of food-grains fluctuates year afteryear if any of factors make an impact. A year of abundant output of cereals is often followed by a year of acute shortage especially in India. Due to this problem, obtained total yield was not meeting to food requirements of people and as a result leaving many people to starvation. This has been for many years due to Traditional Agriculture was followed. In the recent years Government started many initiatives like setting soil testing labs, good quality fertilizers and seeds and modern equipment like tractors etc. and most importantly Modern Agriculture had taken shape. But still many farmers do not have any information on climatic and plant conditions before hand so that requires action can be taken care.

1.2. Purpose:

Through many scientific research, it is found that knowning before hand the climatic conditions by farmers with an easy UI (User Interface) so that they can monitor closely and perform required actions. Therefore the purpose of this project is to make a Smart Agriculture System based on Internet of Things where the dashboard can give all the agricultural conditions of crops and weather conditons, also the water pump can be toggled on/off through the same dashboard, instead of doing it manually.Also the all the climatic and crop condition information is recorded for future reference and analysis.

2.Literature survey:

2.1. Existing Problem:

The Traditional agriculture methods is still used by many farmers and though a small percentage of farmers converted into modern agriculture, majority of yield is not produced due to no easy to use system to closely monitor the crop conditions like moisture, temperature etc. Also another major issue is the unpredictable weathervbdf conditions and the farmers wholly depend on Televison broadcast which does not give real time updates.

2.2. References:

1.Dweepayan Mishra, Arzeena Khan, Rajeev Tiwari, Shuchi Devaki Upadhaye. Agriculture is a substantial source of revenue for Indians and has a huge impact on the Indian economy. Crop development is essential for enhanced yield and higher-quality delivery. As a result, crop beds with ideal conditions and appropriate moisture can have a big influence on output.Traditional irrigation systems, such as stream flows from one end to the other,are usually used. As a result of this delivery, the moisture levels in the fields can alter. A designed watering system can help to enhance the management ofthe water system. This research proposes a terrain-specific programmable water system that will save human work while simultaneously improving waterefficiency and agricultural productivity. The setup is made up of an Arduino kit, a moisture sensor, and a Wi-Fi module. Data is acquired by connecting ourexperimental system to a cloud framework. After then, cloud services analysethe data and take the necessary actions.

2.Anushree Math, Layak Ali, Pruthviraj U. India is a country where agriculture plays a vital role. As a result, it's critical to water the plants wiselyin order to maximise yield per unit space and so achieve good output. Irrigationis the process of providing a certain amount of water to plants at a specific time.The purpose of this project is to water the plants on the National Institute of Technology Karnataka campus with a smart drip irrigation system. To do this,the open source platform is used as the system's fundamental controller.Various sensors have been employed to supply the current parameters of components that impact plant healthiness on a continual basis. By controlling a solenoid valve, water is provided to the plants at regular intervals dependingon the information acquired from the RTC module. The webpage may be usedto monitor and manage the complete irrigation system. This website contains afunction that allows you to manually or automatically control plant watering.

3. H.G.C.R. Laksiri, H.A.C. Dharmagunawardhana, J.V. Wijayakulasooriya. Development of an effective loT-based smart irrigation system is also a crucial demand for farmers in the field of agriculture. This research develops a low cost, weather-based smart watering system. To begin, an effective drip irrigation system must be devised that can automatically regulate water flow to plants based on soil moisture levels. Then, to make this water-saving irrigation system even more efficient, an IoT-based communication feature is added, allowing a remote user tomonitor soil moisture conditions and manually adjust water flow. The system also includes temperature, humidity, and rain drop sensors, which have been updated to allow remote monitoring of these parameters throughthe internet. In real time, these field weather variables are stored in a remotedatabase. Finally, based on the present weather conditions, a weather prediction algorithm is employed to manage water distribution. Farmers would be able to irrigate their crops more efficiently with the proposed smart irrigation system.

2.3. Problem Statement:

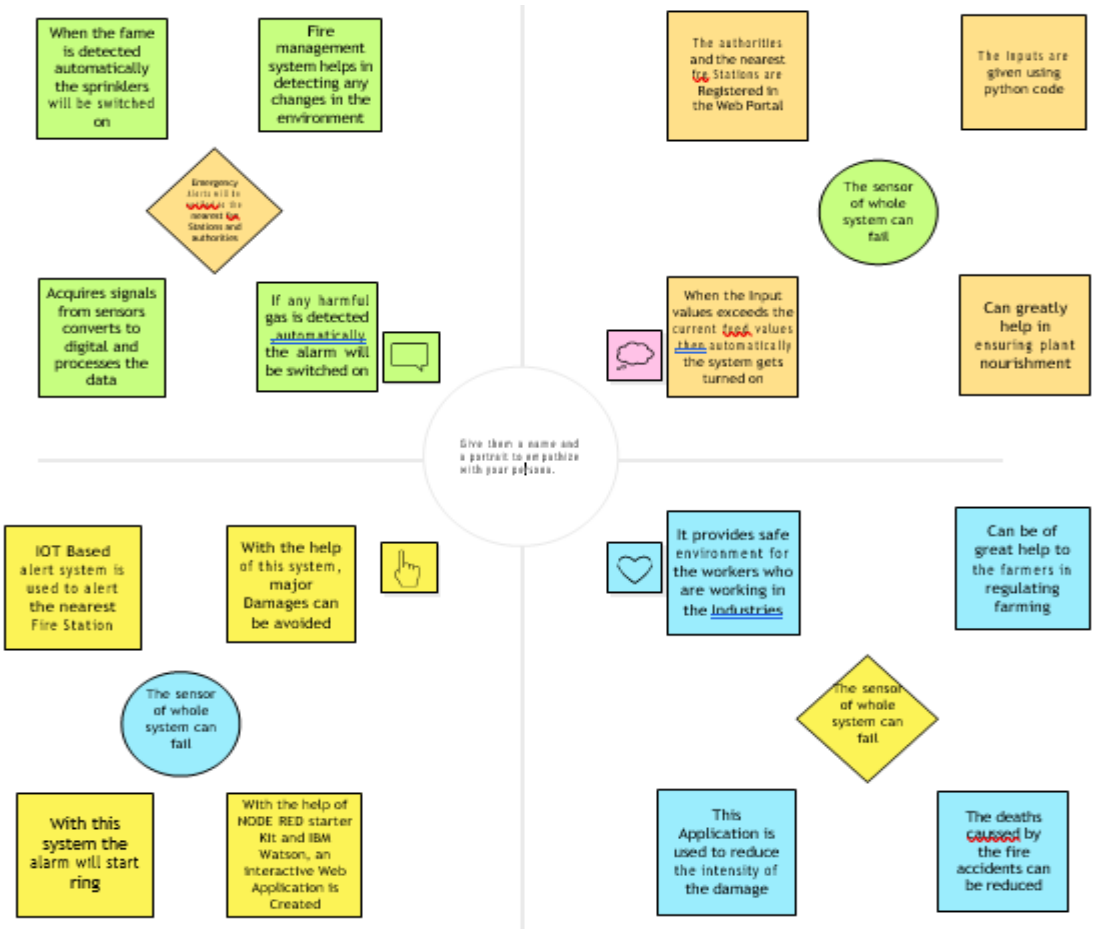
Large farm owners can benefit from IoT applications to collect data regarding the location, well-being, and health of their cattle. To provide efficient decision support system using wireless sensor network which handle different activities of farm and gives useful information related to farm field to farmer.

Problem Statement	I am (Customer)	I'm trying to	But	Because	Which makes me Feel
PS-1	Smart Farmer	To make the user handle easily and remote	They must have the knowledge to use the device	Integration of sensors analytics driving automation response activities	Being Monitored

3. Ideation & Proposed Solution:

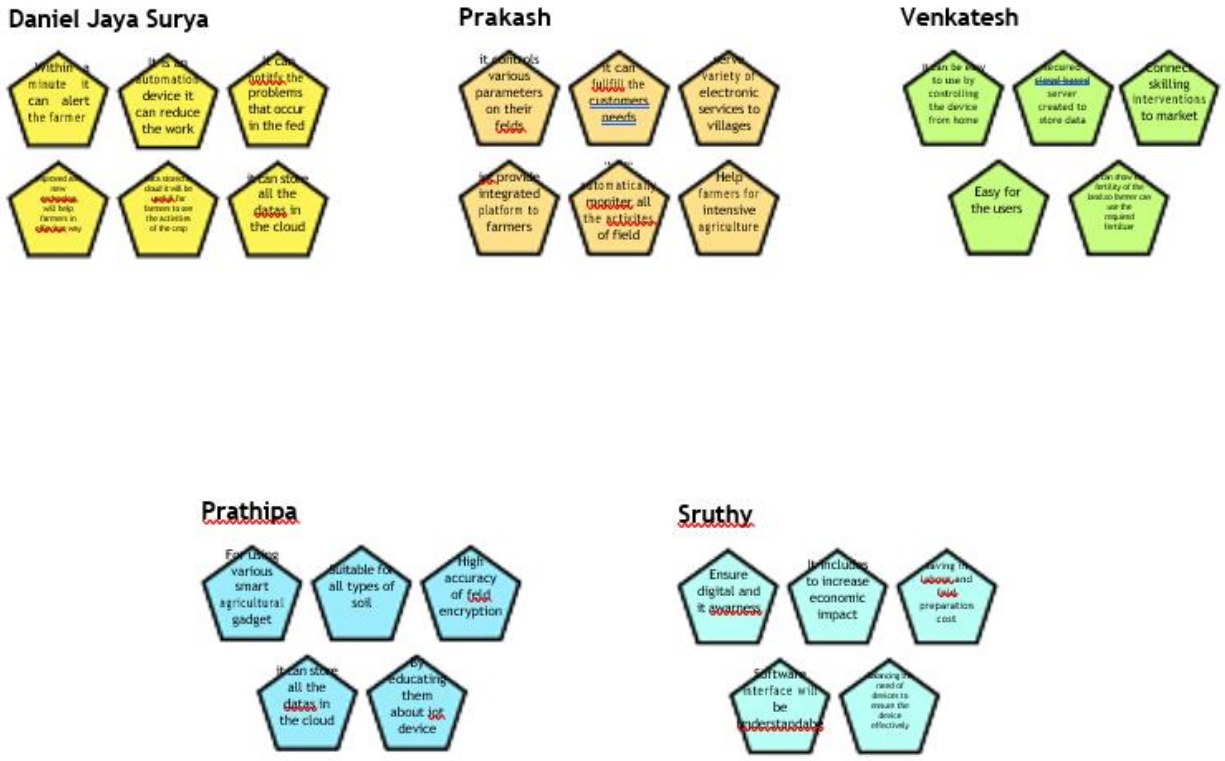
3.1 Empathy Map Canvas:

An empathy map helps to map what a design team knows about the potential audience. This tool helps to understand the reason behind some actions a user takes deeply. This tool helps build Empathy towards users and helps design teams shift focus from the device to the users who are going to use the device. As the team learns more about the users, they place that information on the chart and gain an in-depth view of the user behavior, problems, opportunities. More understanding of the Target Audience. More Organized Information in easy to understand format. Fast and Inexpensive. Easy Customization. Common Understanding and same mindset of whole team members. It describes what users **think, say, feel, do.**



3.2 Ideation:

Brainstorming is when you deliberately try to think up new ideas or solutions to problems. In writing whether creative, academic, or business it's a beneficial preliminary stage that helps writers know precisely what's going into their projects. Ideas are the most valuable resource in any communication, which makes brainstorming for writing a crucial part of the process. But for people who mostly wait around for ideas to find them, brainstorming can be quite difficult or even frustrating.

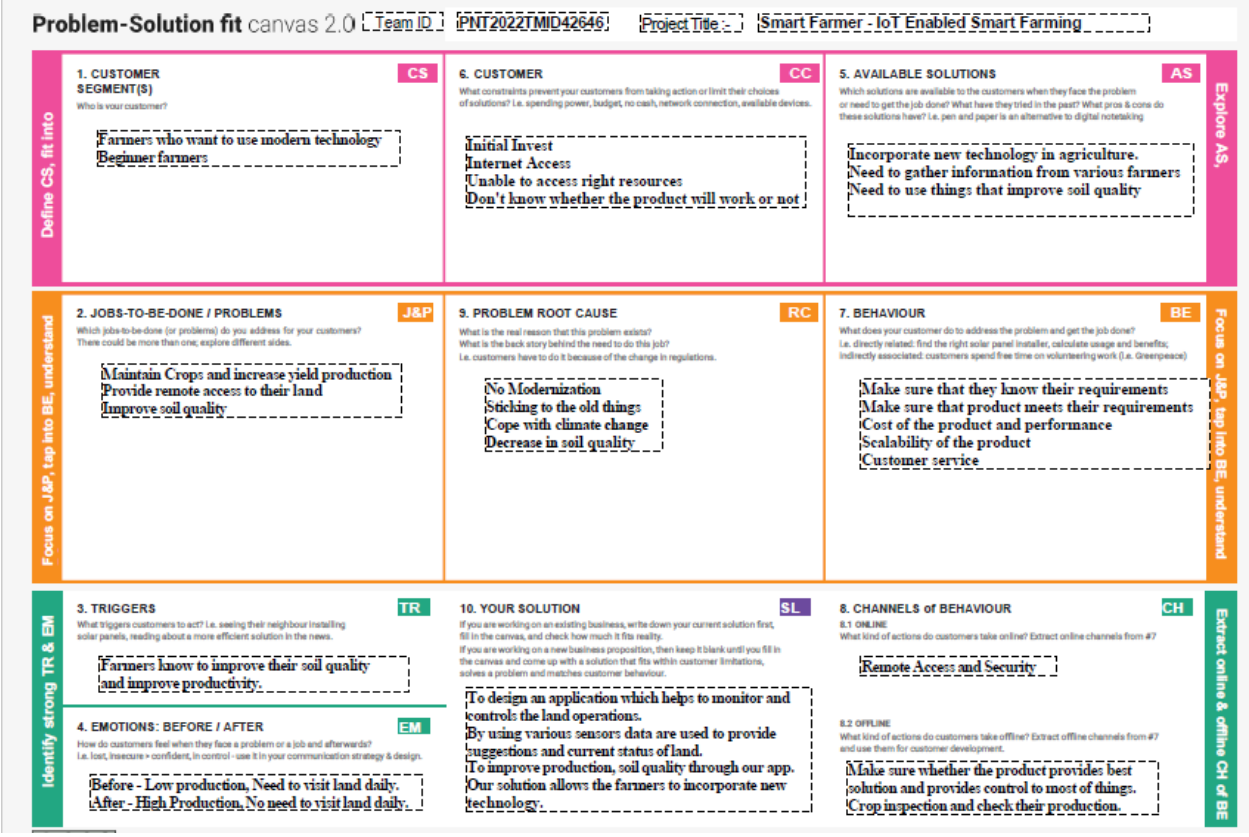


3.3. Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To solve farmer issues like • Lack of Modernization and Mechanization • Invest in farm productivity and improving yield production. • Cope with climate change, soil erosion
2.	Idea / Solution description	An application and device is introduced to know about various data about their land remotely, where they can schedule some events for a month or a day. It also provides suggestions to users based on the crop they planted.
3.	Novelty / Uniqueness	Providing suggestions, Planning events

4.	Social Impact / Customer Satisfaction	Farmers can track and control their land, suggestions of next plant crops and improving yield gives satisfaction.
5.	Business Model (Revenue Model)	• It's a subscription model, where user have to pay for their internet. □ Customer services are supported • It supports third party devices also • Reach customers via Referral, Agents, Third party applications

3.4. Problem Solution fit:



4. Requirement Analysis:

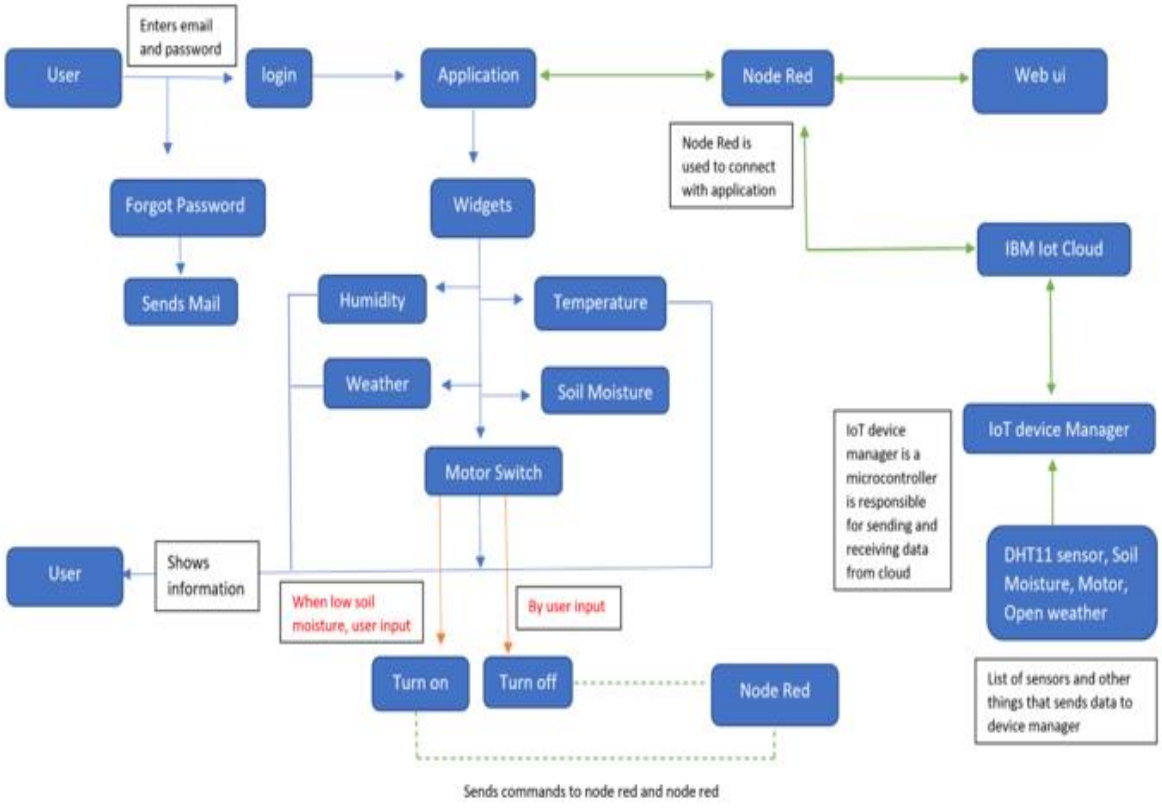
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	User Registration	Registration Through Gmail
2	User Confirmation	Confirmation Via Email Confirmation Via OTP
3	User Login	Login with Email Id and Password
4	Forgot Password	Login with Email Confirmation Of OTP
5	Query Form	Make a note of the problems and issues faced by user when using the application
6	Weather	To find the climate information of a particular area
7	Agro Note	To list of agriculture related information like how to plant, how much litres of water that plant need in a day etc
8	Sensors	To show various data from different sensors like temperature, humidity, soil moisture
9	Database Management	To show various agriculture related data are stored
10	Exit	After user checked every information, user can exit the application

FR No.	Non-Functional Requirement	Description
1	**Usability**	Effective and Easy to Use
2	**Security**	The process of protecting data from Unauthorized Access
3	**Reliability**	Consistency and Accuracy and the shared protection achieves a better trade-off between costs and reliability
4	**Performance**	Measured and estimate the performance of the Productivity
5	**Availability**	24/7 services
6	**Scalability**	Scalability is main concern for IoT platforms. It supports third party sensors. It can be easily scalable for large farming.

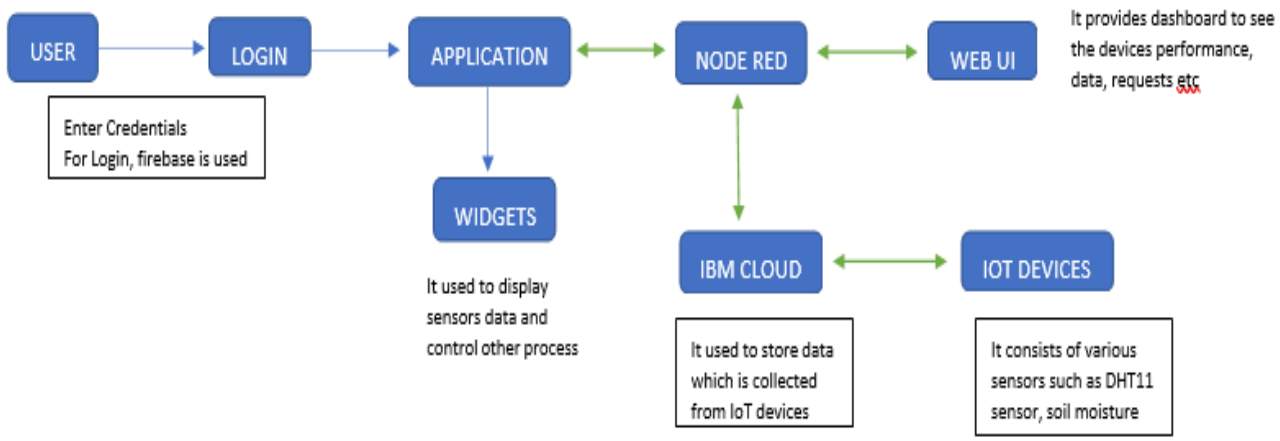
5. Project Design:

5.1. 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.



5.2. 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	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	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	Medium	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-1
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user I want to see everything in single widget		Medium	Sprint-2
		USN-7	As a user I want a organised widgets section		High	Sprint-2
		USN-8	As a user I want a graphical/pictorial representation		Low	Sprint-2
Customer (Web User)	Dashboard	USN-9	As a user I want a graphical representation of data for better understanding		High	Sprint-2
		USN-10	As a user I want to see a dashboard where I can customise myself	Dashboard with customisation	Low	Sprint-2
Customer (Mobile and Web)	IoT Device Setup	USN-10	Have to use a least sensor and get better output		High	Sprint-2

6. Project Planning & Scheduling:

6.1. Sprint Planning & Estimation:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	26 OCTOBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	28 OCTOBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	29 OCTOBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc	29 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	29 OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	30 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	30 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	30 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	30 OCTOBER 2022
Technology Architecture	Prepare the technology architecture diagram.	30 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	29 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	17 NOVEMBER 2022

6.2 Sprint Delivery Schedule:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Simulation Creation	USN-1	Connect sensors, Arduino and esp8266	2	High
Sprint-1	Software	USN-2	Develop an application with MIT App inventor (Login page with firebase)	2	High
Sprint-2	Software and Hardware	USN-3	Connect the hardware with IBM Cloud and API Integration	2	Medium
Sprint-2	Software	USN-4	Application development for project	2	High
Sprint-3	Software	USN-5	Establishing Node-Red connection	2	Medium
Sprint-3	Software	USN-6	Connecting application with Node-Red and further application development	2	High
Sprint-4	Testing	USN-7	Testing developed application and working model of hardware	2	High

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	16	5 Days	25 Oct 2022	29 Oct 2022		30 Oct 2022
Sprint-2	16	8 Days	31 Oct 2022	07 Nov 2022		08 Nov 2022
Sprint-3	16	6 Days	09 Nov 2022	13 Nov 2022		14 Nov 2022
Sprint-4	8	6 Days	15 Nov 2022	17 Nov 2022		17 Nov 2022 – 18 Nov 2022

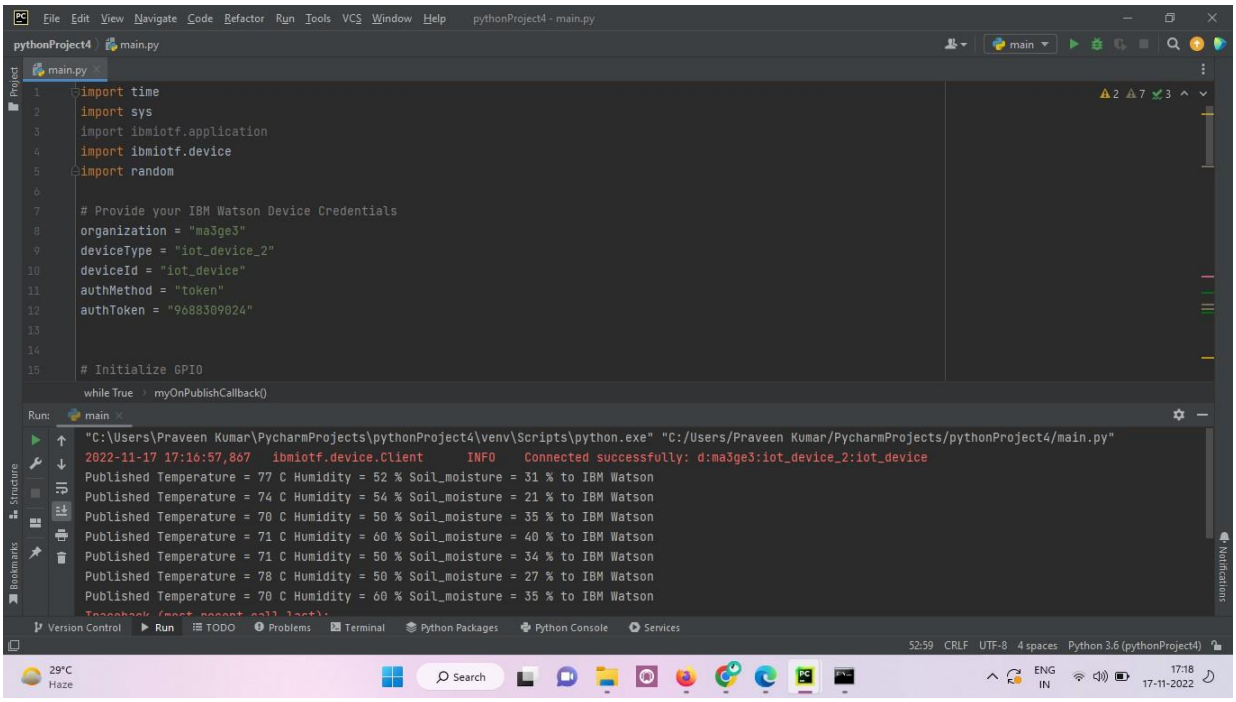
7. Coding & Solutioning:

Python Code:

```
import time
import sys

import ibmiotf.application
import ibmiotf.device import
random


# Provide your IBM Watson Device Credentials
organization = "ma3ge3"

deviceType = "iot_device_2"
deviceId = "iot_device"
authMethod = "token"
authToken = "9688309024"


# Initialize GPIO

def myCommandCallback(cmd):

        print("Command received: %s" % cmd.data['command'])status =
        cmd.data['command']

        if status == "motoron":
                print("motor is on")

        elif status == "motoroff":
                print("motor is off")

        else:

                print("please send proper command")


try:

        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-method":
authMethod,

                                        "auth-token": authToken} deviceCli =
        ibmiotf.device.Client(deviceOptions)
```

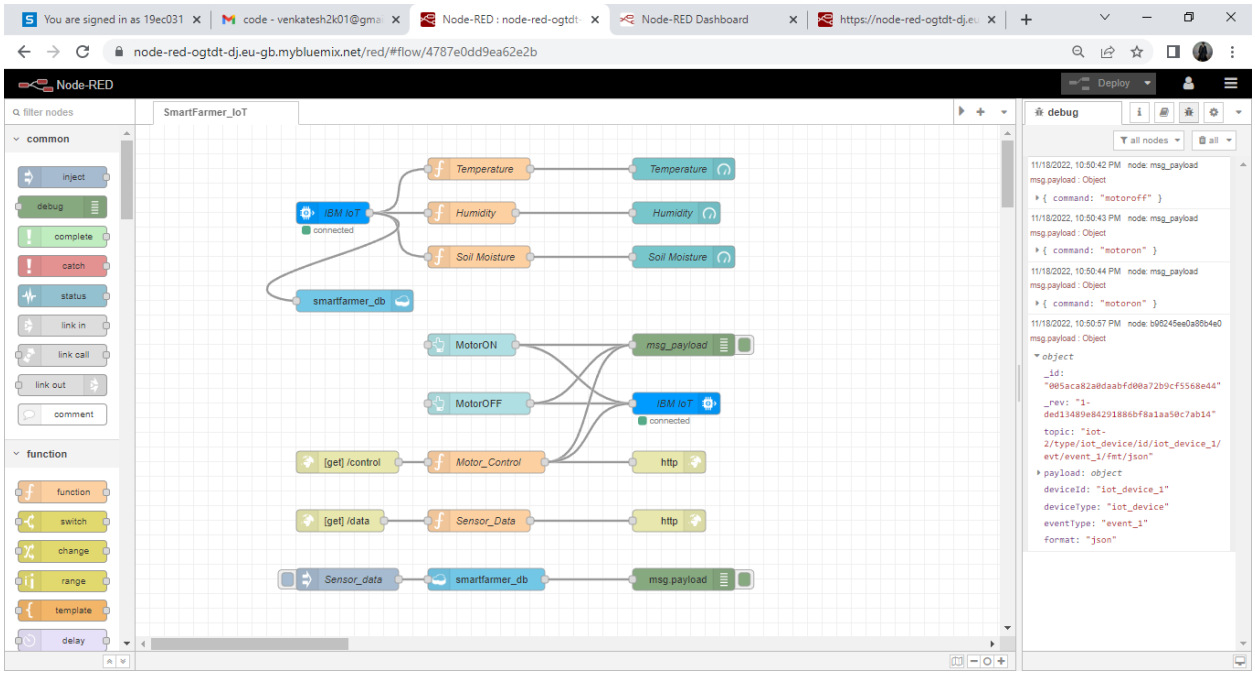
```
# ..........................................


except Exception as e:

        print("Caught exception connecting device: %s" % str(e))sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud asan event of type
"greeting" 10 times

deviceCli.connect()


while True:

        # Get Sensor Data from DHT11


        temperature = random.randint(70, 80) humidity =
        random.randint(50, 60) soil_moisture =
        random.randint(21, 40)


        data = {'temperature': temperature, 'humidity': humidity,'soil_moisture': soil_moisture}



        # print data

        def myOnPublishCallback():

                print("Published Temperature = %s C" % temperature, "Humidity = %s

%%" % humidity, "Soil_moisture = %s %%" % soil_moisture,"to IBM Watson")


        success = deviceCli.publishEvent("venkatesh_smartfarmer", "json", data,qos=0,
on_publish=myOnPublishCallback)

        if not success:

                print("Not connected to IoTF")
        time.sleep(10)


        deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

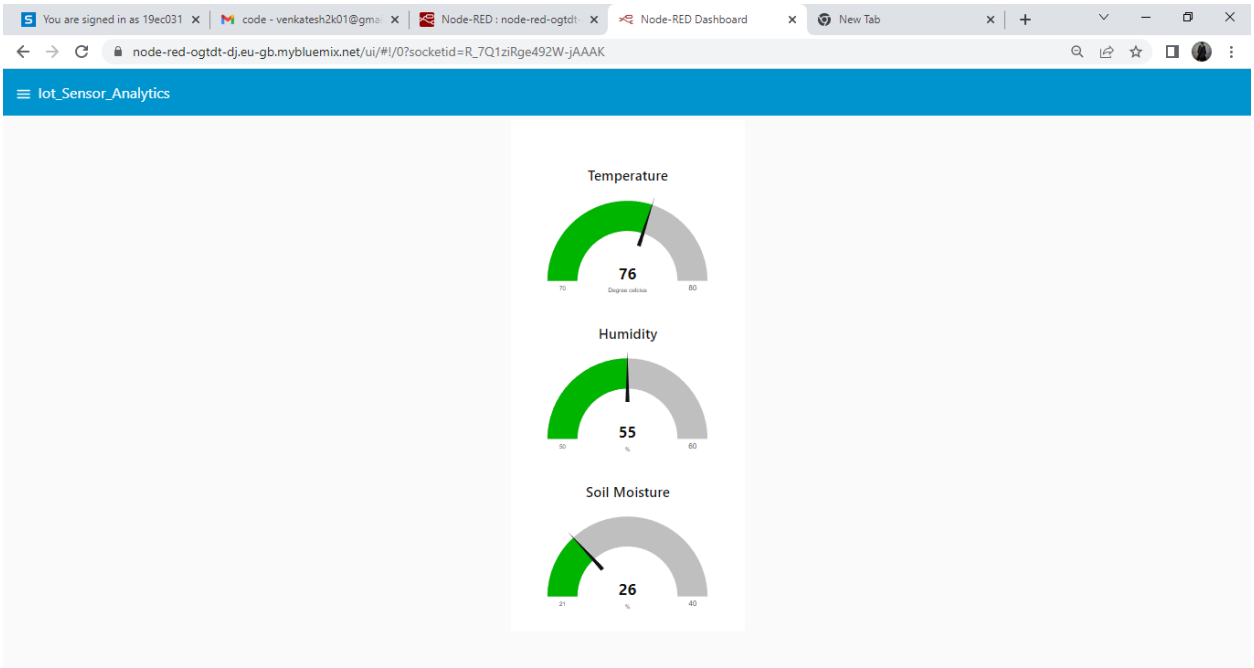
Output:

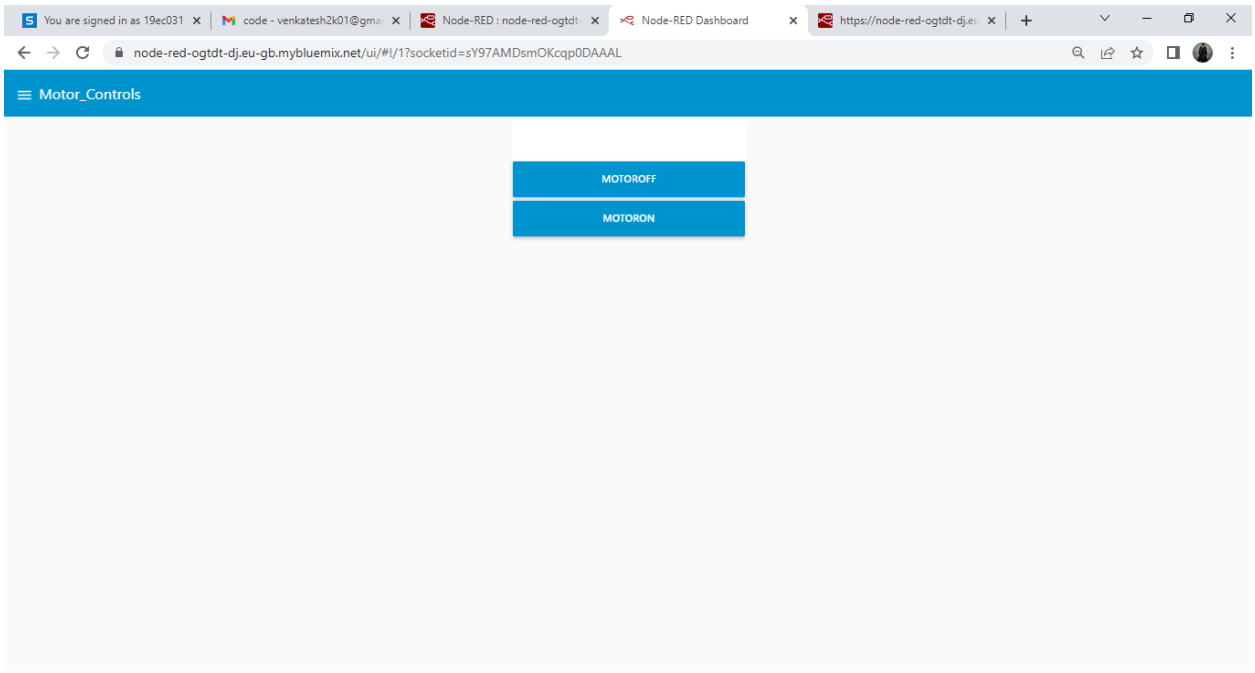

File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject4 - main.py
pythonProject4 main.py
main.py
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
# Provide your IBM Watson Device Credentials
organization = "ma3ge3"
deviceType = "iot_device_2"
deviceId = "iot_device"
authMethod = "token"
authToken = "9688309024"
# Initialize GPIO
while True myOnPublishCallback()
Run: main
"C:\Users\Praveen Kumar\PycharmProjects\pythonProject4\venv\Scripts\python.exe" "C:/Users/Praveen Kumar/PycharmProjects/pythonProject4/main.py"
2022-11-17 17:16:57,867 ibmiotf.device.Client INFO Connected successfully: d:ma3ge3:iot_device_2:iot_device
Published Temperature = 77 C Humidity = 52 % Soil_moisture = 31 % to IBM Watson
Published Temperature = 74 C Humidity = 54 % Soil_moisture = 21 % to IBM Watson
Published Temperature = 70 C Humidity = 50 % Soil_moisture = 35 % to IBM Watson
Published Temperature = 71 C Humidity = 60 % Soil_moisture = 40 % to IBM Watson
Published Temperature = 71 C Humidity = 50 % Soil_moisture = 34 % to IBM Watson
Published Temperature = 78 C Humidity = 50 % Soil_moisture = 27 % to IBM Watson
Published Temperature = 70 C Humidity = 60 % Soil_moisture = 35 % to IBM Watson
Version Control Run TODO Problems Terminal Python Packages Python Console Services
52:59 CRLF UTF-8 4 spaces Python 3.6 (pythonProject4)
29°C Haze
Search
ENG IN
17:18 17-11-2022

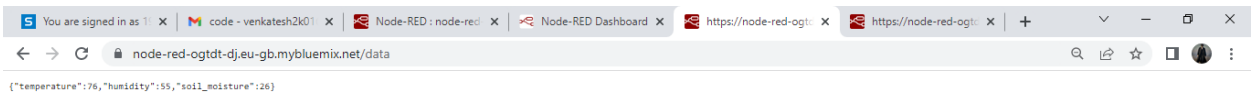
8. Results:

Simulator Data were been shown in UI and can be monitored in real time.






Node-RED Dashboard
node-red-ogtdt-dj.eu-gb.mybluemix.net/ui/#!/1?socketid=sY97AMDsmOKcqp0DAAAL
Motor_Controls
MOTOROFF
MOTORON


node-red-ogtdt-dj.eu-gb.mybluemix.net/data
{"temperature":76,"humidity":55,"soil_moisture":26}

9. Advantages & Disadvantages

Advantages:

- Remote Monitoring
- Easy To Use UI
- Data Collection
- Analysis of Data
- Remote Motor Control

Disadvantages:

- Privacy Issue
- Internet Connectivity

10. Conclusion:

Smart Agriculture System Based On Internet Of Things can deliver the farmer all the required information like temperature, humidity, soil mositure of the crop in realtime . Also instead of using manual based Motor control, the farmer can do this remotely anywhere aslong as he's connected to network. To make this possible we have used IBM Cloud Platform, Watson Iot Platform, By using a Python Script we were able to subscribe to IBM platform to send andreceive commands to motor for controlling it. Using this Smart Agriculture System the farmer can not only monitor all the required data in realtime but also can make smart decisions for better yield based on the data collected.In this way he can produce yield effectively and also earn profitably more based on accurate data received.

11. Future Scope

Future scope of this smart agriculture system will be to add more sensors to the existing micro controller, to add increase the current functionality or to do more automated tasks like automatic watering system, adding pest control information and geotagging the farm etc. This information can be shared on consent to Government authorities or Private companies for more suggestions of better techniques remotely. As the data stored can be used for reefernce and analysis which can be very helpful in future.

12. Appendix:

12.1. Source Code:

Python Code:

```
import time
import sys

import ibmiotf.application
import ibmiotf.device import
random


# Provide your IBM Watson Device Credentials
organization = "ma3ge3"

deviceType = "iot_device_2"
deviceId = "iot_device"
authMethod = "token"
authToken = "9688309024"



# Initialize GPIO

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])status =
    cmd.data['command']
```

```
        if status == "motoron":
                print("motor is on")

        elif status == "motoroff":
                print("motor is off")

        else:

                print("please send proper command")


try:

        deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-method":
authMethod,

                                        "auth-token": authToken} deviceCli =
        ibmiotf.device.Client(deviceOptions)

# .....


except Exception as e:

        print("Caught exception connecting device: %s" % str(e))sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud asan event of type
"greeting" 10 times

deviceCli.connect()


while True:

        # Get Sensor Data from DHT11


        temperature = random.randint(70, 80) humidity =
        random.randint(50, 60) soil_moisture =
        random.randint(21, 40)


        data = {'temperature': temperature, 'humidity': humidity,'soil_moisture': soil_moisture}



        # print data

        def myOnPublishCallback():

                print("Published Temperature = %s C" % temperature, "Humidity = %s
```

```
%%" % humidity, "Soil_moisture = %s %%" % soil_moisture,"to IBM Watson")


        success = deviceCli.publishEvent("venkatesh_smartfarmer", "json", data,qos=0,
on_publish=myOnPublishCallback)

        if not success:

                print("Not connected to IoTF")
        time.sleep(10)


        deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud

deviceCli.disconnect()
```