PROJECT REPORT

SMART FARMER - IoT ENABLED SMART FARMING

TEAM ID - PNT2022TMID20449

TEAM LEADER - KARTHIKA A
TEAM MEMBER 1 - KAVIYA G
TEAM MEMBER 2 - SATHYA HARIKARA SANKAR S
TEAM MEMEBR 3 - SURYA R

CHAPTER 1 - INTRODUCTION

IoT - based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and 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. IoT in agriculture uses robots, drones, remote sensors, and computer imaging combined with continuously progressing machine learning and analytical tools for monitoring crops, surveying, and mapping the fields, and providing data to farmers for rational farm management plans to save both time and money. IoT based Smart Farming improves the entire Agriculture system by monitoring the field in real-time. With the help of sensors and interconnectivity, the Internet of Things in Agriculture has not only saved the time of the farmers but has also reduced the extravagant use of resources such as Water and Electricity.

1.1) PROJECT REVIEW

Connecting IoT devices to the Watson IoT platform and exchanging the sensor data. Explore python client libraries of Watson IoT Platform. Gain knowledge on IBM Cloudant DB Configuring APIs using Node-RED for communicating with a mobile application. Creating a Mobile Application through which the user interacts with the IoT device. The parameters like temperature, humidity, and soil moisture are updated to the Watson IoT platform. The device will subscribe to the commands from the mobile application and control the motors accordingly APIs are developed using Node-RED service for communicating with Mobile Application A mobile application is developed using the MIT App inventor to monitor the sensor parameters and control the motors. Developed a python script to publish and subscribe to the IBM IoT platform Configure the Node-RED and create APIs for communicating with mobile application. Develop a mobile application to display the sensor parameters and control the motors.

1.2) PURPOSE

Smart farming based on IoT technologies enables farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made, and enabling efficient utilization of resources such as water, electricity, etc.

The farmers can monitor the field conditions from anywhere**.** They can also select between manual and automated options for taking necessary actions based on this data. For example, if the soil moisture level decreases, the farmer can deploy sensors to start the irrigation. Smart farming is highly efficient when compared with the conventional approach.

CHAPTER 2 - LITERATURE SURVEY

SURVEY REPORT – 1:

Author : Vu Khanh Quy , Nguyen Van Hau , Dang Van Anh , Nguyen Minh Quy , Nguyen Tien Ban , Stefania Lanza , Giovanni Randazzo 4and Anselme Muzirafuti.

Title : IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges.

The IoT integrates a series of existing state-of-the-art solutions and technologies, such as wireless sensor networks, cognitive radio ad hoc networks, cloud computing, big data, and end-user applications. This study presents a survey of IoT solutions and demonstrates how IoT can be integrated into the smart agriculture sector. To achieve this objective, we discuss the vision of IoT-enabled smart agriculture ecosystems by evaluating their architecture (IoT devices, communication technologies, big data storage, and processing), their applications, and research timeline.

SURVEY REPORT – 2:

Author : Dimitrios Glaroudis, Athanasios Iossifides, Periklis Chatzimisios.

Title : Survey, Comparison and Research Challenges of IoT Application Protocols for Smart Farming.

In this context, the Internet of Things (IoT) technologies have become the major path forward towards novel farming practices. 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.

SURVEY REPORT – 3:

Author : Jash Doshi, Tirthkumar Patel, Santosh kumar Bharti

Title : Smart Farming using IoT, a solution for optimally monitoring Farming conditions.

Internet of Things (IoT) is present and future of every field impacting everyone’s life by making everything intelligent. It is a network of different devices which make a self-configuring network. The new developments of Smart Farming with use of IoT, by farmers and reducing crop wastage. The aim is to propose a technology which can generate messages on different platforms to notify farmers.

SURVEY REPORT – 4 ;

Author : Chandini. K

Title: A Literature Study on Agricultural Production System Using IoT as Inclusive Technology.

The IoT (Internet of Things) based agricultural convergence technology is a technology to create a high value such as improvement of production efficiency, quality increase of agricultural products in the whole process of agricultural production. In addition,implementing agriculture, which is an alternative to the future agriculture, through the convergence technology allows prediction of supply and demand, realtime management and quality maintenance during the entire life cycle of agricultural products we make a literature study on the cited title and present it in the form of this note.

2.1 EXSITING PROBLEM

The most common problem for the Internet of Things in agriculture is connectivity. Every area doesn't have proper internet connectivity.The second most common challenge for Internet of Things based Advanced Farming is the lack of awareness among consumers. Due to various service providers, it becomes really difficult to maintain interoperability between different IoT systems. A scalable solution that can be integrated with thousands of IoT devices for large farms.

2.2) REFERENCES

1) Nurzaman Ahmed, D. D. (2019). Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet of Things Journal;2018;5;6;10.1109/JIOT.2018.2879579*.

2) Brian Gilmore,"The Next Step in Internet Evolution: The Internet of Things", Internet of Things, cmswire, Jan 2014.

3) Pradyumna Gokhale, Omkar Bhat, Sagar Bhat,"Introduction to IOT", International Advanced Research Journal in Science, Engineering and Technology (IARJ SET), Vol. 5, Issue 1, January 2018.

4) A.Anusha, A.Guptha, G.Sivanageswar Rao, Ravi Kumar Tenali, “A Model for Smart Agriculture Using IOT”, International Journal of Innovative Technology and Exploring Engineering (IJITEE),ISSN: 2278-3075, Volume-8 Issue-6, April 2019.

5)Muthunoori Naresh, P Munaswamy,” Smart Agriculture System using IoT Technology”, International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-7 Issue-5, January 2019.

6)Nikesh Gondchawar, Prof. Dr. R. S. Kawitkar, ”IOT based smart agriculture”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 6, June 2016.

7) Anand Nayyar, Er. Vikram Puri,” Smart Farming: IoT Based Smart Sensors Agriculture Stick for Live Temperature and Moisture Monitoring using Arduino, Cloud Computing & Solar Technology”, November 2016.

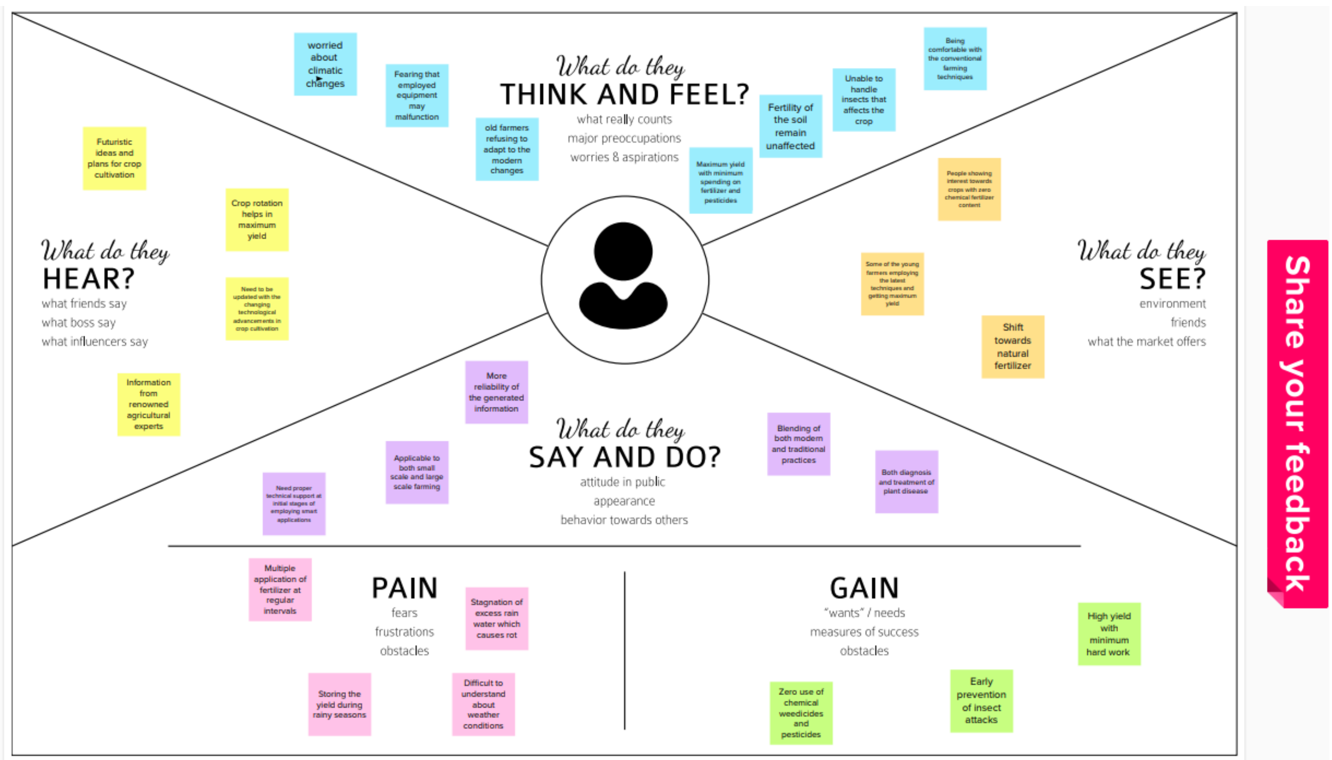
8)Sweksha Goyal, Unnathi Mundra, Prof. Sahana Shetty,” SMART AGRICULTURE USING IOT”, International Journal of Computer Science and Mobile Computing, Vol.8 Issue.5, pg. 143-148, May 2019.

2.3) PROBLEM STATEMENT DEFINITON

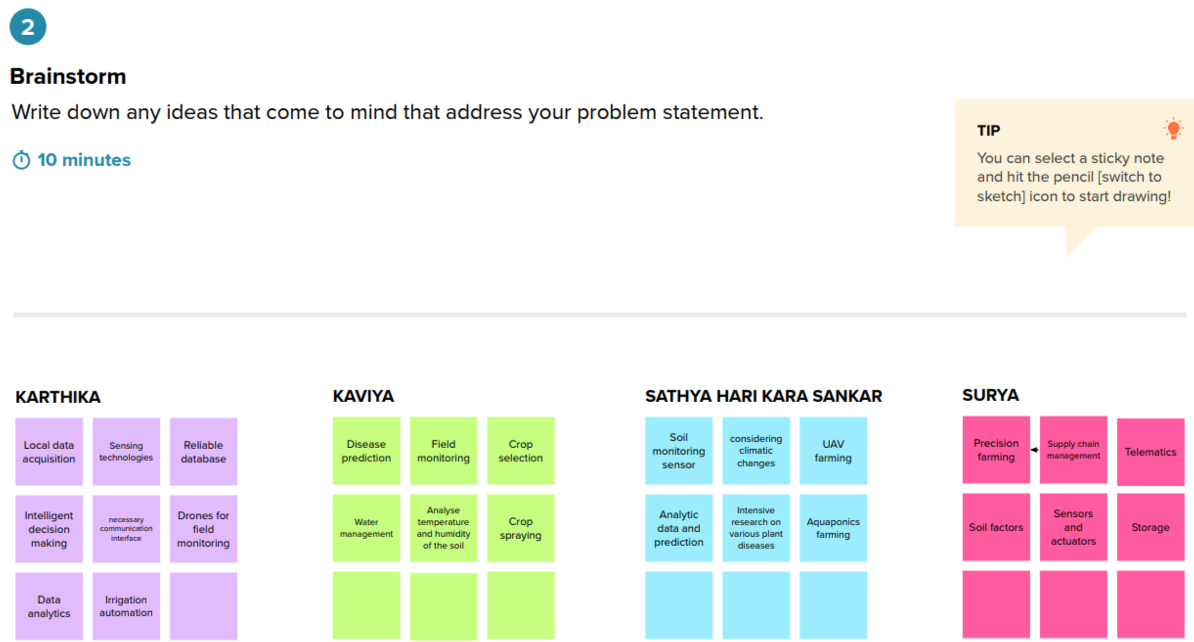
Farmers are under pressure to produce more food and use less energy and water in the process. A remote monitoring and control system will help farmers deal effectively with these pressures. Irrigated farms typically deploy a single pump to irrigate 80 to 100 acres of land. Many large farms, therefore, require 40 to 80 or more irrigation pumps spread over hundreds of square miles. Most are pumping ground water for irrigation purposes, most operate in remote fields, and trucks must roll to tend to them. Ideally, each field should get just the right amount of water at just the right time. Under-watering causes crop stress and yield reduction. Overwatering can also cause yield reduction and consumes more water and fuel than necessary and leads to soil erosion and fertilizer, herbicide, and pesticide runoff. Agricultural operations waste 60% of water consumed each year. Now more than ever, new technologies for water conservation must be adopted. According to U.S. government statistics, however, only 10% of irrigated farms use advanced water management decision tools, including precision irrigation controls and soil moisture/water level sensing devices. Case studies have shown precision irrigation has a 5%–8% impact on yield and a similar impact on operating costs. Smart Farm's systems can be retrofitted on existing sites and provide immediate impact with a very short return on investment time period.

CHAPTER 3 - IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 IDEATION AND BRAINSTORMING :

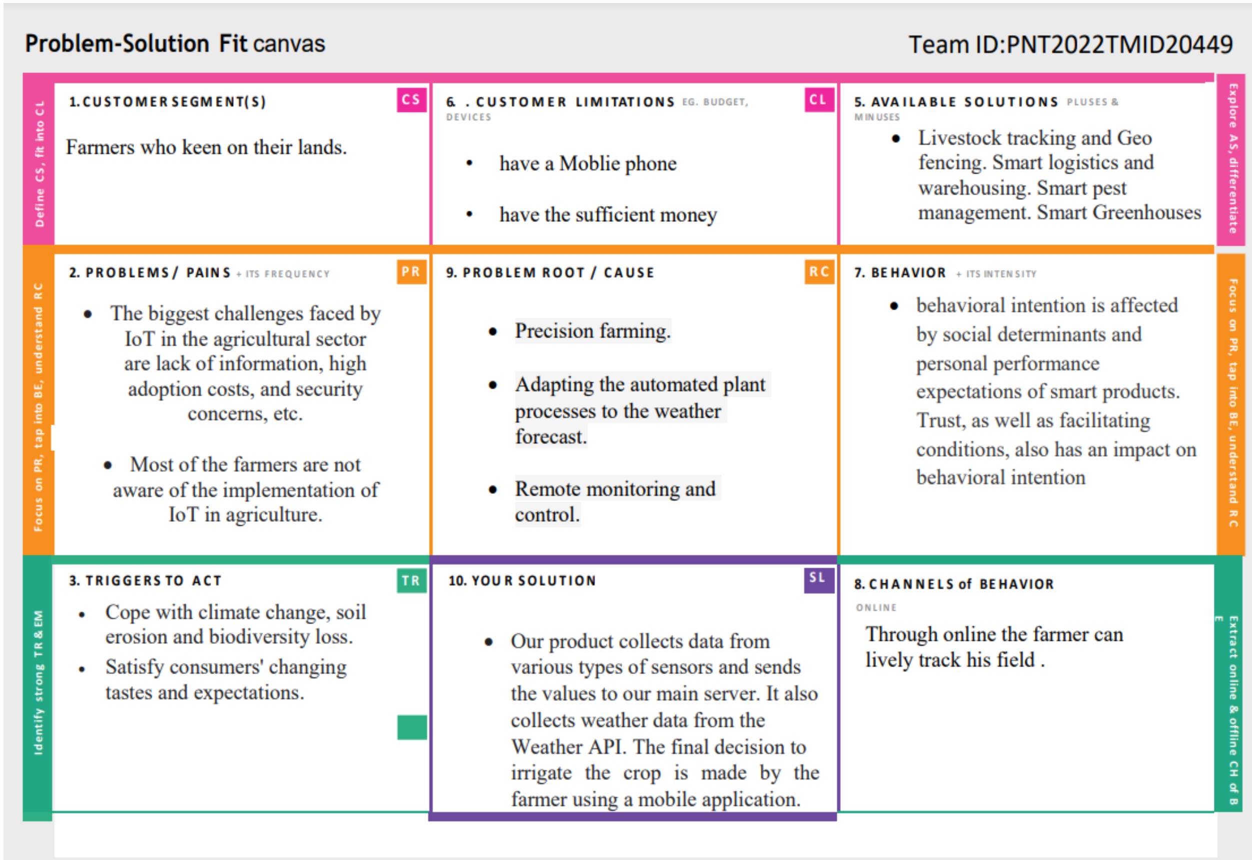


3.3 PROPOSED SOLUTION:

S.No	Parameter	Description
1.	Problem Statement	Our project will be give the problem statement in Smart farming application using IOT.History based soilhealth parameters like soil moisture,pH level, temperature etc.
2.	Idea/Solution description	The most frequently used applications of lot in agriculture are drones for monitoring fieldsandspraying crops,health assessment of livestock and irrigation
3.	Novelty/Uniqueness	Smart farming, which involves the applicationof sensors and automated irrigation practices,can help monitor agricultural land, temperature, soil moisture, etc. This would enable farmers to monitor cropsfrom anywhere
4.	Social Impact / CustomerSatisfaction	Increased production: the optimisation of all the processes related to agriculture and livestock-rearing increases production rates. Water saving: weather forecasts and sensors that measuresoil moisture meanwatering only when necessary and for the right lengthof time
5.	Business Model(Revenue Model)	Climate-smart agriculture is a pathway towardsdevelopment and food security built on three pillars: increasing productivity and incomes, enhancing resilience of livelihoods and ecosystems and reducing and

		removing greenhouse gasemissions from the atmosphere
6.	Scalability of the Solution	Smart Farmingsystems uses moderntechnology to increase the quantity and qualityof agricultural products. Livestock tracking and Geofencing. Smart logistics and warehousing.Smartpest management. SmartGreenhouses

3.4 PROBLEM SOLUTION FIT:



CHAPTER 4 - REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	IoTdevices	Sensors andWifi module.
FR-2	Software	Web UI, Node-red, IBM Watson, MIT app

4.2 NON-FUNCTIONAL REQUIREMENT :

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Consumes less time,although the productivityremains high
NFR-2	**Security**	It has low level of security features due to integration of sensor data.
NFR-3	**Reliability**	Data obtained is accurate and itis completelyreliable
NFR-4	**Performance**	Itgives high performance and high productivity
NFR-5	**Availability**	With the minimum network connectivity also, theapplication is absolutely accessible
NFR-6	**Scalability**	It is scalable in the sensethat additional constraints can be addedin the futureif the situation demands

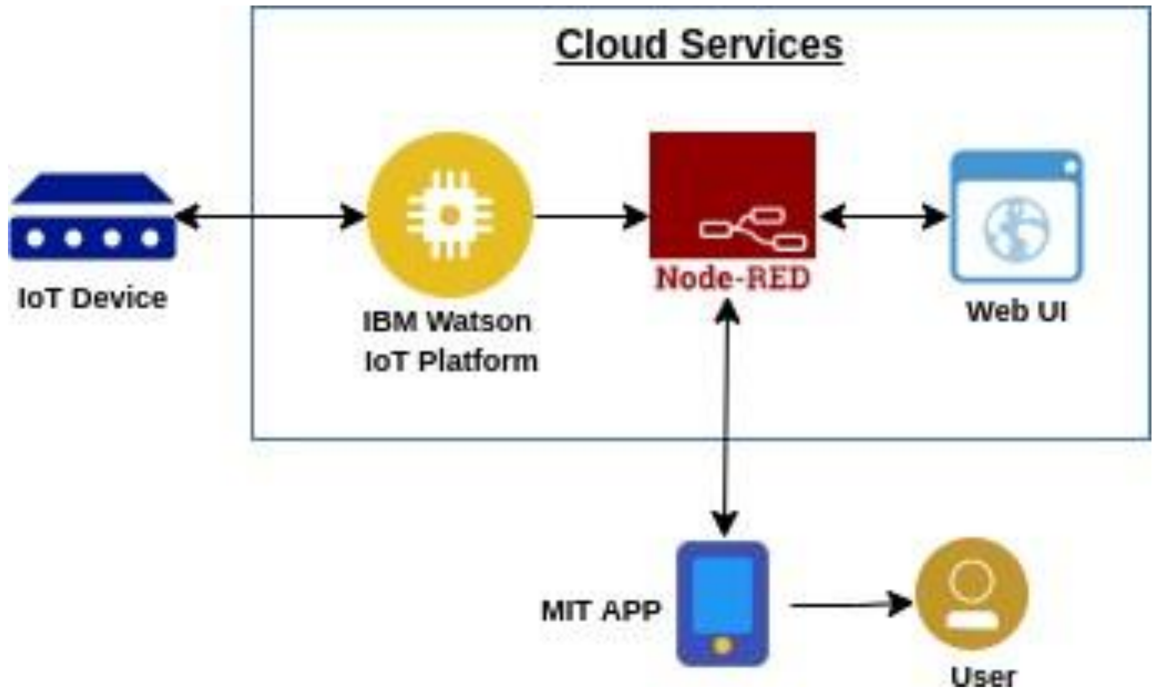
CHAPTER 5 - PROJECT DESIGN

5.1 Data Flow Diagrams :

A Data Flow Diagram (DFD)is a traditional visual representation of the informationflows 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.

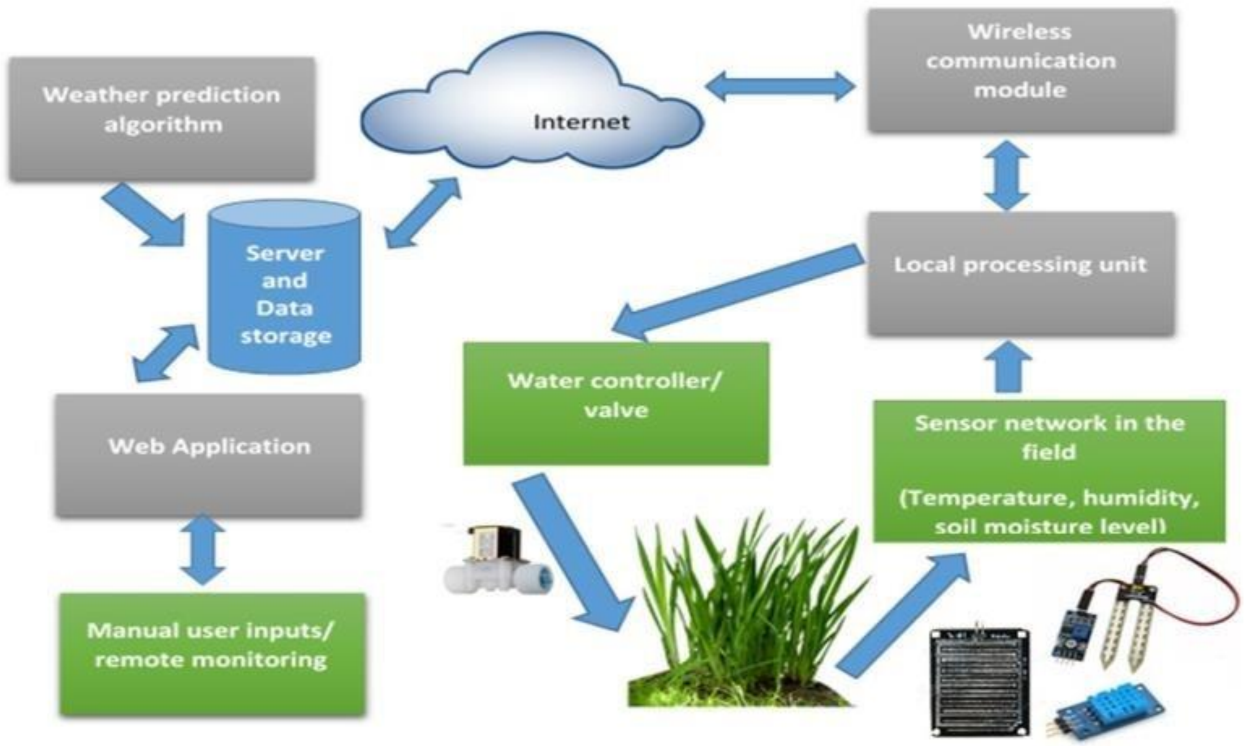
Example :

Iii



DFD LEVEL 0:

lii



5.2 Solution & Technical Architecture :

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts withapplication e.g. Web UI, Mobile App, Chat botetc.	MIT app
2.	Application Logic-1	Logic for a process in theapplication	Node red/IBM Watson/MIT app
3.	Application Logic-2	Logic for a process in theapplication	Node red/IBM Watson/MIT app

4.	Application Logic-3	Logic for a process in theapplication	Node red/IBM Watson/MIT app
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL,etc.
6.	Cloud Database	Database Service on Cloud	IBM cloud.
7.	Temperature sensor	Monitors thetemperature	
8.	Humidity sensor	Monitors the humidity	
9.	Soil moisture sensor	Monitors the soil temperature	
10	Weather sensor	Monitors the weather	.
11	PH sensor	Monitors the PH of the soil	.
12	RTCmodule	Date and time configuration	
13	Relay	Toget the soil moisture data	
14	External API 1	To get the real time values of Humidity, Temperature and Moisture.	Open Weather API.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology

1.	Open-Source Frameworks	MIT app, Node-Red	Software
2.	Scalable Architecture	Drone technology, pesticide monitoring ,Mineral identification in soil	Hardware

5.3 User Stories :

User Type	Functional Requirement	User StoryNumber	User Story/Task	Acceptancecriteria	Priority	Release
Customer	IoT devices	USN-1	Sensorsand wi-fimodule		High	Sprint-1
Customer	Software	USN-2	IBM Watson IoT platform, Workflows for IoT scenarios using Node-red		High	Sprint-2
Customer	MIT app	USN-3	To developan application using MIT		High	Sprint-3
Customer	Web UI	USN-4	To make the user tointeract with the software.	User can access the app for the services.	High	Sprint-4

Administrator	MIT app	USN 5	To make necessary updates and inform the customer about the same.	User can give his feedback and also difficulties that he encounters	High	Sprint-4

CHAPTER 6 - PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation :

Iii


SPRINT PLAN
1. Identify the Problem
2. Prepare an abstract and a problem statement
3. List the requirements needed
4. Create a Code and Run
5. Make a Prototype
6. Test the created code and check with the designed prototype
7. Solution for the problem is found !!

Sprint Deliveryand Schedule

The Sprint schedule is as follows:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint 1	Simulation creation	USN-1	Connect sensors and arduino with python code	2	High	Karthika Kaviya Sathya hari kara sankar surya
Sprint 2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High	Karthika Kaviya Sathya hari kara sankar surya
Sprint 3	MIT A pp inventor	USN-3	Develop an application for the Smart farmers using MIT App inventor	2	High	Karthika Kaviya Sathya hari kara sankar surya
Sprint 3	Dashboard	USN-3	Design the module and test the app	2	High	Karthika Kaviya Sathya hari kara sankar surya
Sprint 4	Web UI	USN-4	To make the user interact with the software	2	High	Karthika Kaviya Sathya hari kara sankar surya

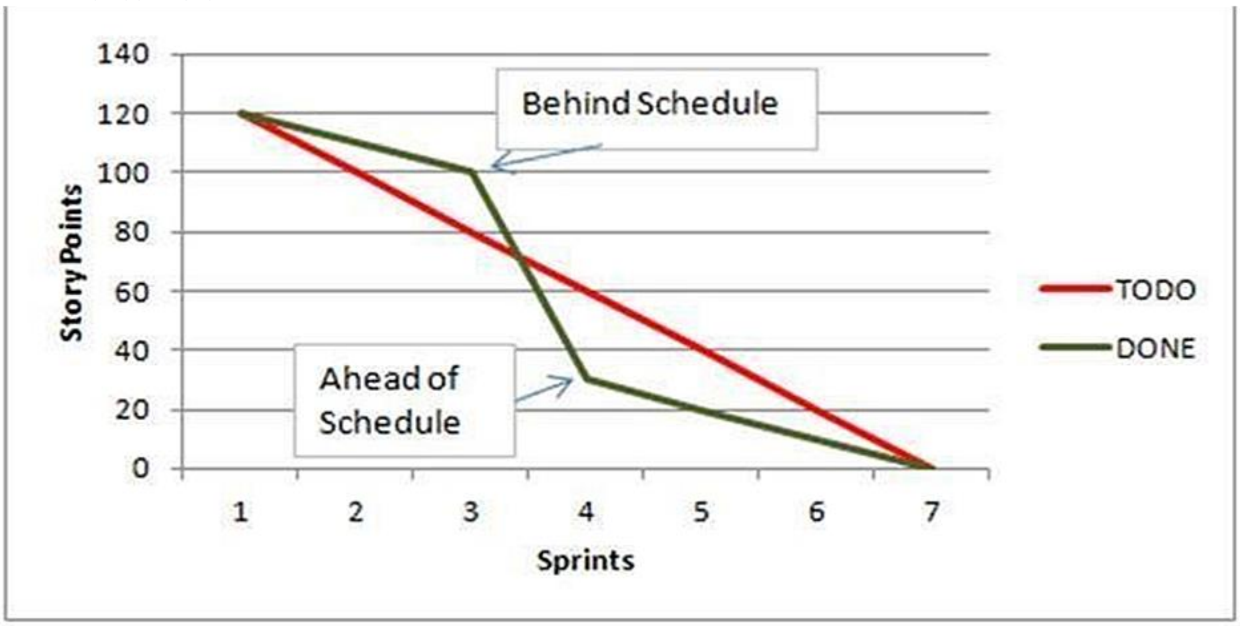
Project Tracker, Velocity& Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as onPlanned End Date)	Sprint ReleaseDate (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	35	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	45	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	07 Nov 2022

Velocity:

Imagine we have a 10-day sprint duration,and the velocityof the team is 20 (points per sprint). Let’s calculate the team’s averagevelocity (AV) per iteration unit (story points per day)

Burndown Chart:



CHAPTER 7 - CODING AND SOLUTIONING

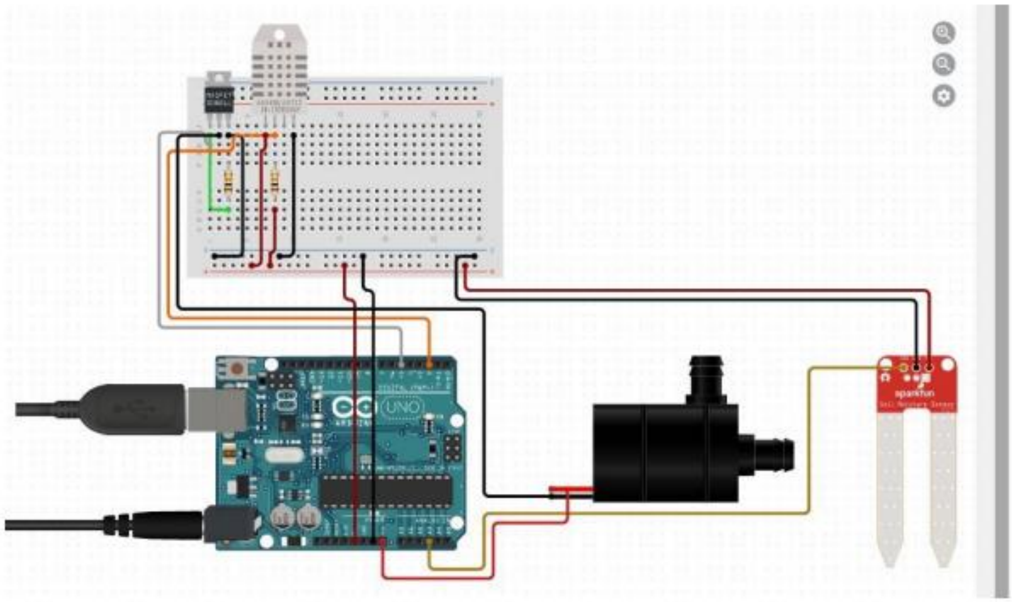
Connecting Sensors with Arduino using C++ code

```
#include "Arduino.h"
#include "dht.h"
#include "SoilMoisture.h"
#define dht_apin A0
const int sensor_pin = A1; //soil moisture
int pin_out = 9;
```

Iii

```
dht DHT;
 int c=0;
void setup()
{
pinMode(2, INPUT); //Pin 2 as INPUT
pinMode(3, OUTPUT); //PIN 3 as OUTPUT
pinMode(9, OUTPUT);//output for pump
}
void loop()
{
if (digitalRead(2) == HIGH)
{
digitalWrite(3, HIGH); // turn the LED/Buzz ON
delay(10000); // wait for 100 msecond
digitalWrite(3, LOW); // turn the LED/Buzz OFFdelay(100);
}
Serial.begin(9600);
delay(1000);
DHT.read11(dht_apin); //temprature
float h=DHT.humidity;
float t=DHT.temperature;
delay(5000);
Serial.begin(9600);
float moisture_percentage
;int sensor_analog;
sensor_analog = analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) *100 ) );
float m=moisture_percentage;
delay(1000);
if(m<40)//pump
 {
while(m<40)
{
digitalWrite(pin_out,HIGH); //open pump
sensor_analog = analogRead(sensor_pin);
moisture_percentage = ( 100 - ( (sensor_analog/1023.00) *100 ) );
m=moisture_percentage;
 delay(1000);
 }
digitalWrite(pin_out,LOW); //closepump
 }
 if(c>=0)
{
mySerial.begin(9600);
delay(15000);
Serial.begin(9600);
delay(1000);
 Serial.print("\r");
delay(1000);
```

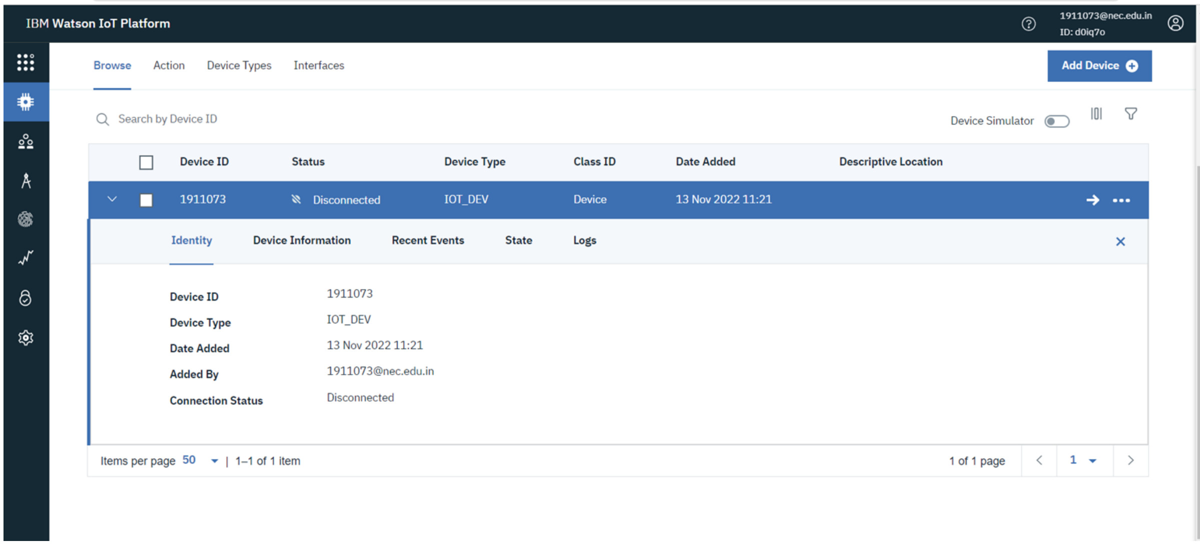
```
Serial.print((String)"update- >"+(String)"Temprature="+t+(String)"Humidity="+h+(String
)"Moisture="+m);
delay(1000);
```



Configuration of the IBM Watson IOT Platform and a device:

In the IBM Watson IOT Platform, under the catalog list, under the Internet of Things platform, a device has been created.From that the device credentials such as Device ID, Device Type, Organization ID,Authentication token were obtained.

Iii



Development of Python Script to publishdata to IBM Watson IOT platform:

Code:

```
import time
import sys
import

ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson

DeviceCredentialsorganization = "d0iq7o"

deviceType = "IOT_DEV"

deviceId = "1911073"

authMethod = "token"

authToken = "1911073abcdefgh" #

Initialize GPIOtry:

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
cloudas # an event of type "greeting" 10 times
deviceCli.connect()
while True: #Get
Sensor Data from
DHT11
temp=randomrandint(0,100)
pulse=random.randint(0,100) moisture=
random.randint(0,100)
humidity=random.randint(0,100);
lat = 17
lon = 18
data = { 'temperature' : temp, 'humidity' : humidity, 'Moisture' :moisture}
#print data

def myOnPublishCallback():
print ("Published Temperature = %s C" % temp,
"Humidity = %s
```

%%" % humidity,"Soil Moisture = %s %%"% moisture,"to IBMWatson")

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

if not success:

print("Not connected to IoTF")

time.sleep(1)

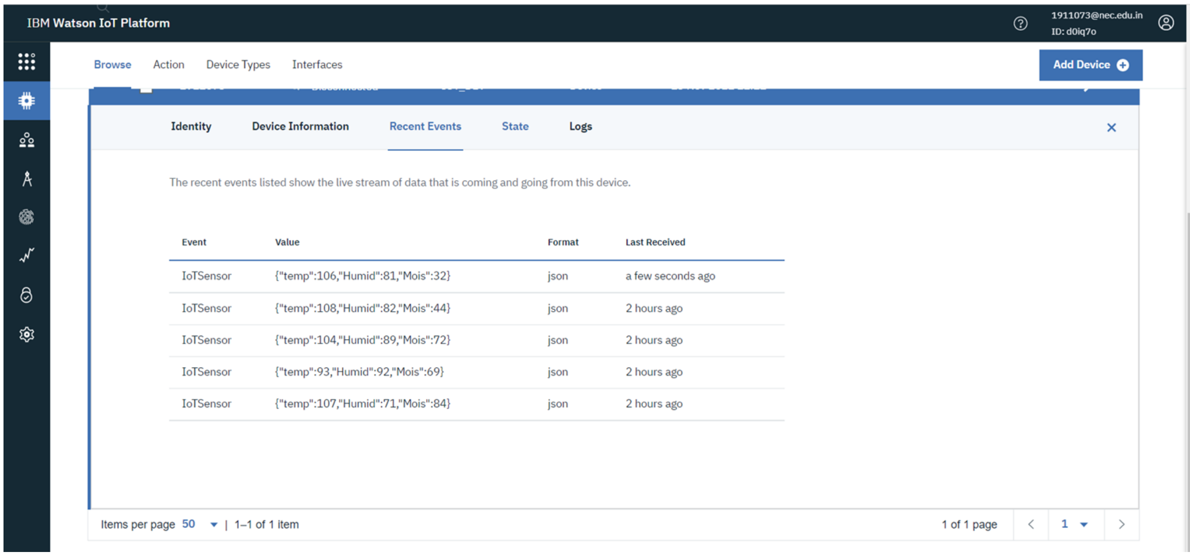
deviceCli.commandCallback = myCommandCallback

Disconnect the device and application from the clouddeviceCli.disconnect()

Python Code Output:

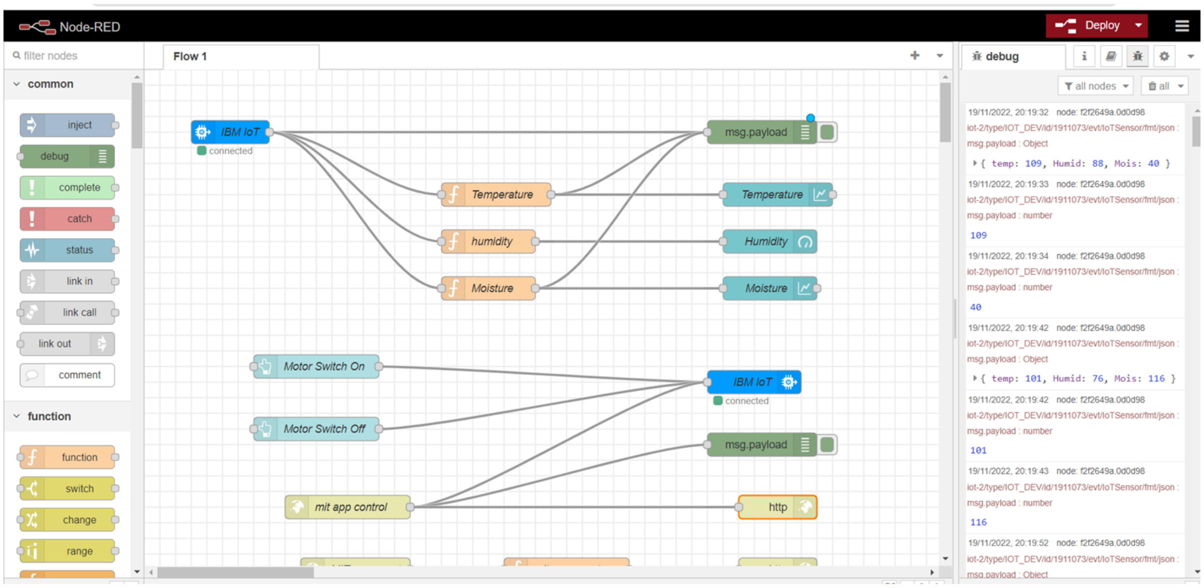
```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: C:\Users\User\Downloads\karthikaibm.py ==============
2022-11-19 18:15:15,790   ibmiotf.device.Client      INFO    Connected successfully: d:d0iq7o:IOT_DEV:1911073
 Published Temperature = 99 C Humidity = 65 % Moisture = 106 ppm to IBM Watson
 Published Temperature = 92 C Humidity = 62 % Moisture = 39 ppm to IBM Watson
 Published Temperature = 96 C Humidity = 99 % Moisture = 40 ppm to IBM Watson
 Published Temperature = 98 C Humidity = 67 % Moisture = 76 ppm to IBM Watson
 Published Temperature = 95 C Humidity = 63 % Moisture = 75 ppm to IBM Watson
 Published Temperature = 90 C Humidity = 82 % Moisture = 116 ppm to IBM Watson
 Published Temperature = 96 C Humidity = 77 % Moisture = 70 ppm to IBM Watson
 Published Temperature = 99 C Humidity = 67 % Moisture = 51 ppm to IBM Watson
Command received: motoron
motor is on
 Published Temperature = 94 C Humidity = 62 % Moisture = 90 ppm to IBM Watson
 Published Temperature = 94 C Humidity = 71 % Moisture = 73 ppm to IBM Watson
 Published Temperature = 106 C Humidity = 81 % Moisture = 31 ppm to IBM Watson
 Published Temperature = 105 C Humidity = 83 % Moisture = 75 ppm to IBM Watson
 Published Temperature = 107 C Humidity = 71 % Moisture = 24 ppm to IBM Watson
 Published Temperature = 91 C Humidity = 98 % Moisture = 46 ppm to IBM Watson
 Published Temperature = 109 C Humidity = 91 % Moisture = 120 ppm to IBM Watson
 Published Temperature = 100 C Humidity = 93 % Moisture = 96 ppm to IBM Watson
 Published Temperature = 101 C Humidity = 62 % Moisture = 66 ppm to IBM Watson
 Published Temperature = 109 C Humidity = 85 % Moisture = 90 ppm to IBM Watson
 Published Temperature = 94 C Humidity = 82 % Moisture = 64 ppm to IBM Watson
 Published Temperature = 96 C Humidity = 69 % Moisture = 53 ppm to IBM Watson
 Published Temperature = 90 C Humidity = 72 % Moisture = 95 ppm to IBM Watson
```

IBM Cloudafter publishing data:



Creation of Node Red Service for device events:

In the IBM Watson IOT platform, under the catalog, under the Node Red app service, an application is deployed using cloud foundry. In the cloud foundry, a group has been created and using the ci pipeline, the app url is obtained.Using the URL, the Node red is launched. The IBM WatsonIOT platform is connected to Node red using the IBM IoT palette. Using appropriate palettes, the data published in the IBM IoT platform is printedin the debug window of Node red.



Code block for the functionpalette:

<u>Soil moisture</u>:

Soil = msg.payload.Moisture

msg.payload = "Soil Moisture :

"global.set('m',Soil)

msg.payload = Math.round(Soil)return msg;

<u>Humidity:</u>

Humidity = msg.payload.humidity

msg.payload = "Humidity : "

global.set('h',Humidity)

msg.payload = Math.round(Humidity)

return msg;

<u>Temperature:</u>

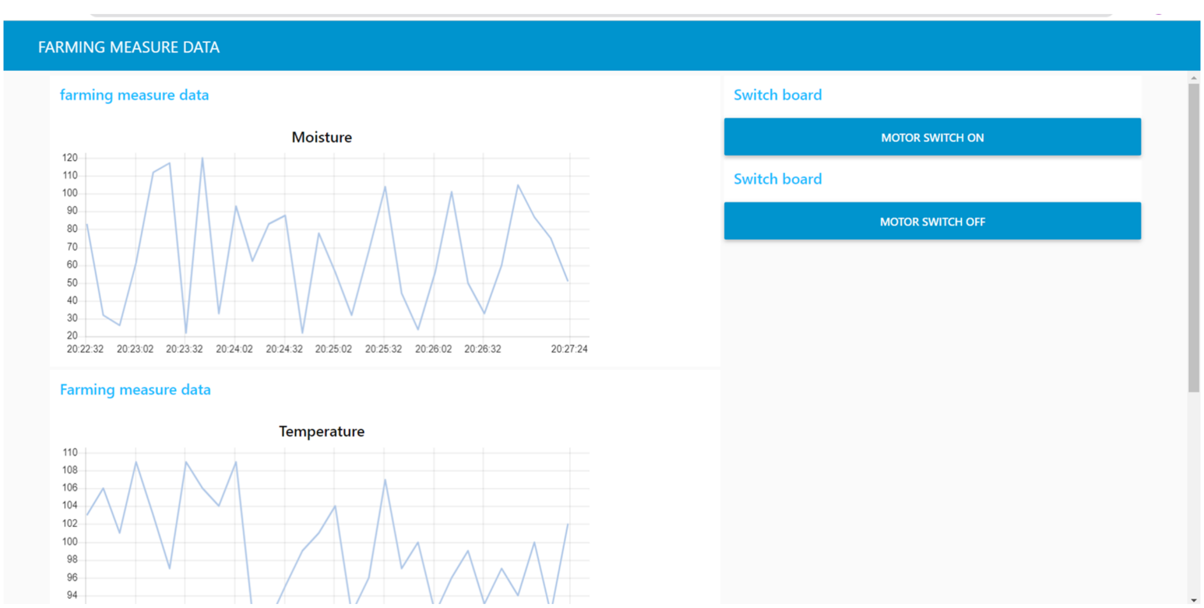
```
Temperature = msg.payload.temperature
msg.payload = "Temperature : "
global.set('t',Temperature)
msg.payload =Math.round(Temperature)
return msg;
```

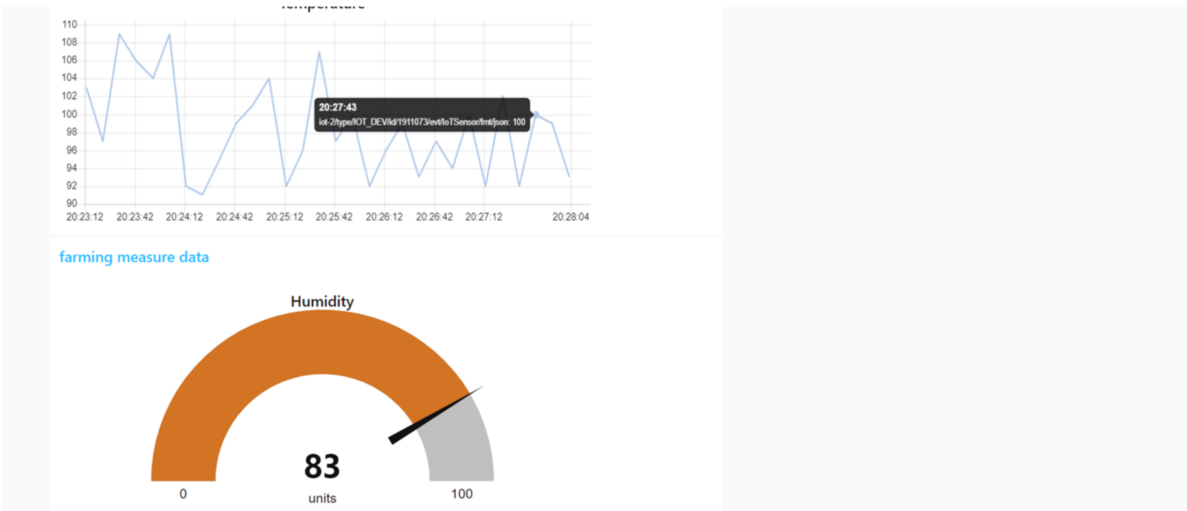
HTTP Function:

```
msg.payload = {"Temperature:": global.get('t'),"Humidity:":
global.get('h'),"Soil Moisture:": global.get('m')}
return msg;
```

Creation of Websitedashboard:

A website dashboard has been created using the gauge palette. It can be accessed by adding “/ui” in the main url of Node red. This dashboard displays the gauge representation of the data published in the IBM IOT platform.

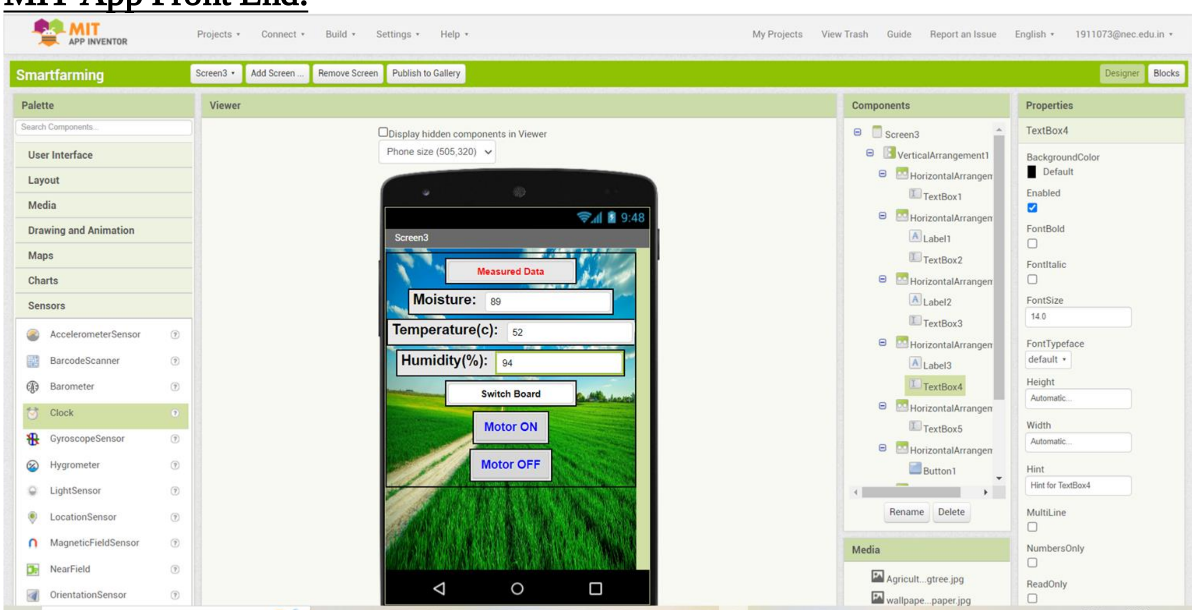




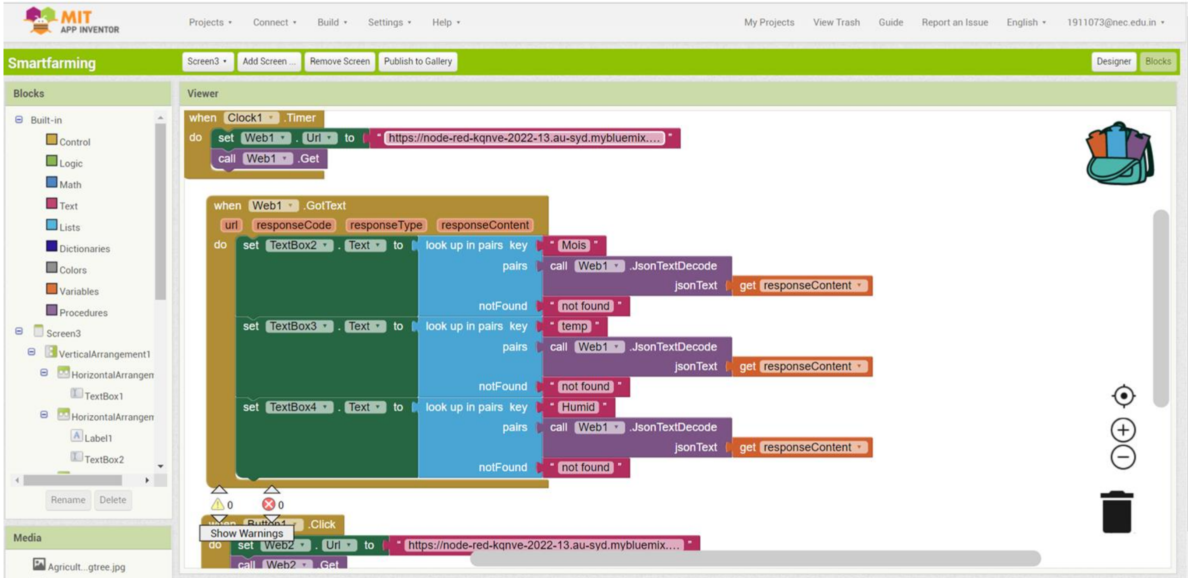
Development of Mobileapplication using MIT App Inventor:

In the MIT App Inventorplatform, an application is created which monitors the farmland parameters such as temperature, humidity, soil moisture and controls the actuators such as motors.

MIT App Front End:



Backend:



App working:

The app works based on HTTP protocol. The app uses HTTP GETmethodto parse the JSON data from the Node red website and displays the value in the UI. Using the HTTP POST method, the app sends command when a specific button is pressed. From where, the python code subscribes the command data from the cloud thereby notifying the commandis received.

CHAPTER 8 - PERFORMANCE METRICS

S. No.	Name of the Phase	Tasks Performed	Performance Metrics
1	Development of Problem Statement	The underlying problem analyzed and a rough idea of the solution was planned	The Problem statement wasdeveloped
2	Ideation Phase	Extracting use and test cases	Empathy map,Ideation and Literature surveywere formulated.

3	Project DesignPhase 1	Solution for theproblem is formulated and architecture is designed	Problem solutionfit was designed and the Proposed solutionis finalized with the help of Solution architecture.
4	Project DesignPhase 2	In depth analysis of the solution isperformed including requirements, tech stack,etc.	Solution Requirements, Overall Technology stack, Data flowdiagrams, User stories were formulated.
5	Project PlanningPhase	Various sprints were designed as individual progressive steps	Project Milestone and Sprint Plans were developed.

CHAPTER 9 - ADVANTAGES AND DISADVANTAGES

Advantages:

- By monitoring the soil parameters of the farm, the user can have a complete analysisof the field, in terms of numbers.
- Using the website and the application, an interactive experiencecanbe achieved.
- As the data gets pushed tothe cloud, one can access the data anywhere from this world.
- Without human intervention, water pump can be controlled through the mobile application and it's flow can be customized using servo motors.
- By using Raspberry Pi MCU, scalability can be increased due to itshigh processing power and enoughavailability of GPIO pins

Disadvantages:

- Data transfer is through the internet. So data fetch and push might delay due to slow internet connection, depending on the location andother physicalparameters.
- System can only monitora certain area of the field. In order to sense and monitor an entire field,sensors should be placed in manyplaces,which may increase the cost.
- Data accuracy may vary according to various physical parameters such as temperature, pressure, rain.
- Cost of the system is high due to usage of Raspberry Pi.
- Rodent and insects may cause damageto the system.

CHAPTER 10 – CONCLUSION

The project thus monitorsimportant parameters presentin the field such as temperature, humidity, soil moisture etc., and controls important actuators such as motors etc. It is helpful for farmers to remotely monitor their fields even during adverse weatherconditions and help them controlfarming equipments remotely using cloud.

CHAPTER 11 - FUTURE SCOPE

The project can be further extended by monitoring other parameters such as nutrient contents in the soil, soil texture etc. AI techniques integrated with cloud can be integrated to monitor any pest attacks present in the plant. The application can be made interactive which provides suggestions to farmers to improve their farmlands.

CHAPTER 12 – APPENDIX

Source Code:

```
import time
import sys
```

```
import
ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson
DeviceCredentialsorganization = "d0iq7o"
deviceType = "IOT_DEV"

deviceId = "1911073"
authMethod = "token"
authToken = "1911073abcdefgh" #
Initialize GPIOtry:
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
cloudas # an event of type "greeting" 10 times
deviceCli.connect()
        while True: #Get
        Sensor Data from

        DHT11
```

```
temp=randomrandint(0,100)
pulse=random.randint(0,100) moisture=
random.randint(0,100)
humidity=random.randint(0,100);
lat = 17
lon = 18
data = { 'temperature' : temp, 'humidity' : humidity, 'Moisture' :moisture}
                #print data


def myOnPublishCallback():
                print ("Published Temperature = %s C" % temp,
                "Humidity = %s
                %%" % humidity,"Soil Moisture = %s %%"%
                moisture,"to IBMWatson")

success = deviceCli.publishEvent("IoTSensor",        "json",
data,       qos=0,on_publish=myOnPublishCallback)
if not success:
                print("Not connected to
          IoTF")
                time.sleep(1)
deviceCli.commandCallback =
myCommandCallback
# Disconnect the device and application from the clouddeviceCli.disconnect()
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-26859-1660038860

PROJECT DEMO VIDEO LINK:

https://drive.google.com/file/d/1NkqB_v1s24a4rkELoQqPV7oMKN-33R6r/view?usp=drivesdk