PROJECT BASED EXPERIENTIAL LEARNING PROGRAM

(NALAIYA THIRAN)

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

Team ID : PNT2022TMID11102

Team Members:

SURESH KUMAR . M

SENTHIL RAJA . G

SOUNDARARAJAN . J

SARANKUMAR . S

INTRODUCTION

1.1 PROJECT OVERVIEW

Crops in farms are many times ravaged by animals like cows, buffaloes, goats, and birds etc. this leads to huge losses for the farmers. It is not possible for farmers to secure and protect the fields .so here we propose Smart crop protection system from animals and birds. This is a IOT based system we used. The microcontroller now sound an alarm to woo the animal away from the field as well as sends SMS to the farmer so that he may about the issue and come to the spot in case the animal don't turn away by the alarm. This ensures complete safety of crop from animals and thus protect the farmers from loss.

1.2 PURPOSE

Our Ultimate goal of the project is to develop intruder alert to the farm, to avoid losses due to animal and fire. These intruder alert protect the crop that damaging that indirectly increase yield of the crop. The develop system will not harmful and injurious to animals as well as human beings. Theme of project is to design a smart crop protection system for crops by using IOT system.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

The existing system mainly provides the surveillance functionality. Also these system don't provide protection from wild animals, especially in such an application area. They also need to take actions based on the type of animals that tries to enter the area, as different methods are adopted to prevent different animals from entering restricted areas. The other commonly used method by farmer in order to prevent the crop vandalization by animals include building physical barriers, use of electric fences and manual survillance and various such exhaustive and dangerous method.

2.2 REFERENCES

i. Mr.Pranav shitap, Mr.Jayesh redij, Mr.Shikhar Singh, Mr.Durvesh Zagade, Dr. Sharada Chougule. Department of ELECTRONICS AND TELECOMMUNICATION ENGINEERING, Finolex Academy of Management and technology, ratangiri, India.

ii. N.Penchalaiah, D.Pavithra, B.Bhargavi, D.P.Madhurai, K.EliyasShaik,S.Md.sohaib.Assitant Professor, Department of CSE,AITS, Rajampet,India UG Student, Department of CSE,AITS,Rajampet, India.

iii. Mr.P.Venkateswara Rao, Mr.Ch Shiva Krishna ,MR M Samba Siva ReddyLBRCE,LBRCE,LBRCE.

iv. Mohit Korche,Sarthak Tokse, ShubhamShirbhate, Vaibhav Thakre,S. P. Jolhe(HOD). Students , Final Year,Dept.of Electrical engineering,Government College of engineering,Nagpur head of dept.,Electrical engineering,GovernmentCollege of engineering,Nagpur.

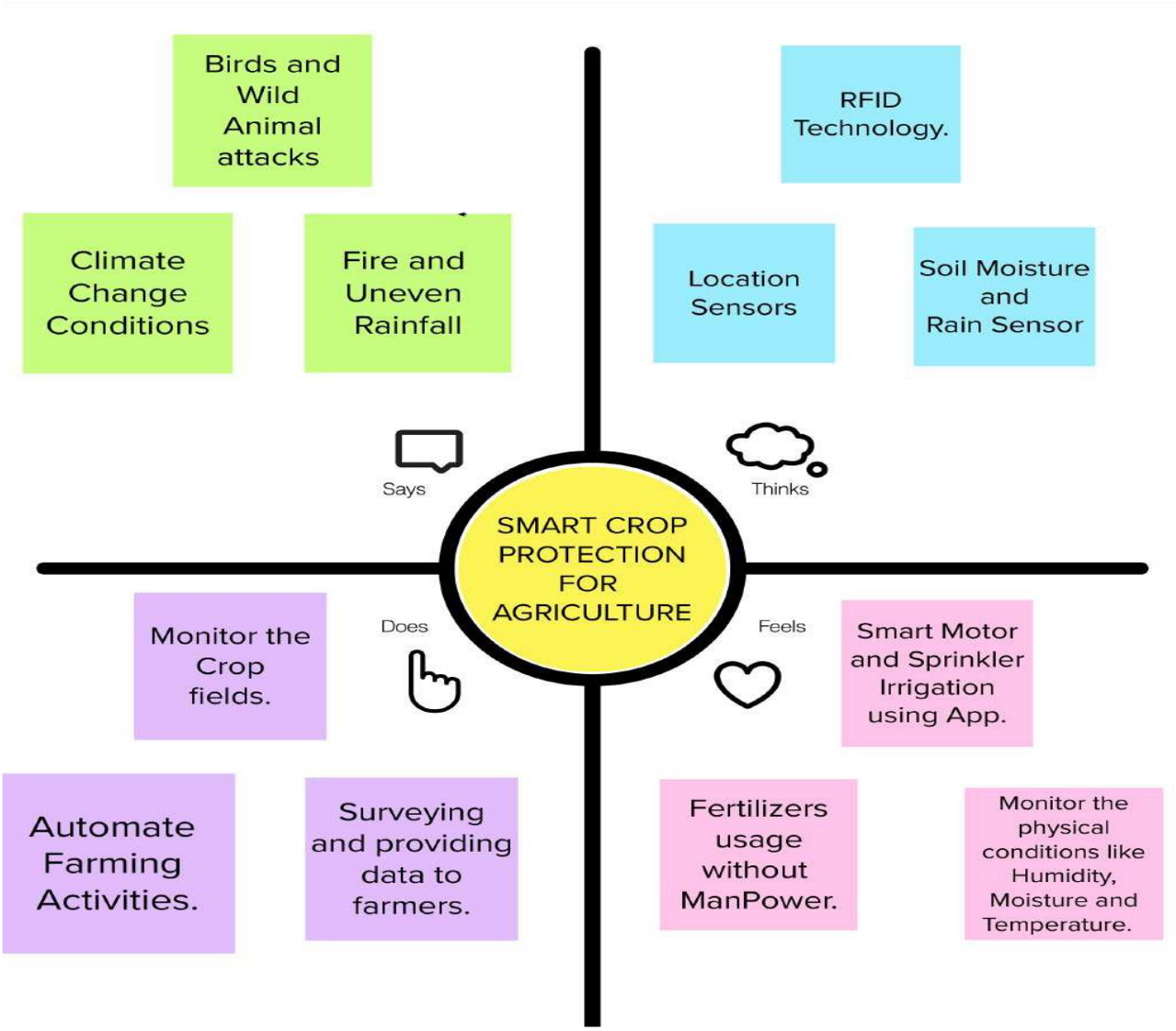
2.3 PROBLEM STATEMENT

This project describes the method of tracking the crops and protecting the crops from the birds and animals then it maintains the soil moisture, temperature etc.The traditional agriculture and allied sector cannot meet the requirements of modern Agriculture which requires high-yield, high quality and efficient output.Thus, it is very Important to turn towards modernization of existing methods and using the information Technology and data over a certain period to predict the best possible productivity and crop Suitable on the very particular land.The adoptions of access to high-speed internet, mobile devices, and reliable, low-cost Satellites (for imagery and positioning) are few key technologies characterizing the precision Agriculture trend.Precision agriculture is one of the most famous applications of IoT in the agricultural sector And numerous organizations are leveraging this technique around the world.

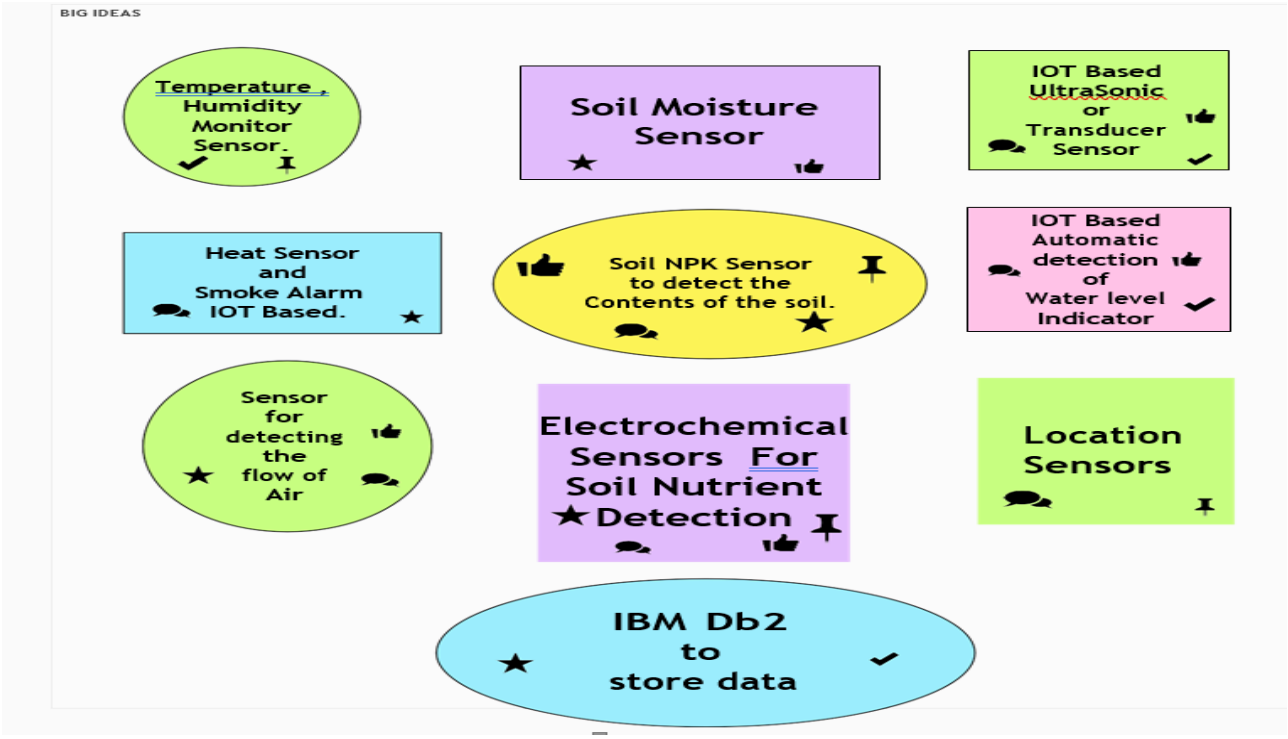
Birds and Wild animals attacks are a specific challenges for farmers in the world. Animals and birds may cause serious damage to crops. They can damage the crops by feeding on plant parts and by running over the fields. Therefore, the birds and wild animals may cause significant yield and financial problems. crop destruction and infest stored grains and food products. Additionally, they damage and weaken the quality of food crops.

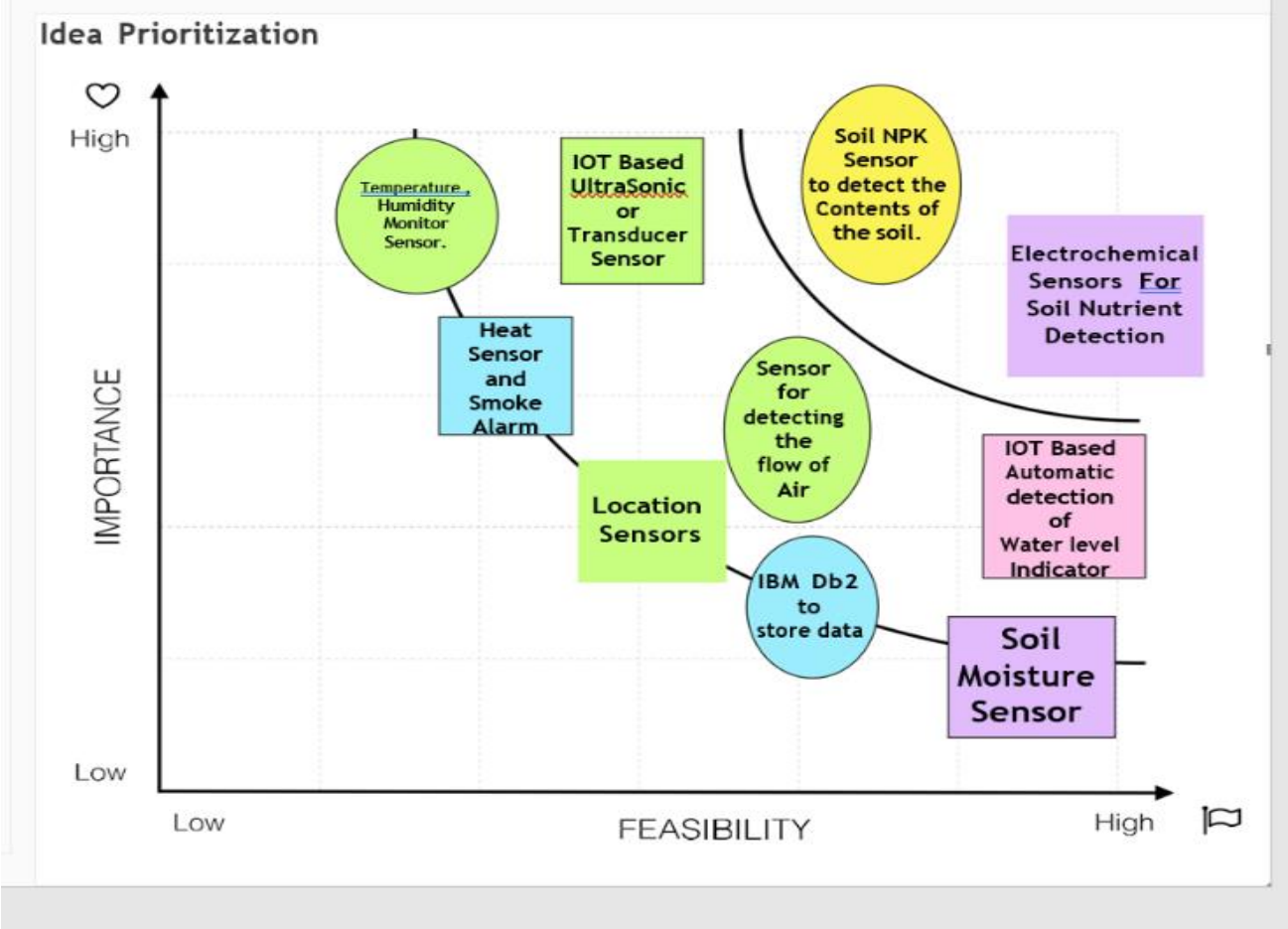
IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



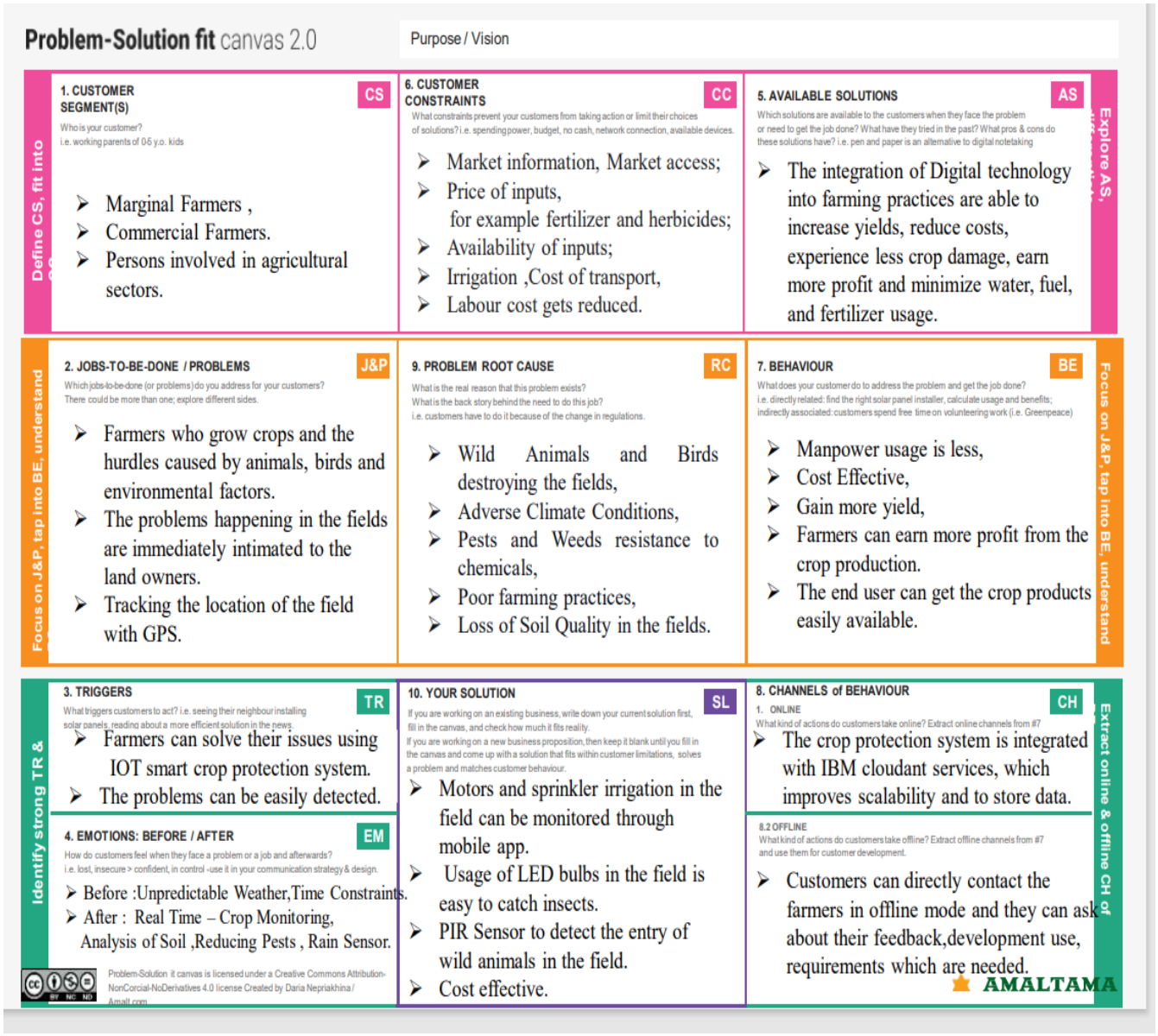


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	ProblemStatement (Problemtobe solved)	Birds and Wild animals attacks are a specific challenges for farmers in the world. Animals and birds may cause serious damage to crops. They can damage the crops by feeding on plant parts and by running over the fields. Therefore, the birds and wild animals may cause significant yield and financial problems. crop destruction and infest stored grains and food products. Additionally, they damage and weaken the quality of food crops.

2.	Idea/Solution description	A smart crop protection system helps farmers to keep crops protected from animals and birds that destroy crops. Automatically spraying the natural pesticides Use ultrasonic waves to protect crops from bugs and animals. It also helps farmers to monitor soil moisture levels in the field, as well as temperature and humidity values in the field. Motors and sprinkler irrigation in the field can be monitored through mobile app. Usage of LED bulbs in the field is easy to catch insects. Usage of Cameras and PIR Sensor to detect the entry of wild animals in the field.
3.	Novelty/Uniqueness	This Smart crop protection system helps the farmers to reduce their work and time. Monitors the soil moisture, Temperature and humidity level with the help of Sensors. Usage of Rain sensor in the field is efficient to sense the rain fall. Automatic irrigation can be done with the help of this crop protection system.
4.	Social Impact/Customer Satisfaction	The integration of Digital technology into farming practices are able to increase yields, reduce costs, experience less crop damage, earn more profit and minimize water, fuel, and fertilizer usage
5.	Business Model(Revenue Model)	Manpower usage is less, Cost Effective, Gain more yield, Farmers can earn more profit from the crop production. The end user can get the crop products easily available.
6.	Scalability of the Solution	The farmers can get a great relief from weed formation and climatic changes. They can concentrate on crop production. The farmers do not suffer from the drought and poverty due to loss in farming. This crop protection system uses techniques are integrated with IBM cloudant services, which helps efficiently in large scale and it improves scalability.

3.4 PROBLEM SOLUTION FIT



REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

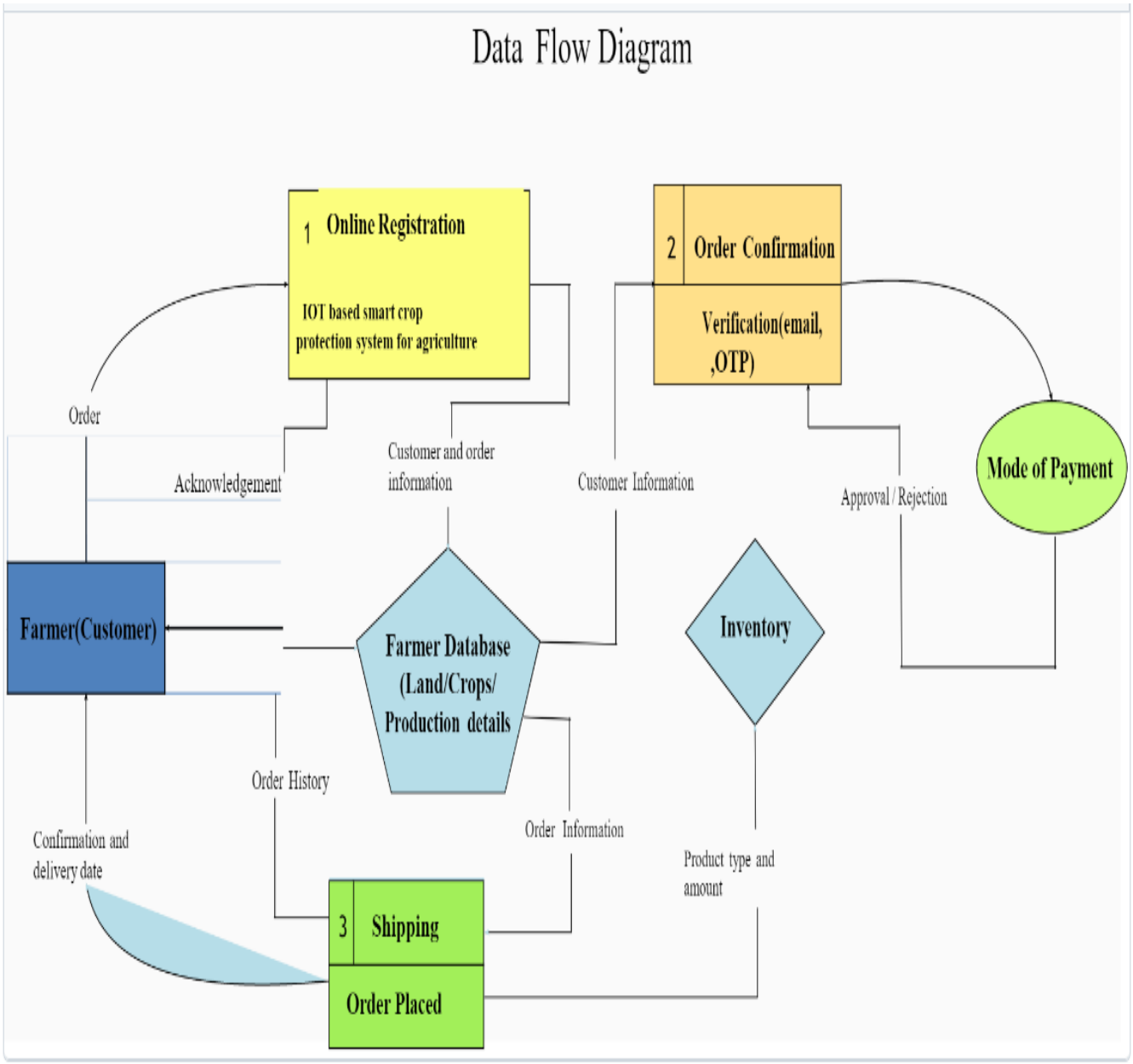
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	**User Requirements**	Crop Protection System, Automatic Irrigation System, Monitors the Soil Moisture, Monitors the field through Mobile app.
FR-2	**User Registration**	Registration through Gmail, Registration through Form, Registration through website,
FR-3	**User Confirmation**	Confirmation via Email Confirmation via OTP
FR-4	**Payment Options**	Bank Transfer, Cash on Delivery, Online Payments, Credit Card / Debit Card.
FR-5	**Product Delivery and Installation**	Door Step delivery,Take away, Free Installation with 1 year Warranty.
FR-6	**Product Feedback**	Through Website, Through Google forms and Through Email.

4.2 NON-FUNCTIONAL REQUIREMENT

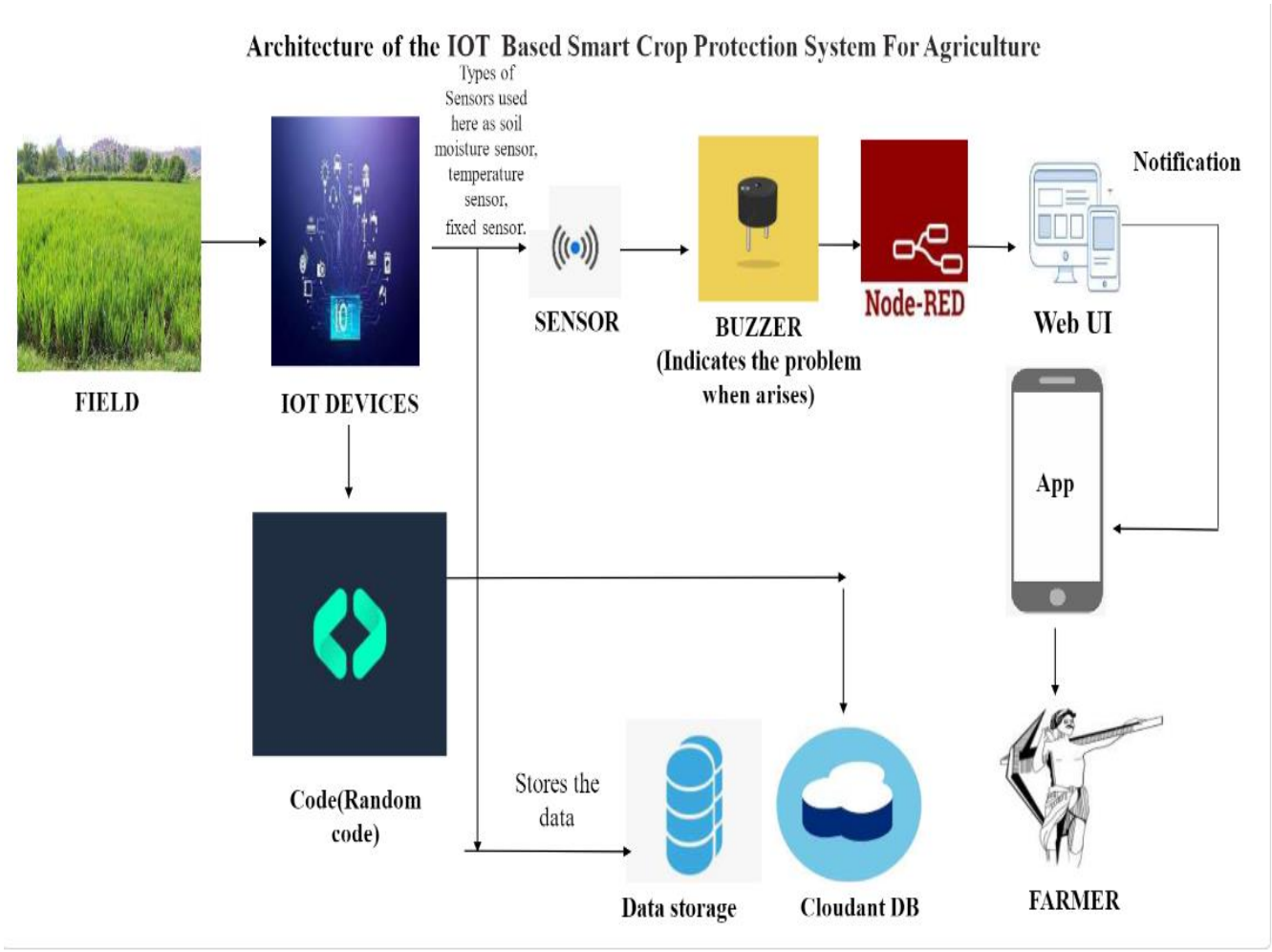
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usage is effective, Product information will be provided clearly, Products can be used without any complications and it is easy to use.
NFR-2	**Security**	Application is secured with Two step Verification, Passwords and patterns will be assigned as per the user requirements they needed. Cloud data includes in the network.
NFR-3	**Reliability**	Hardware requires a regular and frequent check and Software Service is Periodically Updated.
NFR-4	**Performance**	The application has a well user interface, Energy requirement is less.
NFR-5	**Availability**	The features will be available at the user required time. It depends on the need of the farmer and the Customization of the the user.
NFR-6	**Scalability**	The product has to cover all the space of a farm field and it is user and eco-friendly in nature.

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3USER STORIES

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		USN-1	Create the IBM Cloud services which are being used in this project.	6	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-1		USN2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-2		USN-3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform.	5	Medium	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-2		USN-4	In order to connect the IoT device to the IBM cloud, create a device in the IBM Watson IoT platform and get the device credentials.	5	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-3		USN-1	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.	10	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-3		USN-2	Create a Node-RED service.	10	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-3		USN-1	Develop a python script to publish random sensor data such as temperature, moisture, soil and humidity to the IBM IoT platform	7	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-3		USN-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-4		USN-3	Publish Data to The IBM Cloud	8	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-4		USN-1	Create Web UI in Node- Red	10	High	SureshKumar Senthil Raja Soundararajan Saran Kumar
Sprint-4		USN-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also	10	High	SureshKumar Senthil Raja Soundararajan
			use Cloudant DB nodes to store the received sensor data in the Cloudant DB			Saran Kumar

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

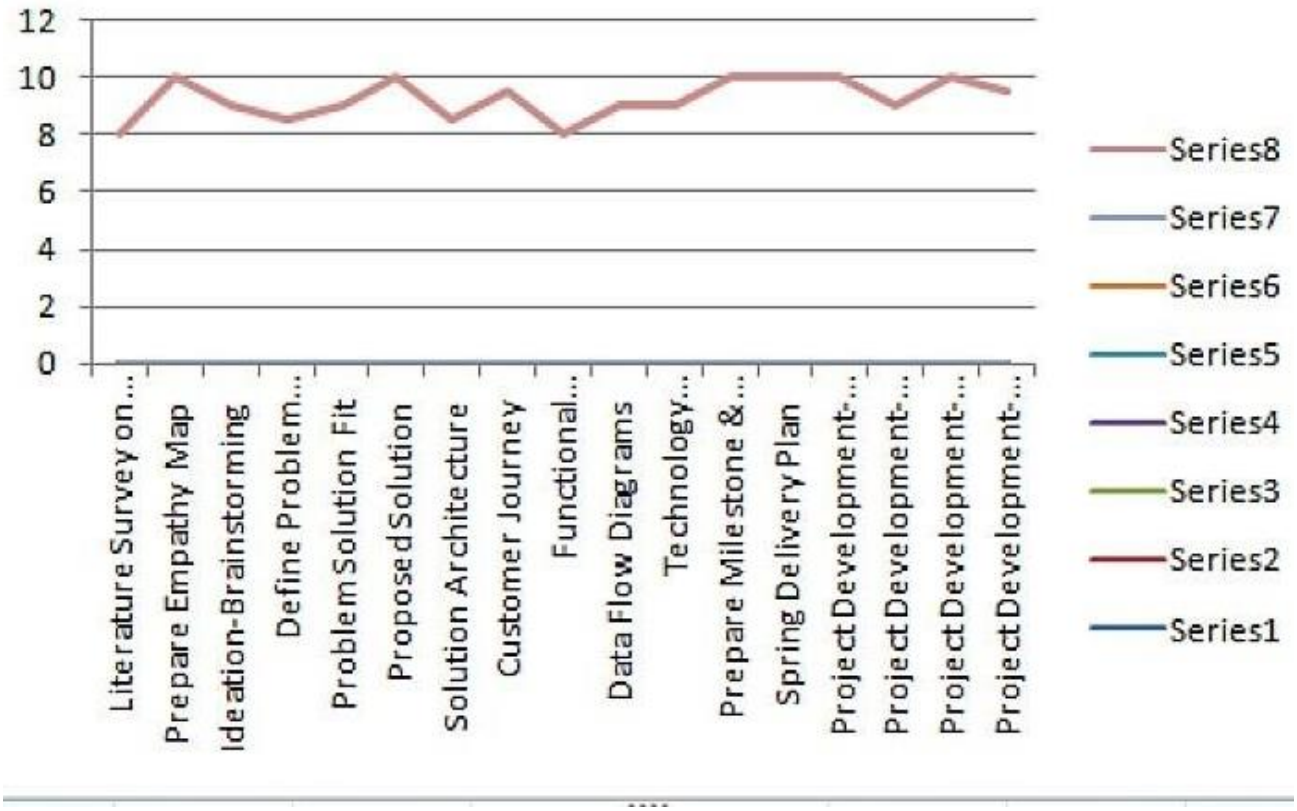
Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	05 Nov 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	08 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 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)

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



CODING AND SOLUTION:

7.1 FEATURE 1

```
import random
import
ibmiotf.application
import
ibmiotf.device from
time import sleep
import sys
organization = "gjx22e"
deviceType =
"smartcrop" deviceId =
"53302945" authMethod
= "use-token-auth"
authToken =
"987654321"
def
myCommandCallback(cmd)
: print("%s" %
cmd.data['command'])
status=cmd.data['command']
if status=="sprinkler_on":
print ("sprinkler is
turning ON")else :
print ("sprinkler is turning OFF")
try:
```

```
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "authmethod": authMethod, "auth-token": authToken}

deviceCli =

ibmiotf.device.Client(deviceOptions)

except Exception as e:

print("Exception detected in connecting device:

%s" % str(e))sys.exit()

deviceCli.conn

ect()while

True:

temp = round(

random.uniform(0,80),2)PH =

round(random.uniform(1,14),3)

moisture=

round(random.uniform(0,100),2)

water_level =

round(random.uniform(0,30),2)

temp_data = { 'Temp' : temp }

PH_data = { 'PH value' : Ph }

moist_data = { 'Moisture level' :

moist_level}water_data = { 'Water

level' : water_level}

success = deviceCli.publishEvent("Temperature sensor", "json",

temp_data, qos=0)sleep(1)

if success:

print ("... ...publish ok. ")

print ("Published Temp = %s C" % temp, "to IBM Watson")
```

```
success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
sleep(1)if
success:
print ("Published PH value = %s" % Ph, "to IBM Watson")
success = deviceCli.publishEvent("camera", "json",
camera_data, qos=0)sleep(1)
if success:
print ("Published Moisture level = %s " % moist_level, "to
IBM Watson") success = deviceCli.publishEvent("Water
sensor", "json", water_data, qos=0)sleep(1)
if success:
print ("Published Water level = %s cm" % water_level,
"to IBM Watson")print ("")
if (temp > 35):
print("sprinkler-1 is ON")
success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is
high, sprinkerlers are turnedON" %temp }, qos=0)
sleep(
1)if
success:
print( 'Published Alert1 : ', "Temperature(%s) is high, sprinkerlers are turned ON"
%temp,"to IBM Watson")print("")
else:
print("sprinkler-1 is
OFF")print("")
if (Ph > 7.5 or Ph < 5.5):
```

success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH

level(%s) is not safe,use otherfertilizer" %Ph } , qos=0)

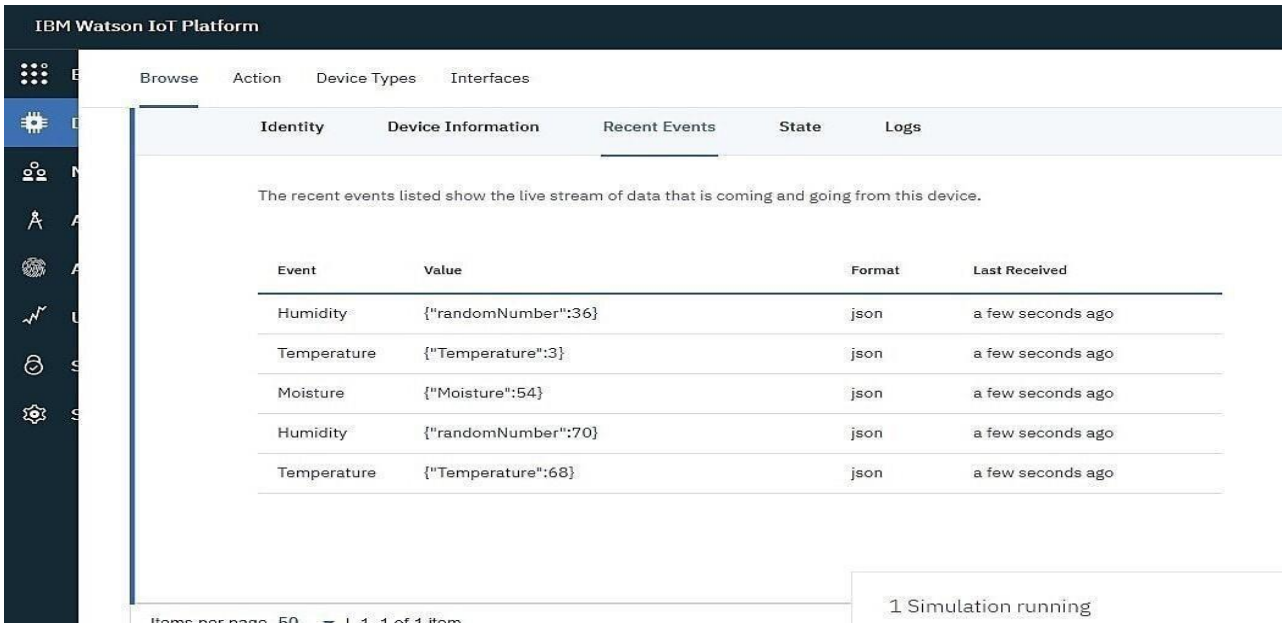
sleep(

1)if

success:

print('Published Alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer"

%Ph,"to IBM Watson")print("") deviceCli.commandCallback = myCommandCallbackdeviceCli.disconnect()



Delivery of Sprint-1

Bird's detection circuit:

To Protect the crops , fruits and vegetables from the birds by using Piezo electric buzzer with Arduino.

void setup()

{

pinMode(2,INPUT);

pinMode(13,OUTPUT

```
);
}
void loop()
{
if (digitalRead(2)==HIGH)
{
digitalWrite(13,HIGH);
}
else
{
digitalWrite(13,LOW);
}
delay(10);
}
```

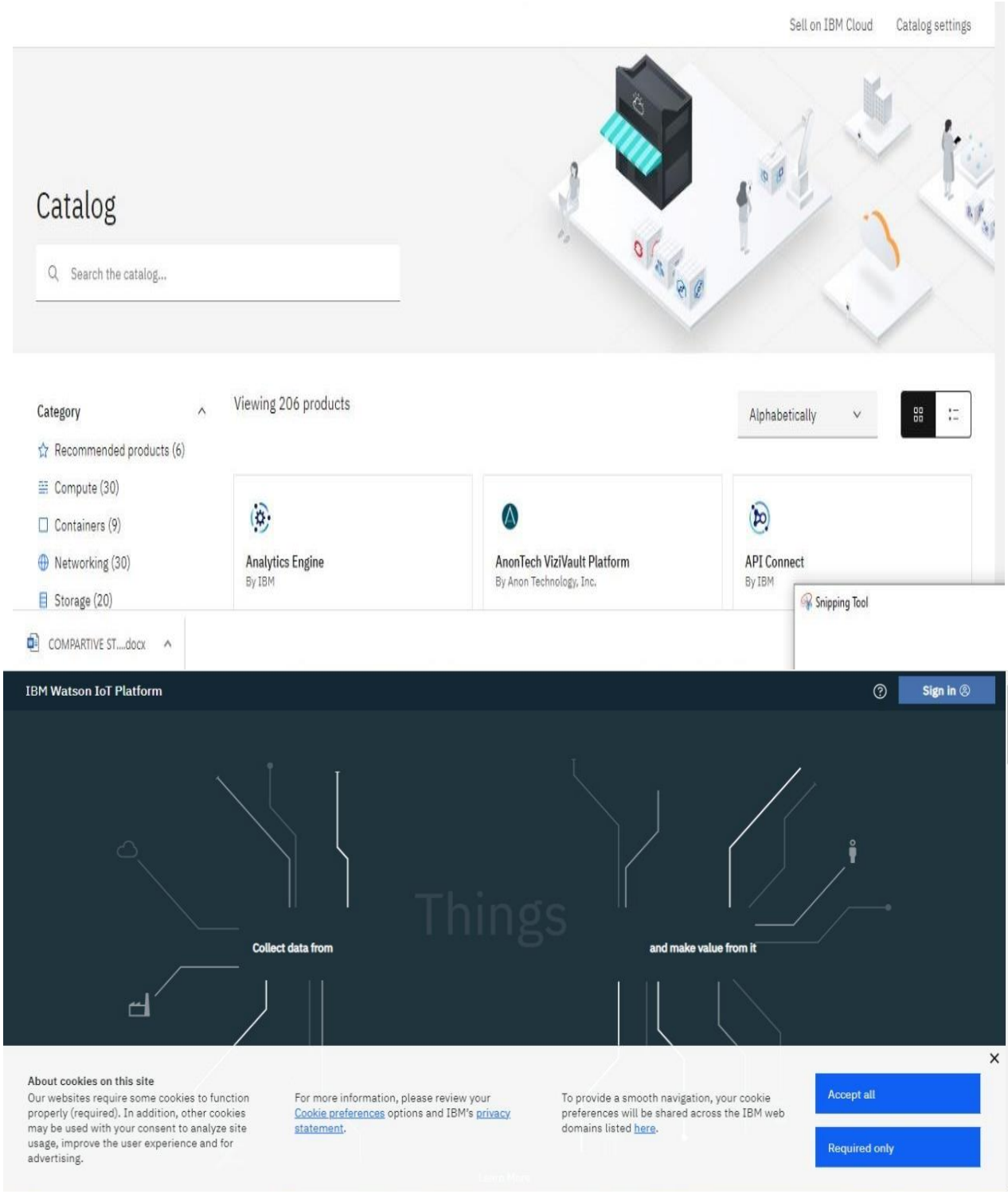
Moisture circuit:

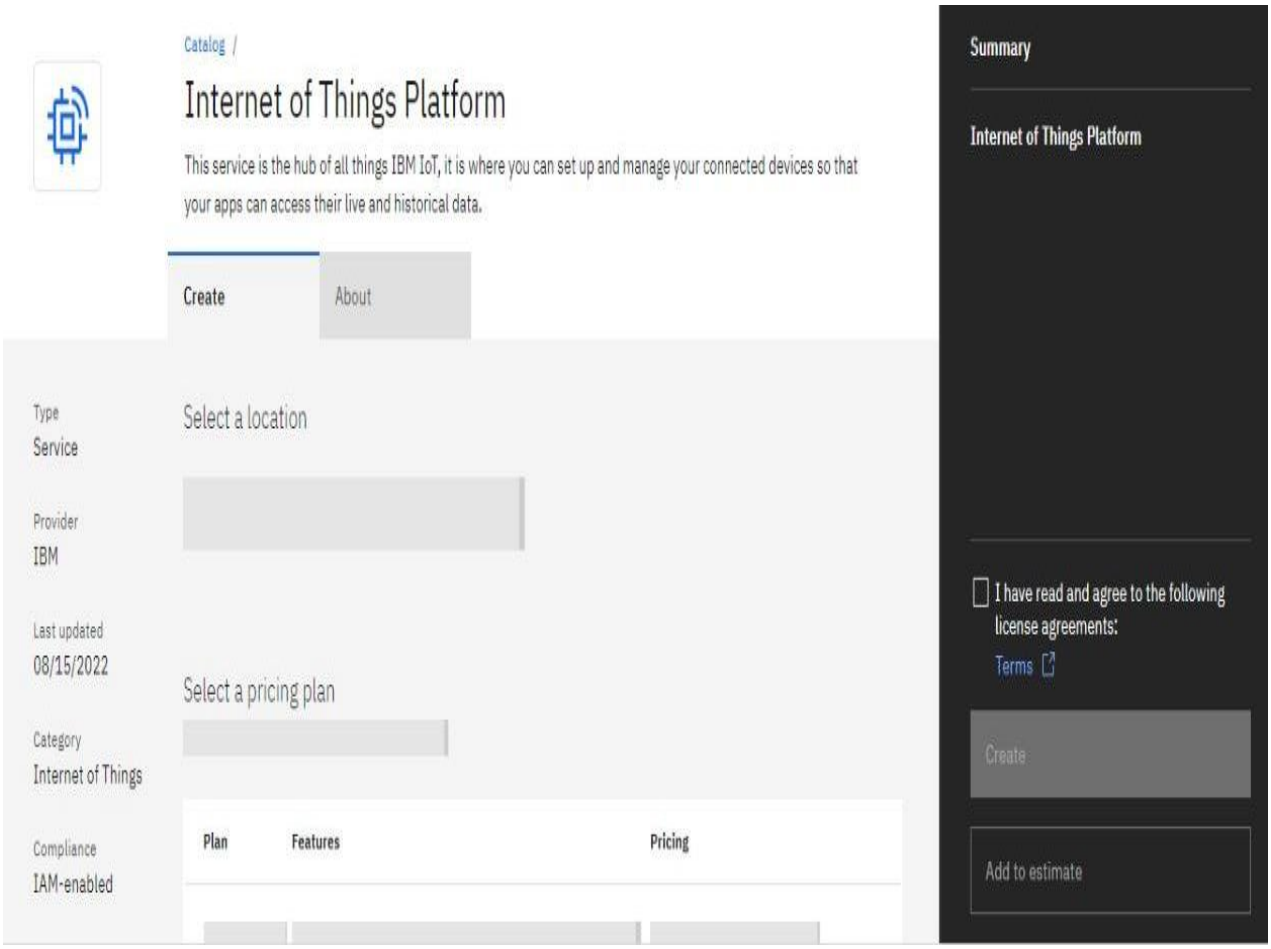
To detect the moisture level in the soil.

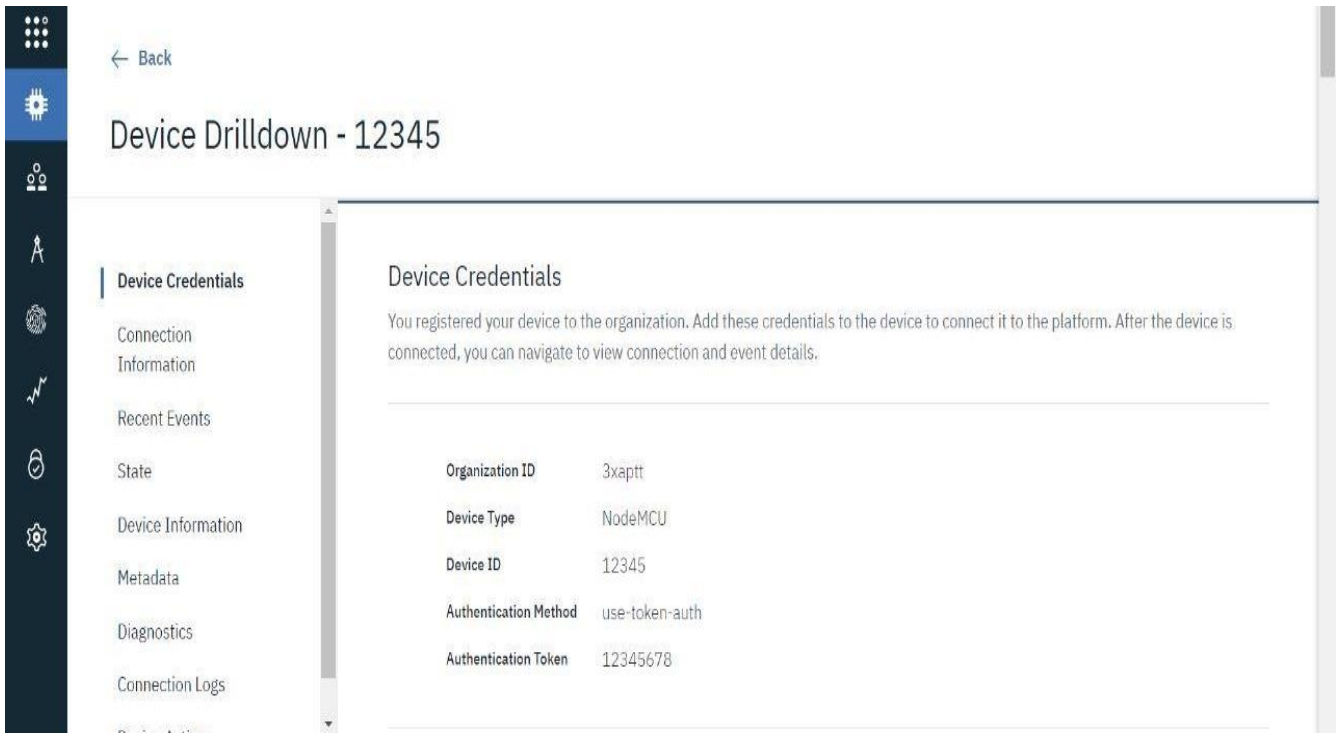
```
int moistureValue;
float moisture_percentage;
void setup()
{
Serial.begin(9600);
}
void loop()
```

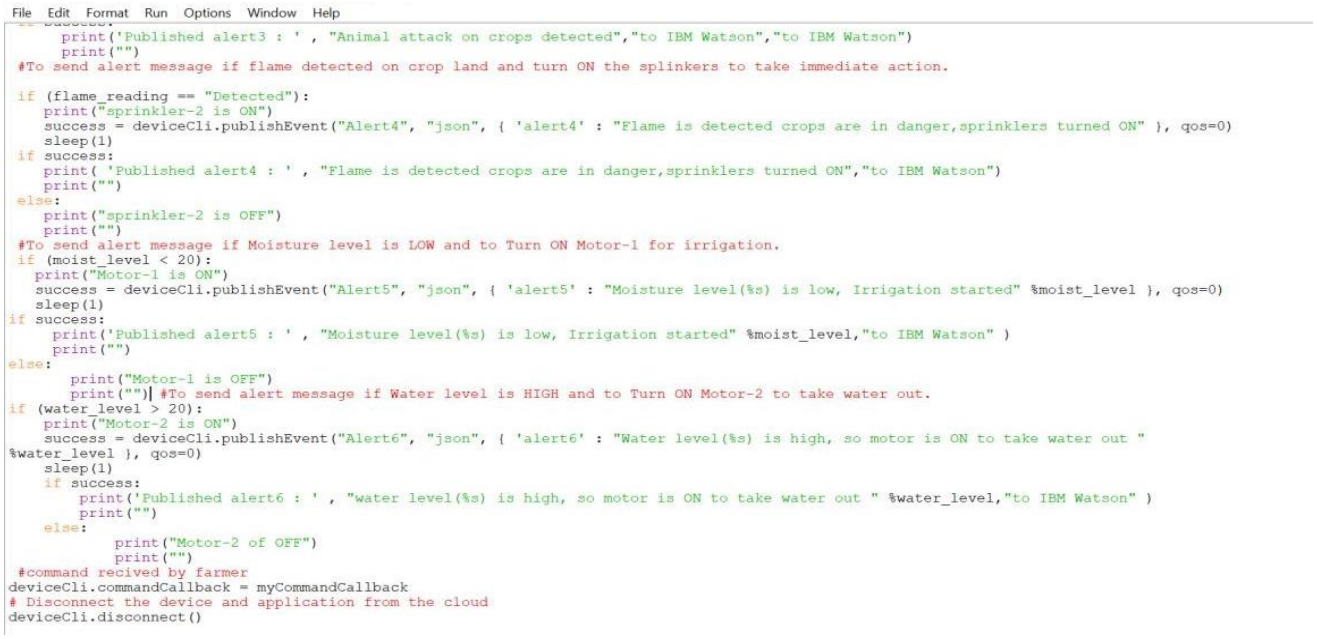
DELIVERY OF SPRINT-2

CREATION OF IBM WATSON CLOUD PLATFROM:




Catalog /
Internet of Things Platform
This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and historical data.
Create
About
Type
Service
Provider
IBM
Last updated
08/15/2022
Category
Internet of Things
Compliance
IAM-enabled
Select a location
Select a pricing plan
Plan
Features
Pricing
Summary
Internet of Things Platform
I have read and agree to the following license agreements:
Terms
Create
Add to estimate


Back
Device Drilldown - 12345
Device Credentials
Connection Information
Recent Events
State
Device Information
Metadata
Diagnostics
Connection Logs
Device Credentials
You registered your device to the organization. Add these credentials to the device to connect it to the platform. After the device is connected, you can navigate to view connection and event details.
Organization ID
3xaptt
Device Type
NodeMCU
Device ID
12345
Authentication Method
use-token-auth
Authentication Token
12345678



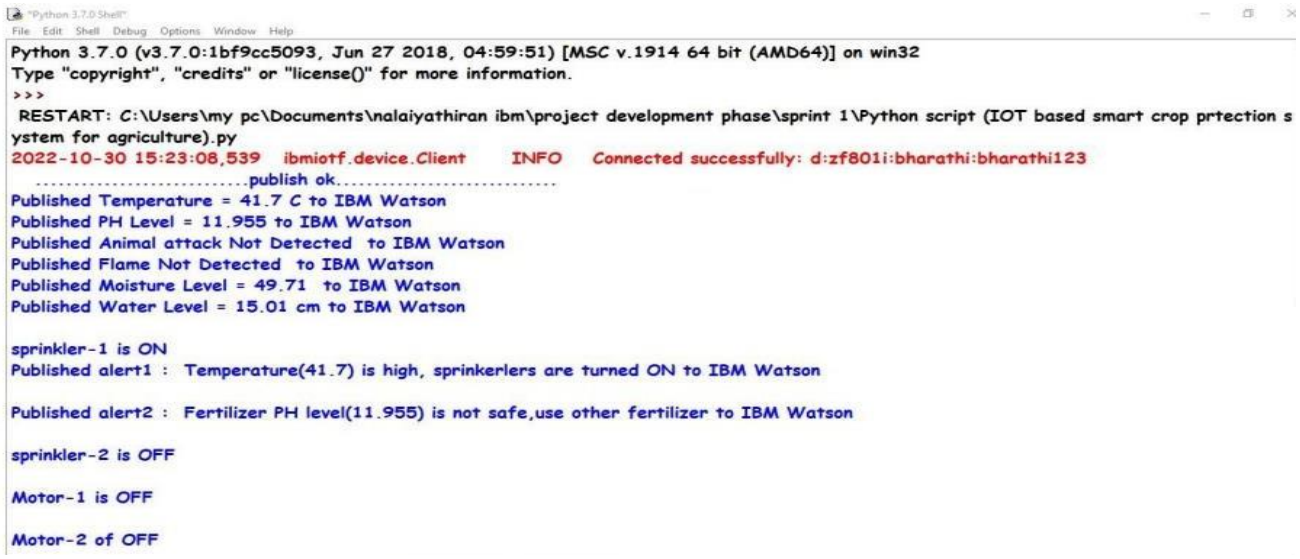
```
File  Edit  Format  Run  Options  Window  Help
      print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM Watson")
      print("")
 #To send alert message if flame detected on crop land and turn ON the splinkers to take immediate action.

 if (flame_reading == "Detected"):
    print("sprinkler-2 is ON")
    success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in danger,sprinklers turned ON" }, qos=0)
    sleep(1)
 if success:
    print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM Watson")
    print("")
 else:
    print("sprinkler-2 is OFF")
    print("")
 #To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.
 if (moist_level < 20):
   print("Motor-1 is ON")
   success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation started" %moist_level }, qos=0)
   sleep(1)
if success:
     print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started" %moist_level,"to IBM Watson" )
     print("")
else:
       print("Motor-1 is OFF")
       print("") #To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.
if (water_level > 20):
    print("Motor-2 is ON")
    success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON to take water out "
%water_level }, qos=0)
    sleep(1)
    if success:
        print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out " %water_level,"to IBM Watson" )
        print("")
    else:
            print("Motor-2 of OFF")
            print("")
 #command recived by farmer
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

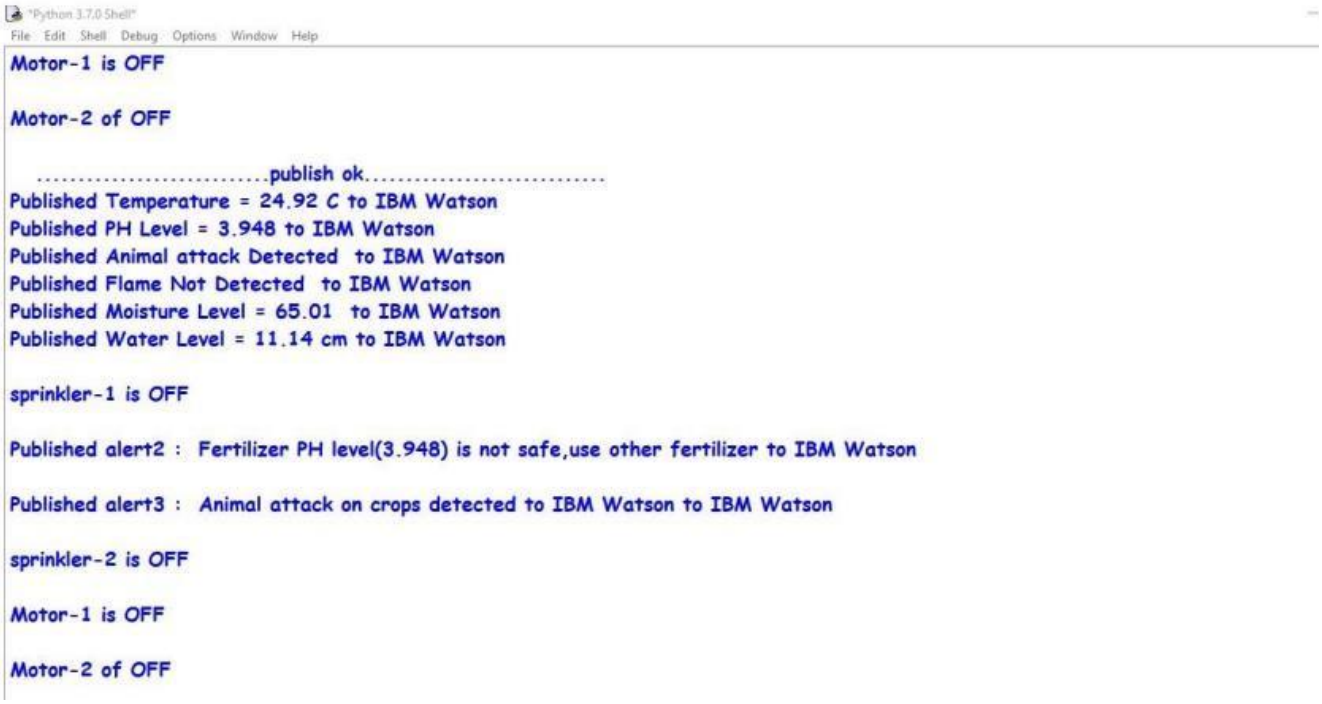
```
 import random
import
ibmiotf.application
import
ibmiotf.device from
time import sleep
import sys
organization = "gjx22e"
deviceType =
"smartcrop" deviceId =
"53302945" authMethod
= "use-token-auth"
authToken =
"987654321"
def
myCommandCallback(cmd)
: print("%s" %
cmd.data['command'])
status=cmd.data['command']
if status=="sprinkler_on":
print ("sprinkler is
turning ON")else :
print ("sprinkler is turning OFF")
try:
deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,
"authmethod": authMethod, "auth-token": authToken}
deviceCli =
ibmiotf.device.Client(deviceOptions)
except Exception as e:
print("Exception detected in connecting device:
%s" % str(e))sys.exit()
```

```
deviceCli.conn
ect()while
True:
temp = round(
random.uniform(0,80),2)PH =
round(random.uniform(1,14),3)
moisture=
round(random.uniform(0,100),2)
water_level =
round(random.uniform(0,30),2)
temp_data = { 'Temp' : temp }
PH_data = { 'PH value' : Ph }
moist_data = { 'Moisture level' :
moist_level}water_data = { 'Water
level' : water_level}
success = deviceCli.publishEvent("Temperature sensor", "json",
temp_data, qos=0)sleep(1)
if success:
print ("... ...publish ok. ")
print ("Published Temp = %s C" % temp, "to IBM Watson")
success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
sleep(1)if
success:
print ("Published PH value = %s" % Ph, "to IBM Watson")
success = deviceCli.publishEvent("camera", "json",
camera_data, qos=0)sleep(1)
if success:
print ("Published Moisture level = %s " % moist_level, "to
IBM Watson") success = deviceCli.publishEvent("Water
sensor", "json", water_data, qos=0)sleep(1)
if success:
print ("Published Water level = %s cm" % water_level,
"to IBM Watson")print ("")
if (temp > 35):
print("sprinkler-1 is ON")
success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is
high, sprinkerlers are turnedON" %temp }, qos=0)
sleep(
1)if
success:
print( 'Published Alert1 : ', "Temperature(%s) is high, sprinkerlers are turned ON"
%temp,"to IBM Watson")print("")
else:
print("sprinkler-1 is
OFF")print("")
if (Ph > 7.5 or Ph < 5.5):
```

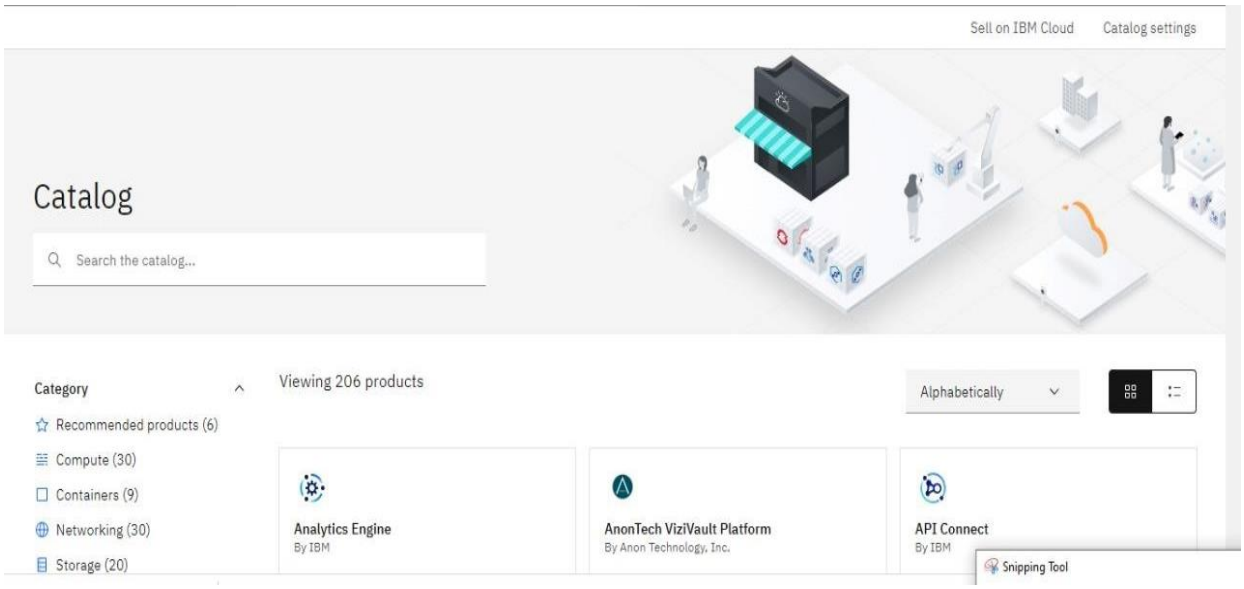
success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use otherfertilizer" %Ph } , qos=0)
sleep(
1)if
success:
print('Published Alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %Ph,"to IBM Watson")print("")
deviceCli.commandCallback = myCommandCallbackdeviceCli.disconnect()

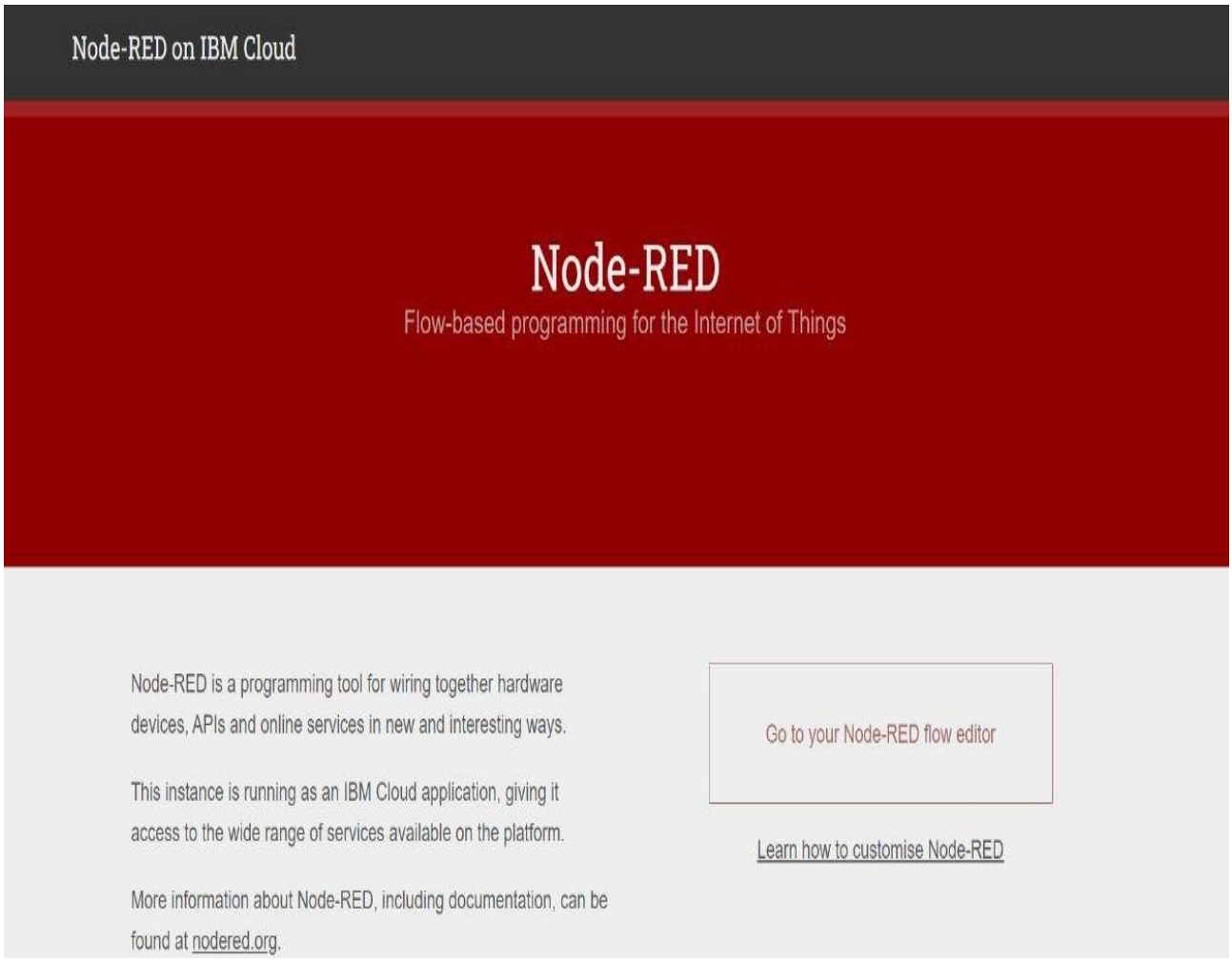
OUTPUT:

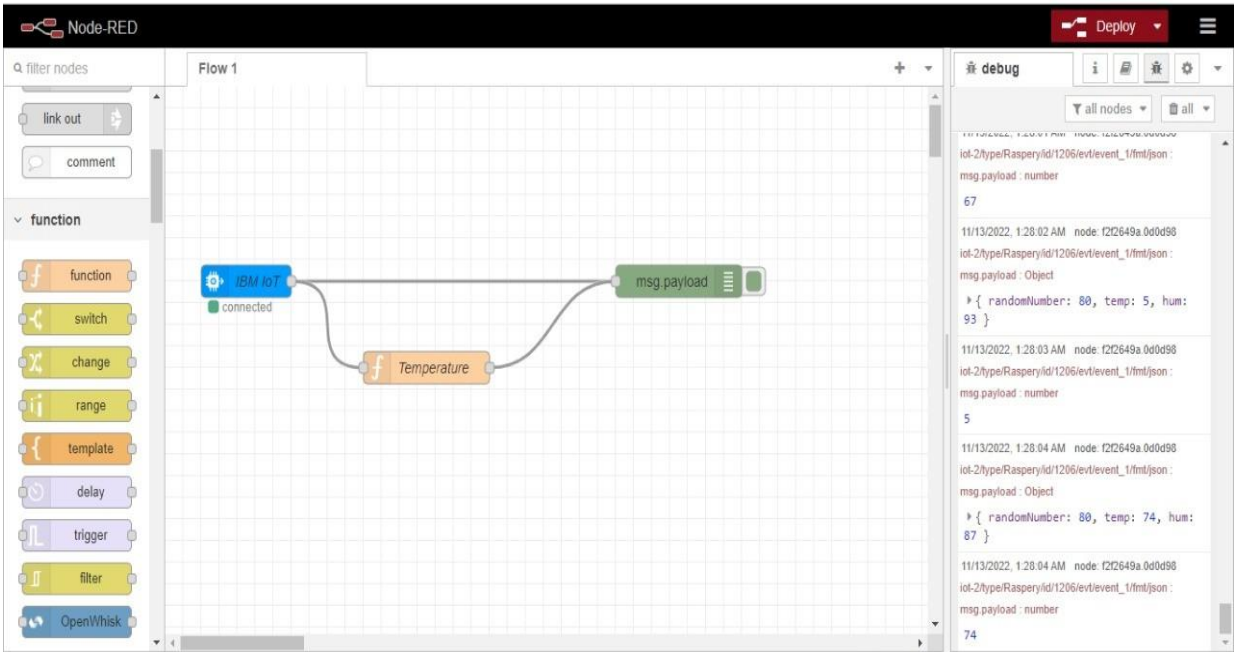


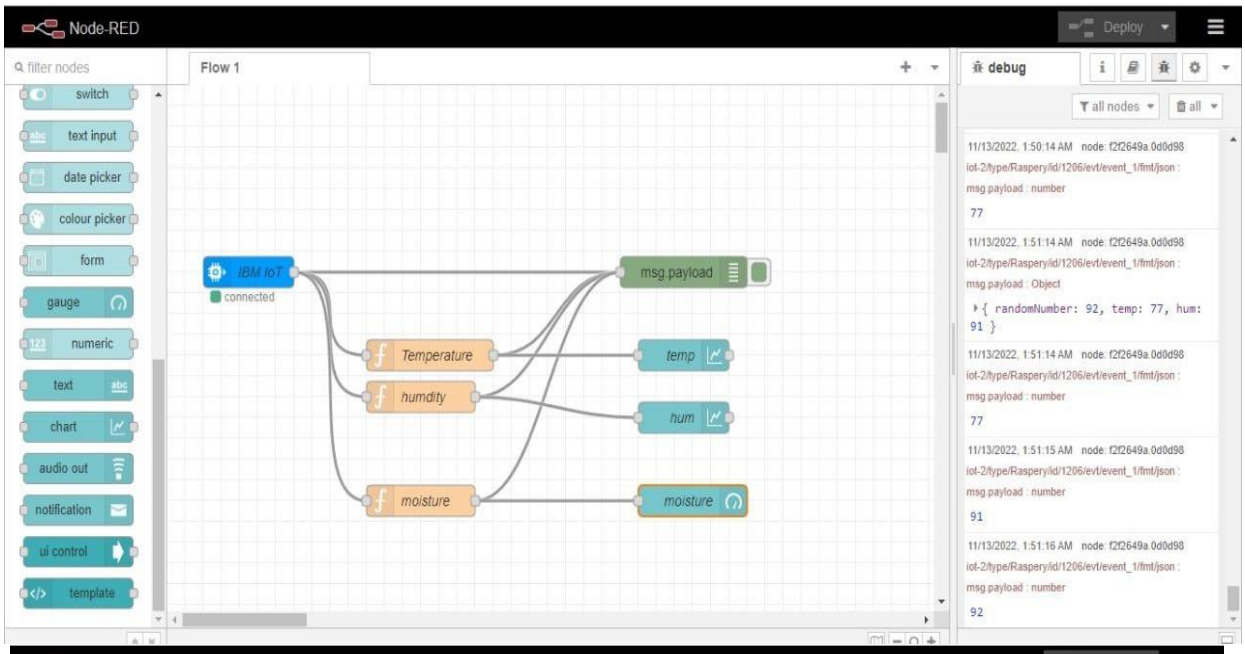


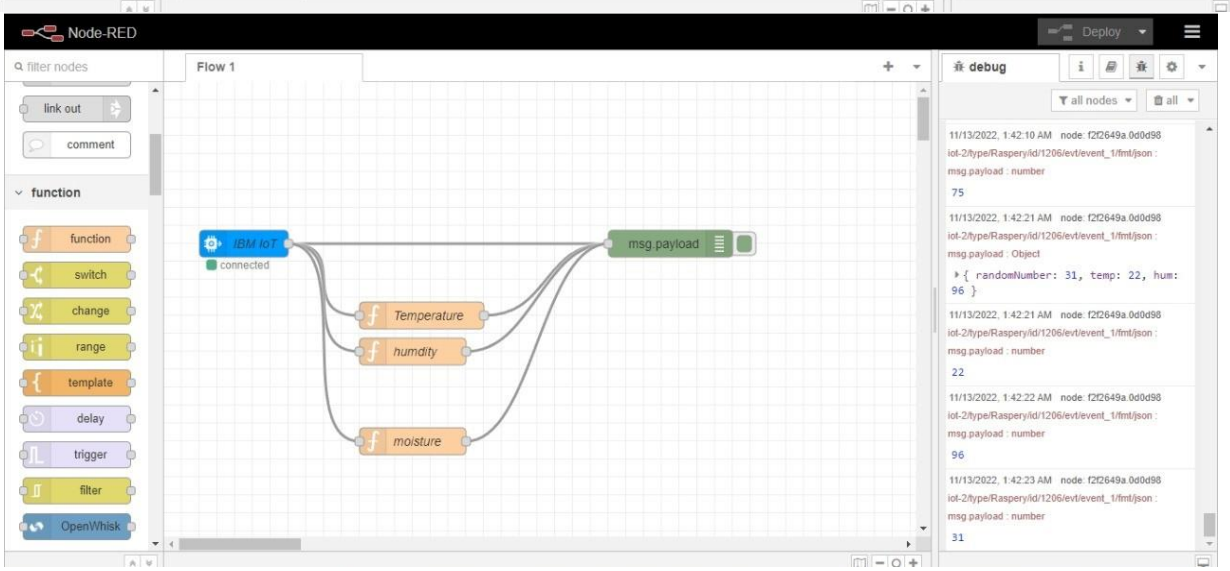
DELIVERY OF SPRINT-3:






Node-RED
Deploy
filter nodes
Flow 1
debug
link out
comment
function
function
switch
change
range
template
delay
trigger
filter
OpenWhisk
IBM IoT
connected
msg.payload
Temperature
all nodes
all
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
67
11/13/2022, 1:28:02 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : Object
{ randomNumber: 80, temp: 5, hum: 93 }
11/13/2022, 1:28:03 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
5
11/13/2022, 1:28:04 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : Object
{ randomNumber: 80, temp: 74, hum: 87 }
11/13/2022, 1:28:04 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
74


Node-RED
Deploy
filter nodes
Flow 1
debug
switch
text input
date picker
colour picker
form
gauge
numeric
text
chart
audio out
notification
ui control
template
IBM IoT
connected
msg.payload
Temperature
humdity
moisture
temp
hum
moisture
all nodes
all
11/13/2022, 1:50:14 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
77
11/13/2022, 1:51:14 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : Object
{ randomNumber: 92, temp: 77, hum: 91 }
11/13/2022, 1:51:14 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
77
11/13/2022, 1:51:15 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
91
11/13/2022, 1:51:16 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
92


Node-RED
Deploy
filter nodes
Flow 1
debug
link out
comment
function
function
switch
change
range
template
delay
trigger
filter
OpenWhisk
IBM IoT
connected
msg.payload
Temperature
humdity
moisture
all nodes
all
11/13/2022, 1:42:10 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
75
11/13/2022, 1:42:21 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : Object
{ randomNumber: 31, temp: 22, hum: 96 }
11/13/2022, 1:42:21 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
22
11/13/2022, 1:42:22 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
96
11/13/2022, 1:42:23 AM node: f2f2649a.0d0d98
iot-2/type/Raspery/id/1206/evt/event_1/fmt/json : msg.payload : number
31

SMART CROP PROTECTION

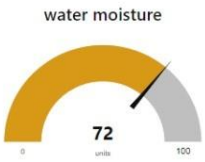
HALL AC

garden

SMART CROP PROTECTION

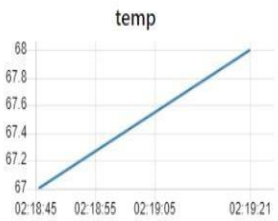
HALL AC

garden

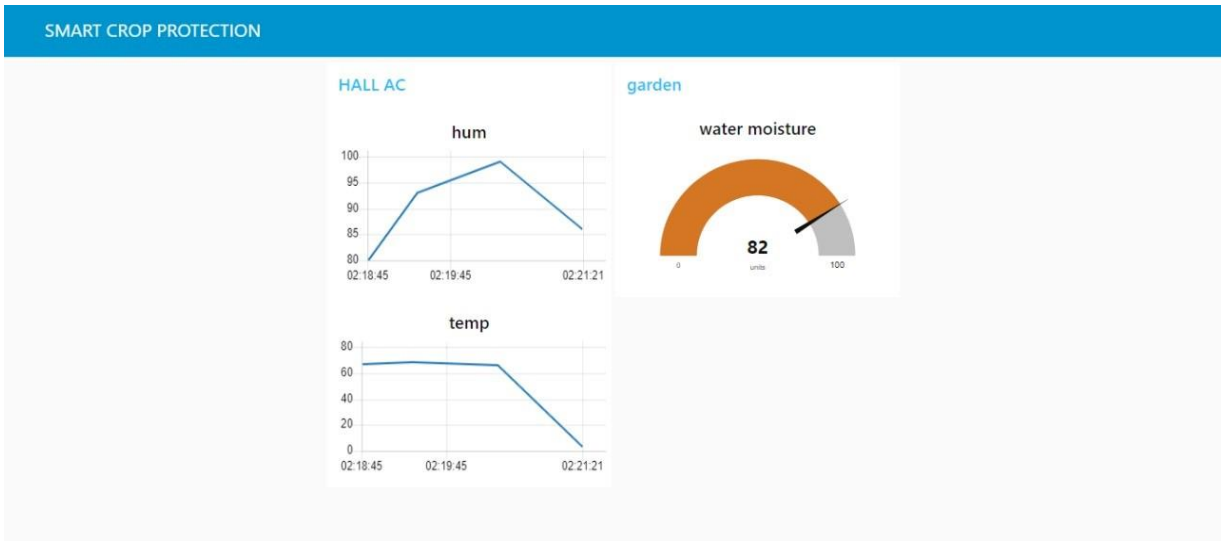


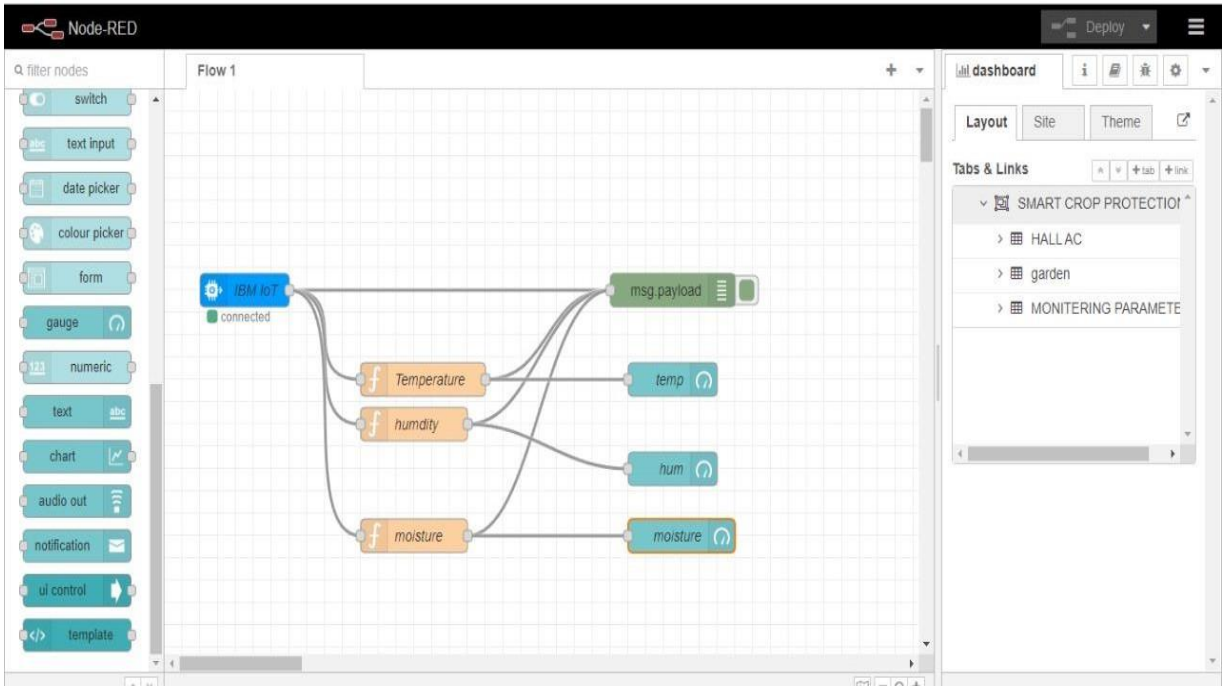
SMART CROP PROTECTION

HALL AC



garden


SMART CROP PROTECTION
HALL AC
hum
100
95
90
85
80
02:18:45
02:19:45
02:21:21
temp
80
60
40
20
0
02:18:45
02:19:45
02:21:21
garden
water moisture
82
0
units
100


Node-RED
Deploy
filter nodes
Flow 1
switch
text input
date picker
colour picker
form
gauge
numeric
text
chart
audio out
notification
ui control
template
IBM IoT
connected
Temperature
humdity
moisture
msg.payload
temp
hum
moisture
dashboard
Layout
Site
Theme
Tabs & Links
SMART CROP PROTECTIOI
HALL AC
garden
MONITERING PARAMETE

SMART CROP PROTECTION

MONITERING PARAMETERS

hum

water moisture

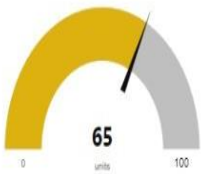
temp

SMART CROP PROTECTION

MONITERING PARAMETERS

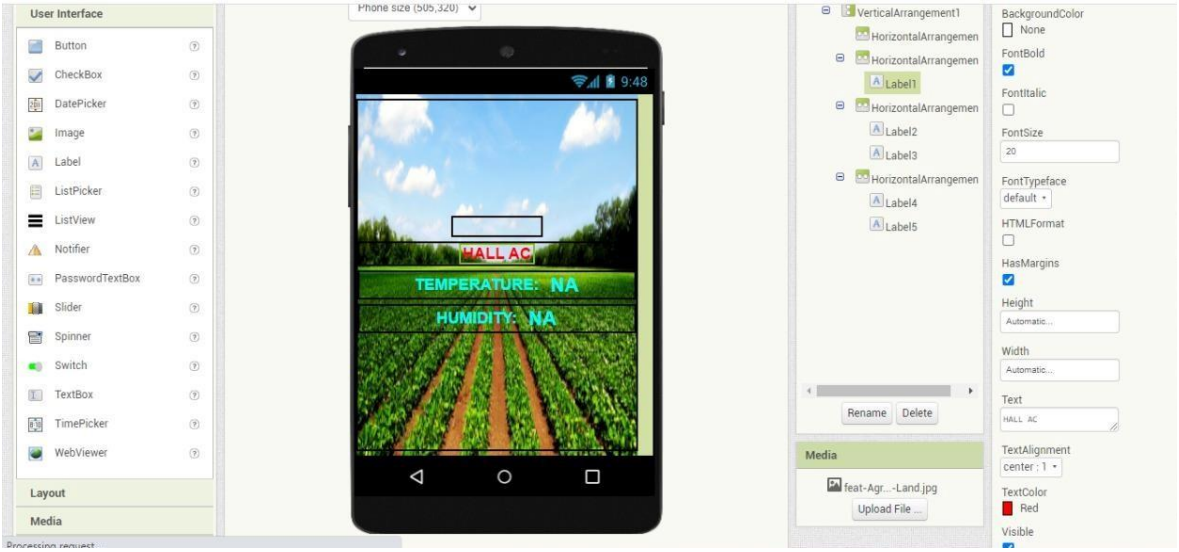
hum

water moisture

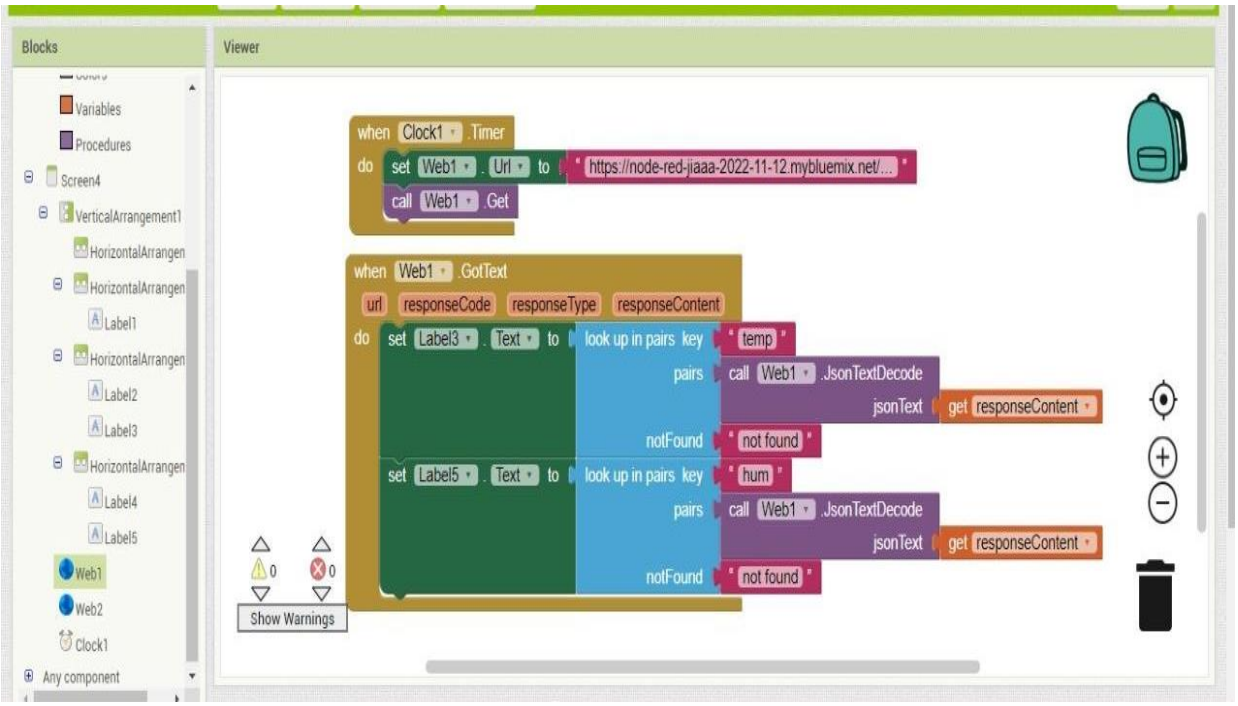


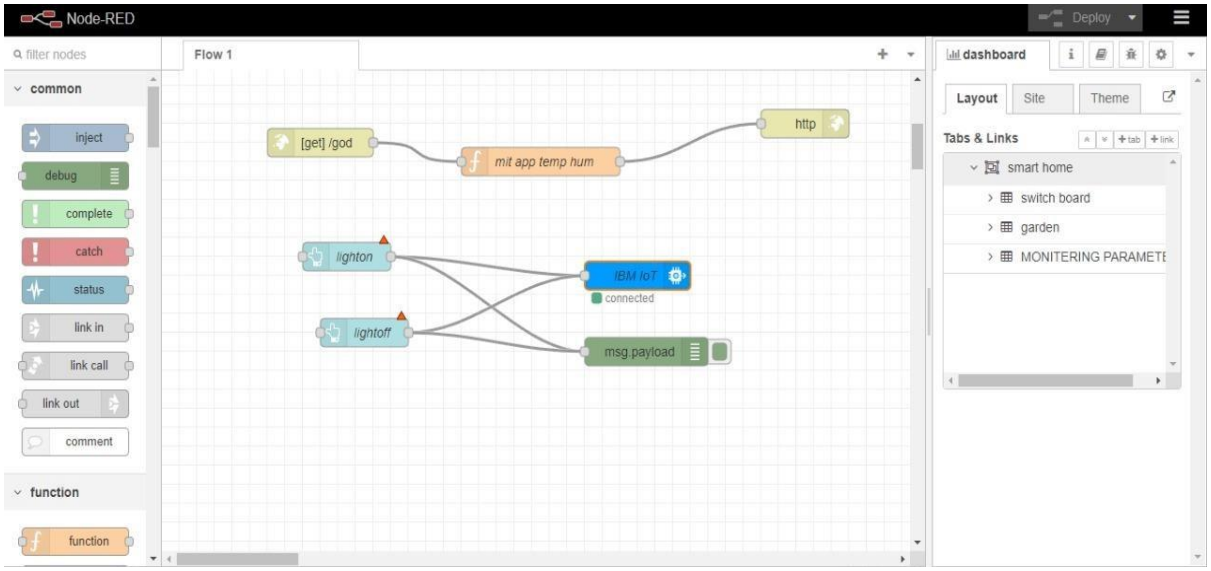
temp

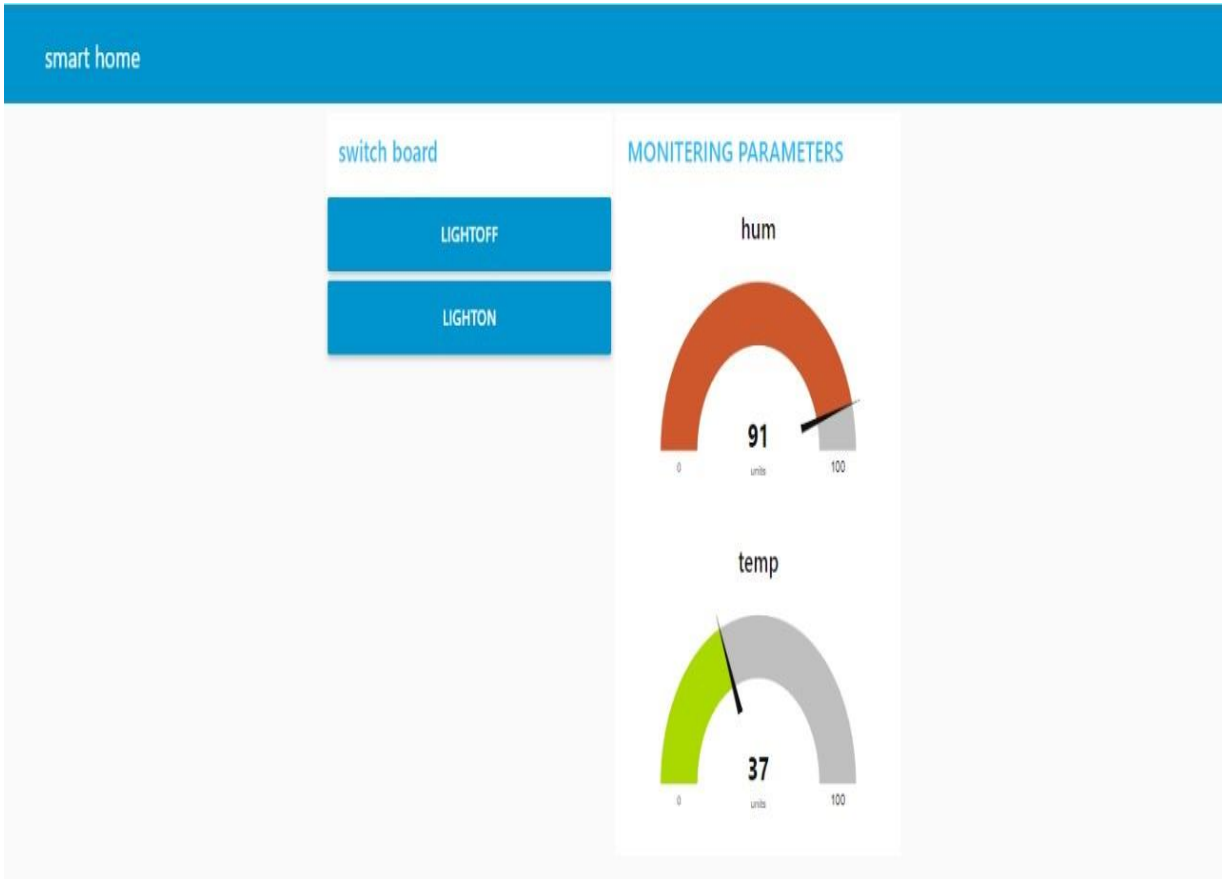
DELIVERY OF SPRINT-4:



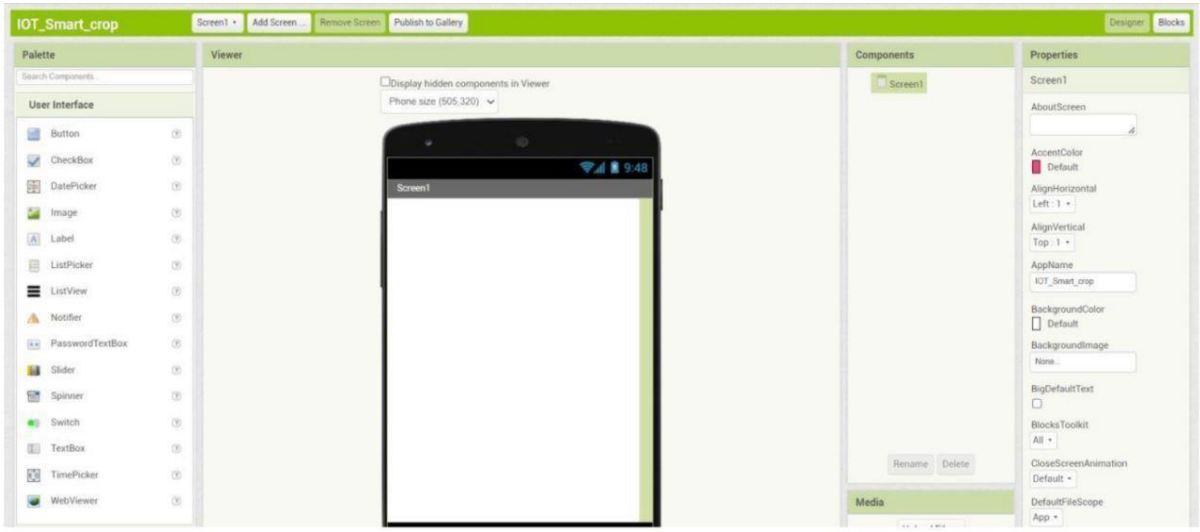
Step2: Customize the APP interface to display the values




Node-RED
Deploy
filter nodes
Flow 1
dashboard
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
[get] /god
mit app temp hum
http
lighton
lightoff
IBM IoT
connected
msg.payload
Layout
Site
Theme
Tabs & Links
smart home
switch board
garden
MONITERING PARAMETE


smart home
switch board
LIGHTOFF
LIGHTON
MONITERING PARAMETERS
hum
91
0
units
100
temp
37
0
units
100

Step1:**MIT APP** inventor to design the app.



Step2:Customize the APP interface to display the values.