ST.JOSEPH COLLEGE OF ENGINEERING

(Affiliated to AICTE & ANNA UNIVERSITY)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

REPORT ON

HX 8001 PROFESSIONAL READINESS FOR INNOVATION,

EMPLOYABILITY AND ENTREPENEURSHIP

(Nalaiya thiran program)

PROJECT TITLE

IoT Based Smart Crop Protection System For Agriculture

TEAM ID:PNT2022TMID26608

TEAM MEMBERS

1. ISHNU PRIYA.K(TEAM LEADER)
2. SUBANANDHINI.G
3. MONISHA.E
4. SADHANAPAUL.J

MENTOR

Mr.RATHANA SABAPATHY

EVALUATOR

Mrs.ANU.K

INDEX

1.INTRODUCTION

1.1 PROJECT OVREVIEW:

Crops in farms are many times ravagedby local animalslike buffaloes, cows, goats, birds etc. this leads to huge losses for the farmers. It is not possible for farmers tobarricade entire fields or stay on field 24 hours and guard it.so here we proposeautomatic crop protection system from animals. This is a microcontroller based system using PIC family microcontroller. 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 safetyof crop from animals thus protecting farmers loss.

1.2 PURPOSE:

Our main purpose 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 animal as well as human beings. Theme of project is to design a intelligent security system for farm protecting by using embedded system.

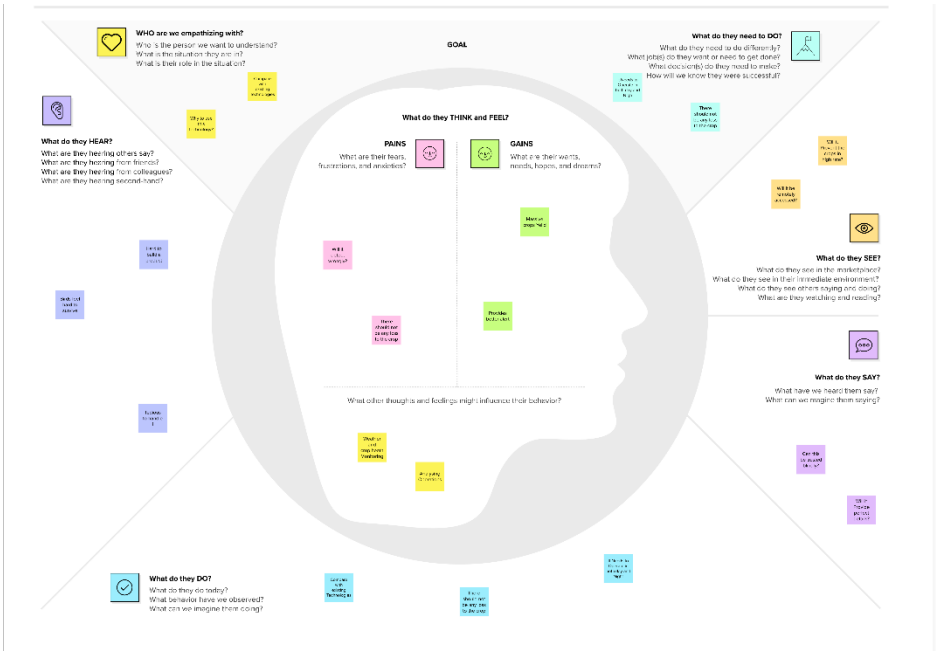
2.LITERATURE SURVEY

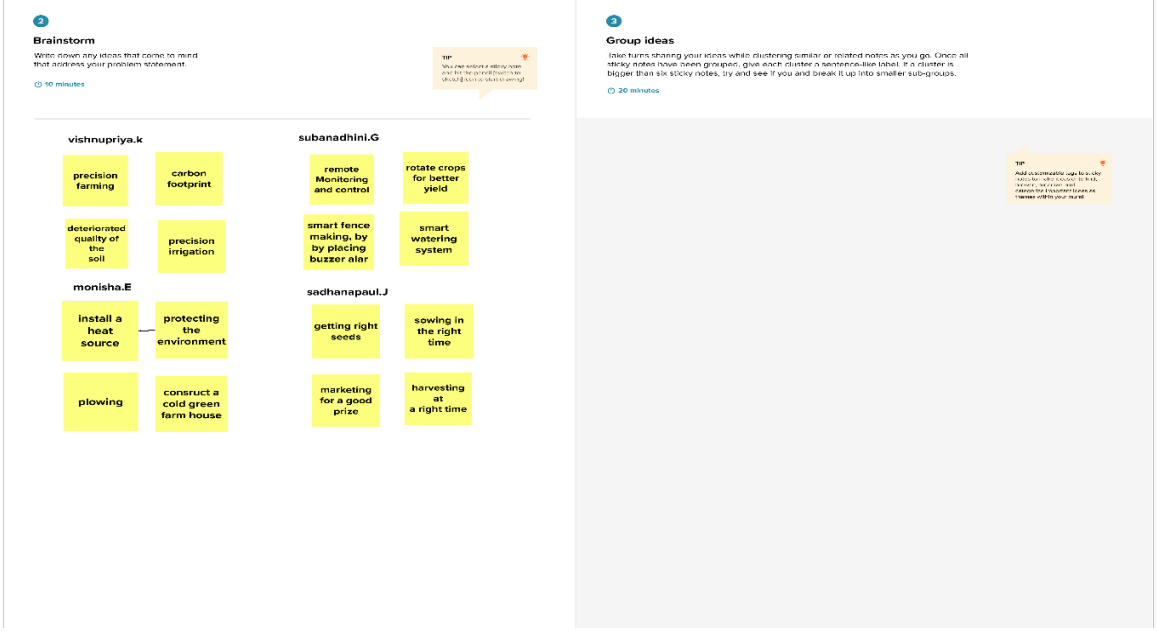
2. 1 EXISTING PROBLEM:

The existing system mainly provide 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 animal 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 andmanual surveillance and various such exhaustive and dangerous method.

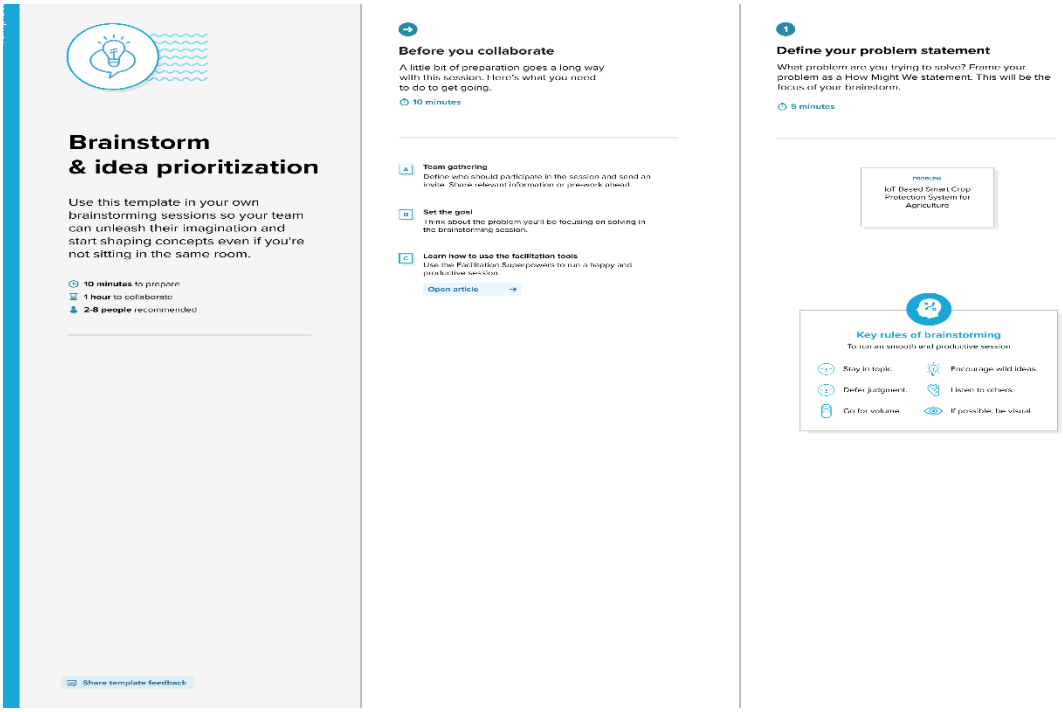
3.IDEATION AND PROPOSED SOLUTION

3.1. EMPATHY MAP CANVAS:





3.2 IDEATION AND BRAINSTORMING:

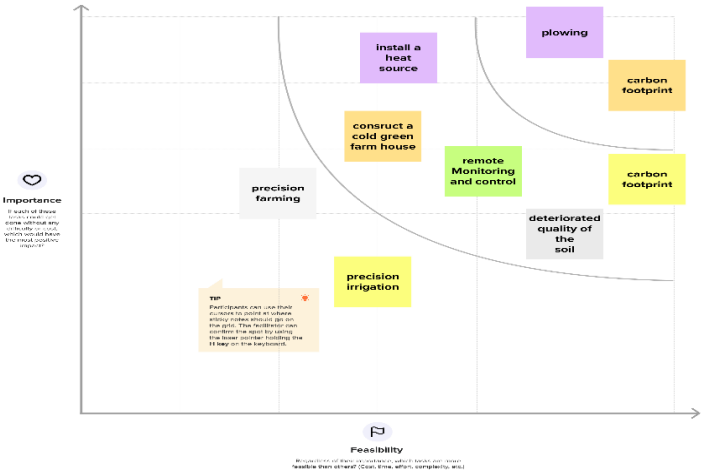


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

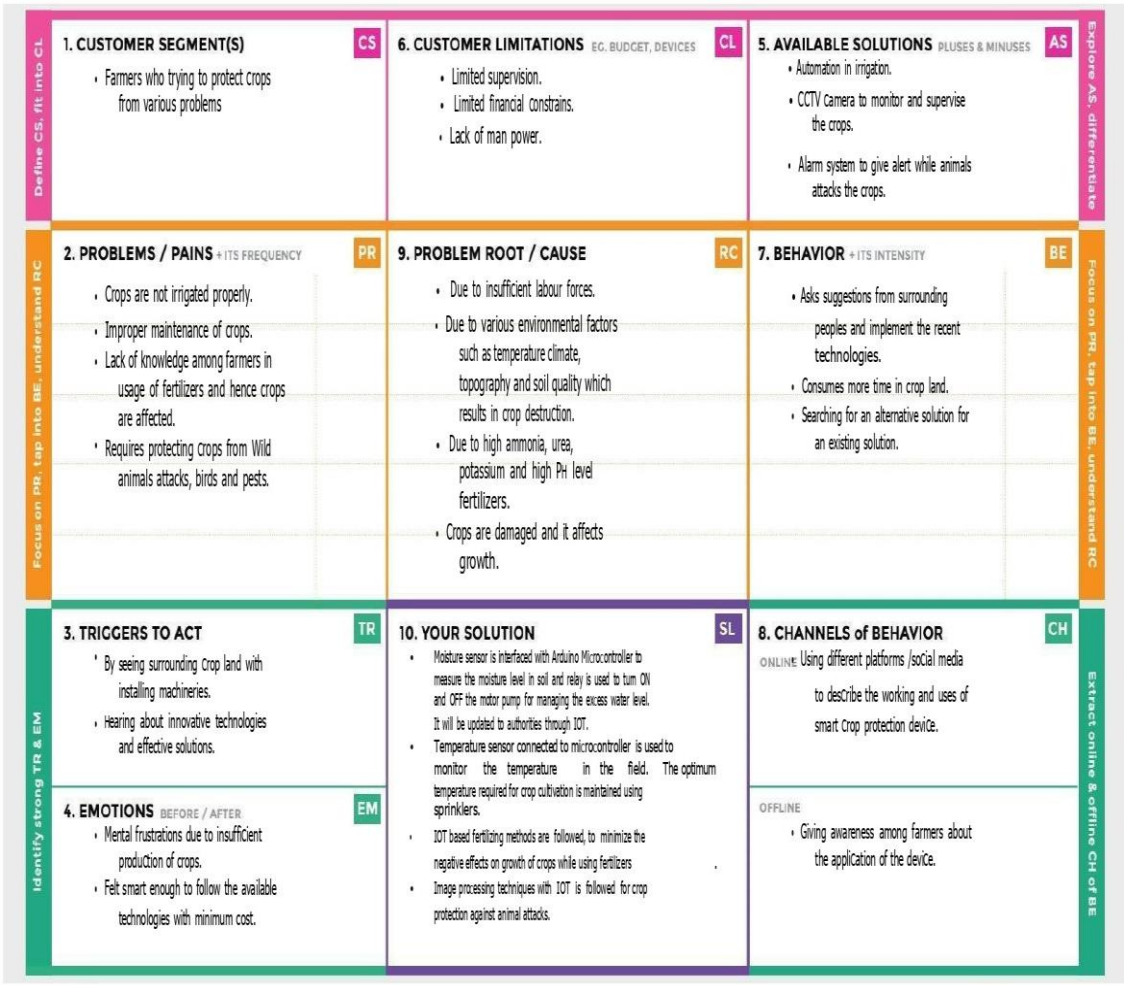
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3. PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Crops in farms are ravaged by local animals like buffaloes, cows, goats, birds etc. This leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field 24 hours and guard it.
2.	Idea / Solution description	Here we propose an automatic crop protection system from animals. This system uses a motion sensor to detect wild animals approaching near the field.
3.	Novelty / Uniqueness	Motion sensor, Temperature sensor, Humidity sensor, Moisture sensor, Alarm, GSM.
4.	Social Impact / Customer Satisfaction	Crop protection combines strategies, tools, and products that protect against various pests. These include diseases, viruses, weeds, and insects. All of them can significantly lower or even kill plants. The best decision is to control the situation by reducing the risks rather than deal with the problem's consequences.
5.	Business Model (Revenue Model)	Monitor the crop 24/7, Avoid animals, Check the weather condition, Alert the farmer.
6.	Scalability of the Solution	Scalability is an aspect or rather a functional quality o f a system, software or solution

3.4. Problem Solution fit


Define CS, fit into CL
1. CUSTOMER SEGMENT(S) CS
• Farmers who trying to protect crops from various problems
6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL
• Limited supervision.
• Limited financial constrains.
• Lack of man power.
5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS
• Automation in irrigation.
• CCTV camera to monitor and supervise the crops.
• Alarm system to give alert while animals attacks the crops.
Explore AS, differentiate
Focus on PR, tap into BE, understand RC
2. PROBLEMS / PAINS + ITS FREQUENCY PR
• Crops are not irrigated properly.
• Improper maintenance of crops.
• Lack of knowledge among farmers in usage of fertilizers and hence crops are affected.
• Requires protecting crops from Wild animals attacks, birds and pests.
9. PROBLEM ROOT / CAUSE RC
• Due to insufficient labour forces.
• Due to various environmental factors such as temperature climate, topography and soil quality which results in crop destruction.
• Due to high ammonia, urea, potassium and high PH level fertilizers.
• Crops are damaged and it affects growth.
7. BEHAVIOR + ITS INTENSITY BE
• Asks suggestions from surrounding peoples and implement the recent technologies.
• Consumes more time in crop land.
• Searching for an alternative solution for an existing solution.
Focus on PR, tap into BE, understand RC
Identify strong TR & EM
3. TRIGGERS TO ACT TR
• By seeing surrounding Crop land with installing machineries.
• Hearing about innovative technologies and effective solutions.
4. EMOTIONS BEFORE / AFTER EM
• Mental frustrations due to insufficient production of crops.
• Felt smart enough to follow the available technologies with minimum cost.
10. YOUR SOLUTION SL
• Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF the motor pump for managing the excess water level. It will be updated to authorities through IOT.
• Temperature sensor connected to microcontroller is used to monitor the temperature in the field. The optimum temperature required for crop cultivation is maintained using sprinklers.
• IOT based fertilizing methods are followed, to minimize the negative effects on growth of crops while using fertilizers .
• Image processing techniques with IOT is followed for crop protection against animal attacks.
8. CHANNELS of BEHAVIOR CH
ONLINE Using different platforms /social media to describe the working and uses of smart Crop protection device.
OFFLINE
• Giving awareness among farmers about the application of the device.
Extract online & offline CH of BE

4. REQUIREMENTS ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

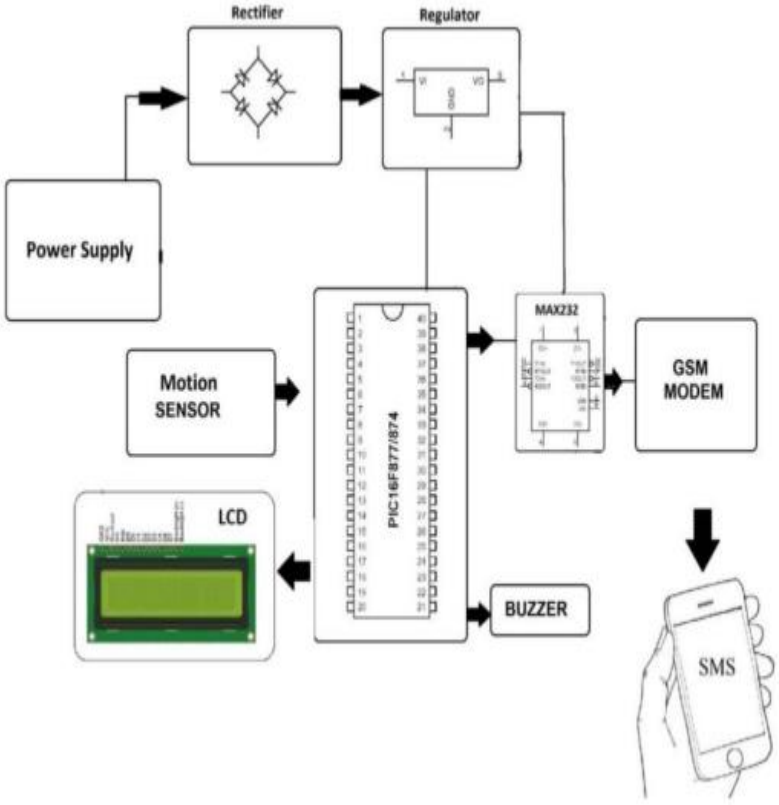
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
	Safety of production	The Smart Protection System identifies this initiative as aiding farmers in preserving land. The IOT gadget is used to warn the farmer when birds visit the farm and an SD card module is used to store a specific sound to make the animals afraid.
FR-2	Real time monitoring	Farm crops are frequently destroyed by neighbourhood animals including buffalo, cows, goats, birds, etc. The farmer suffers significant losses as a result. Deforestation caused by overpopulation leads to a lack of food, water, and shelter in forested areas. Therefore, animal intrusion into residential areas is growing daily, endangering human life and property and leading to human-animal conflict
FR-3	Eliminate man power	The device can be check the soil whether,it's wet or dry after checking in the device can be sent the message to there respective owner. Alarm system has been set to avoid conflicts
FR-4	Fast communication	This system uses a motion sensor to detect wild animals approaching near the field and smoke sensor to detect the fire. In such a case the sensor signals the microcontroller to take action. The microcontroller now sounds an alarm to woo the animals away from the field as well as sends SMS to the farmer and makes call, so that farmer may know about the issue and come to the spot in case the animals don't turn away by the alarm
FR-5	Performance	Using IOT network the sensor sends an message to the user
FR-6	Scalable Architecture	Justify the scalability of architecture.

4.2 NON FUNCTIONAL REQUIREMENT

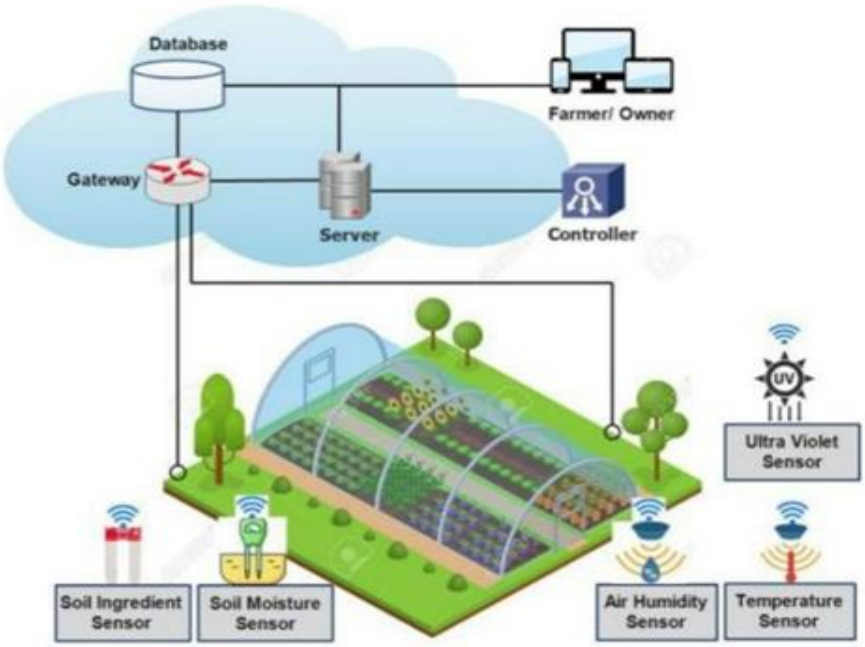
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability is a unique and significant perspective to examine user requirements, which may further enhance the design quality, according to IOT devices. Analysis of consumer product usability may help designers better understand users' prospective demands in gas leakage monitoring, behaviour, and experience in the design process where user experience is at the centre.
NFR-2	**Security**	It helps to prevent from material loss and human injuries
NFR-3	**Reliability**	Crop Protection System Using IOT to Prevent Bird and Wild Animal Attacks Using Arduino, a smart crop security device against wildlife Using Arduino, a smart crop protection system against fire and animals.
NFR-4	**Performance**	This device employs a motion sensor to find approaching wild animals close to the field and a smoke sensor to find a fire. The sensor instructs the microcontroller to operate in this situation.
NFR-5	**Availability**	This device employs a motion sensor to find approaching wild animals close to the field and a smoke sensor to find a fire. The sensor instructs the microcontroller to operate in this situation
NFR-6	**Scalability**	

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM:



5.2 Solution & Technical Architecture


Database
Farmer/ Owner
Gateway
Server
Controller
UV
Ultra Violet Sensor
Soil Ingredient Sensor
Soil Moisture Sensor
Air Humidity Sensor
Temperature Sensor

5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
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	Sprint-1
	Register	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user I can register for the application by entering my email, password and confirming my password	I can register & access the dashboard with Facebook Login	Low	Sprint-2
	Upload the image	USN-4	As a user I must upload the image to identify the problem and works on it		Medium	Sprint-1
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 compare to mobile user	Sprit-2

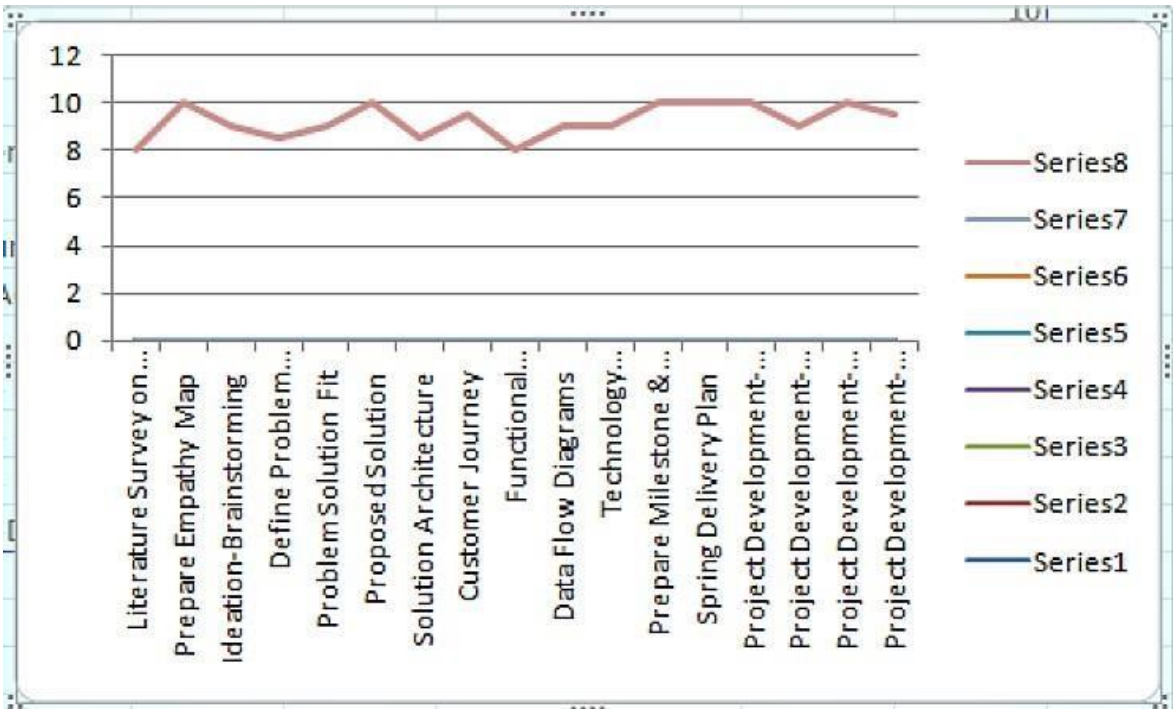
6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNINGAND ESTIMATION:

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	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 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 iteration unit (story points per day)

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



7. CODING & SOLUTIONING

Progra

m

```
Import cv;
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()
#Connecting to IBM watson.
deviceCli.conn
ect() while
True:
#Getting values from sensors.
temp_sensor = round(
random.uniform(0,80),2)
PH_sensor =
round(random.uniform(1,14),3)
camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
camera_reading = random.choice(camera)
flame = ["Detected","Not Detected","Not Detected","Not
Detected","Not Detected","Not Detected",] flame_reading =
random.choice(flame)
moist_level =
round(random.uniform(0,100),2
) water_level =
round(random.uniform(0,30),2)

#storing the sensor data to send in json
```

format to cloud. temp_data = {

'Temperature' : temp_sensor }

PH_data = { 'PH Level' : PH_sensor }

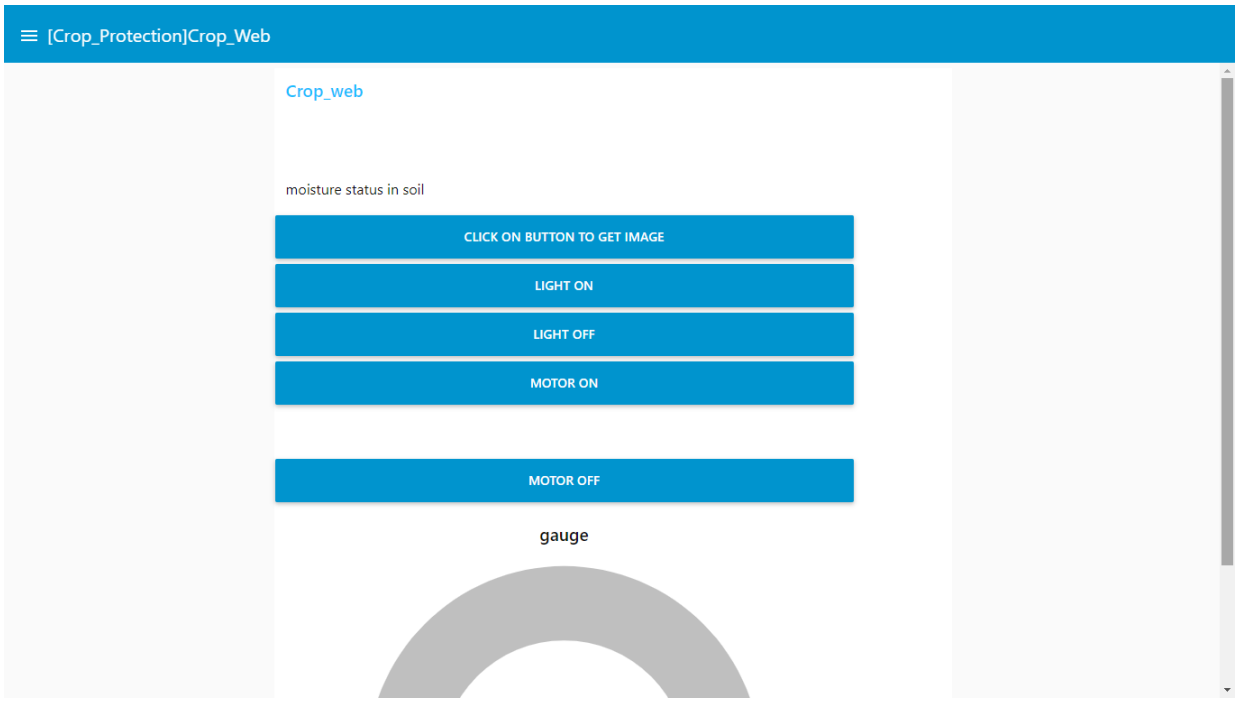
```
camera_data = { 'Animal attack' :
camera_reading} flame_data = {
'Flame' : flame_reading }
moist_data = { 'Moisture Level' :
moist_level} water_data = {
'Water Level' : water_level}

# publishing Sensor data to IBM Watson for every 5-10 seconds.
success = deviceCli.publishEvent("Temperature
sensor", "json", temp_data, qos=0) sleep(1)
if success:
  print (" .....publish ok.")
print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")

success = deviceCli.publishEvent("PH sensor",
"json", PH_data, qos=0) sleep(1)
if success:
  print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")

success = deviceCli.publishEvent("camera",
"json", camera_data, qos=0) sleep(1)
if success:
  print ("Published Animal attack %s " %
camera_reading, "to IBM Watson") success =
deviceCli.publishEvent("Flame sensor", "json",
flame_data, qos=0) sleep(1)
if success:
  print ("Published Flame %s " % flame_reading, "to IBM Watson")

success = deviceCli.publishEvent("Moisture sensor",
"json", moist_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 ("")
#Automation to control sprinklers by present temperature an to send alert message to IBM Watson.

if (temp_sensor > 35):
  print("sprinkler-1 is ON")
success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is high, sprinkerlers are
turned ON" %temp_sensor }
,
qos=
0)
sleep
(1)
if success:
  print( 'Published alert1 : ', "Temperature(%s) is high, sprinkerlers are turned
ON" %temp_sensor,"to IBM Watson")  print("")
else:
print("sprinkler-
1 is OFF")
print("")

#To send alert message if farmer uses the
unsafe fertilizer to crops. if (PH_sensor >
7.5 or PH_sensor < 5.5):
  success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use other
  fertilizer" %PH_sensor } ,
qos=
0)
sleep
(1)
if success:
  print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other
fertilizer" %PH_sensor,"to IBM Watson")  print("")
```

```
#To send alert message to farmer that

animal attack on crops. if

(camera_reading == "Detected"):

  success = deviceCli.publishEvent("Alert3", "json", { 'alert3' :
"Animal attack on crops detected" }, qos=0) sleep(1)

if success:

  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")


#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("")

#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("")
#command recived by farmer deviceCli.commandCallback = myCommandCallback
device disconnect()
```

OUTPUT:



7.1 Features 1:

Output: Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors

and capacitors on the PCB and differ from sensor to sensor. Power supply: 5V-12V input voltage for most modules (they have a 3.3V regulator),but 5V is ideal in case the regulator has different specs.

BUZZER

Specifications

- RatedVoltage : 6V DC
- Operating Voltage : 4 to 8V DC
- Rated Current*: ≤30mA
- SoundOutput at 10cm* : ≥85dB
- Resonant Frequency : 2300 ±300Hz
- Tone: Continuous A buzzer is a loud noise maker.

Most modern ones are civil defense or air- raid sirens, tornado sirens, or the sirens on emergency service vehiclessuch as ambulances, police cars and fire trucks. There are two general types, pneumatic and electronic.

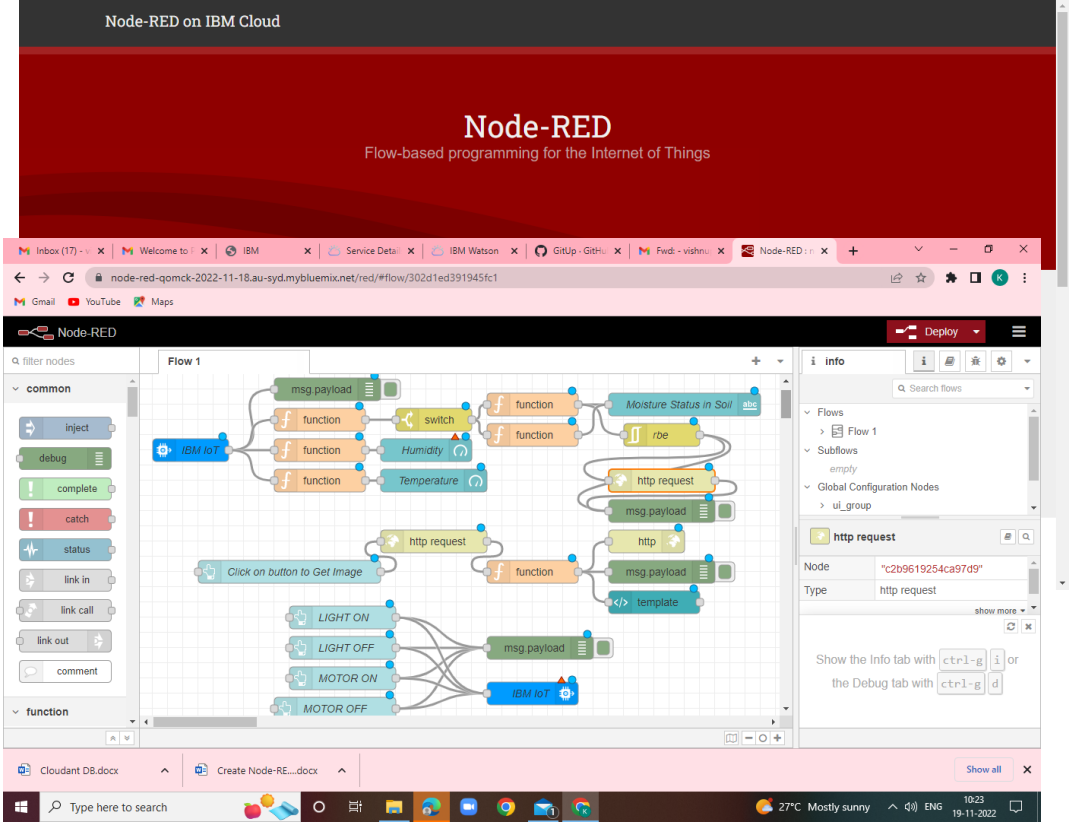
7.2 Features 2 :

i. Goodsensitivity to Combustible gas in wide range .
ii. Highsensitivity to LPG, Propane and Hydrogen .
iii. Longlife and low cost.
iv. Simpledrive circuit.

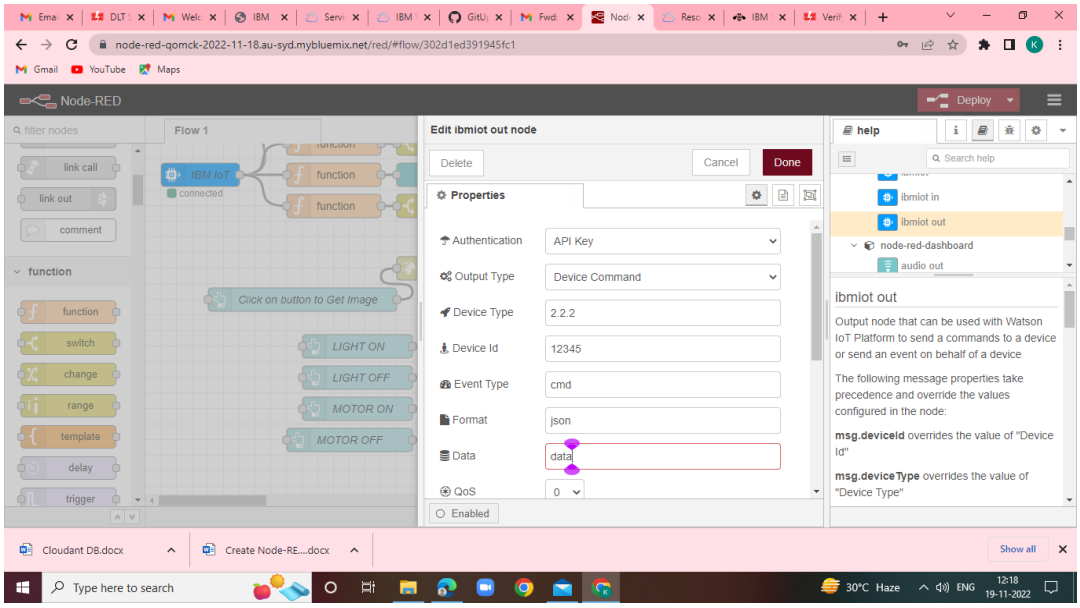
8. TESTING

8.1 TEST CASES:

sno	Parameter	Values
1	Model summary	-
2	Accuracy	Training accuracy-95% Validationaccuracy-72%
3	Confidence score	Class detected-80% Confidencescore-80%



8.2 User Acceptance Testing:


node-red-qomck-2022-11-18.au-syd.mybluemix.net/red/#flow/302d1ed391945fc1
Node-RED
Deploy
Flow 1
IBM IoT
connected
function
function
Click on button to Get Image
LIGHT ON
LIGHT OFF
MOTOR ON
MOTOR OFF
link call
link out
comment
function
switch
change
range
template
delay
trigger
Edit ibmiot out node
Delete
Cancel
Done
Properties
Authentication
API Key
Output Type
Device Command
Device Type
2.2.2
Device Id
12345
Event Type
cmd
Format
json
Data
data
QoS
0
Enabled
help
ibmiot in
ibmiot out
node-red-dashboard
audio out
ibmiot out
Output node that can be used with Watson IoT Platform to send a commands to a device or send an event on behalf of a device
The following message properties take precedence and override the values configured in the node:
msg.deviceId overrides the value of "Device Id"
msg.deviceType overrides the value of "Device Type"
Cloudant DB.docx
Create Node-RE....docx
Show all
Type here to search
30°C Haze
ENG
12:18
19-11-2022

9. RESULTS

The problem of crop vandalization by wild animals and fire has become a major social problem in current time.

It requires urgent attention as no effective solution exists till date for this problem. Thus this project carries a great social relevance as it aims to address this problem. This project willhelp farmers in protecting their orchards and fields and save them from significant financial losses and will save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achievingbetter crop yields thus leading to theireconomic wellbeing.

10. ADVANTAGES AND DISADVANTAGES

Advantage:

Controllable food supply. you might have droughts or floods, but if you are growing the crops and breeding them to be hardier, you have a better chanceof not straving. It allows farmers to maximize yields using minimum resources such as water,fertilizers.

Disadvantage: The main disadvantage is the time it can take to process the information.in order to keep feeding people as the population grows you have to radically change theenvironment of the planet

11. CONCLUSION

A IoT Web Application is built for smart agricultural system using Watson IoT platform, Watsonsimulator, IBM cloud and Node-RED

12. FUTURE SCOPE

In the future, there will be very large scope, this project can be made based on Image processing in which wild animaland fire can be detected by cameras and if it comes towards farmthen system will be directly activated through wireless networks. Wild animals can also be detected by using wireless networks such as laser wireless sensors and by sensing this laser or sensor's security system will beactivated.

13. APPENDIX

13.1 SOURCE CODE

```
Import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
organization = "op701j"
deviceType = "Lokesh"
deviceId = "Lokesh89"
authMethod = "token"
authToken = "1223334444"
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="sprinkler_on":
    print ("sprinkler is ON")
 else :
    print ("sprinkler is OFF")
 #print(cmd)



try:
 deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-tok
 deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
sys.exit()
#Connecting to IBM watson.
deviceCli.connect()
while True:
#Getting values from sensors.
 temp_sensor = round( random.uniform(0,80),2)
 PH_sensor = round(random.uniform(1,14),3)
 camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
 camera_reading = random.choice(camera)
 flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
 flame_reading = random.choice(flame)
 moist_level = round(random.uniform(0,100),2)
```

```
 print ("")
 #Automation to control sprinklers by present temperature an to send alert message to IBM Watson.


 if (temp_sensor > 35):
    print("sprinkler-1 is ON")
  success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is high, sprinkerlers are turned O
, qos=0)
 sleep(1)
 if success:
     print( 'Published alert1 : ', "Temperature(%s) is high, sprinkerlers are turned ON" %temp_sensor,"to IBM Wats
 print("")
else:
 print("sprinkler-1 is OFF")
 print("")


 #To send alert message if farmer uses the unsafe fertilizer to crops.


 if (PH_sensor > 7.5 or PH_sensor < 5.5):
    success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use other fertil
qos=0)
 sleep(1)


 If success:
print (" .....publish ok ..... ")
print ("Published Temperature = %s C" % temp_sensor, "to IBM Watson")


success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
sleep(1)
if success:
  print ("Published PH Level = %s" % PH_sensor, "to IBM Watson")
```

GITHUB LINK:

IBM-Project-36062-1660292249

PROJECT DEMO LINK:

https://youtu.be/yJg1D0C2K-c