IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICALTURE

Team id:PNT2022TMID39999

SUBMITTED BY

KALPANA.K	511519104011
OVIYA.V	511519104023
KARTHIGA K	511519104710
SANGEETHA S	511519104027

In partial fulfilment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE& ENGINEERING
P.T.LEE CNCET COLLEGE OF ENGINEERING AND TECHNOLOGY

CHAPTER-1

INTRODUCTION

1.1 PROJECT OVERVIEW

Crops in farms are many times 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.so here we propose automatic 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 safety of 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.

CHAPTER-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.

2.2 REFERENCES

1. 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

2. Mohit Korche,Sarthak Tokse, ShubhamShirbhate, Vaibhav Thakre,S. P. Jolhe(HOD). Students , Final Year,Dept.of Electrical engineering,Government

3. 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.

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

2.3 PROBLEM STATEMENT DEFINITION

This project describes the method of tracking the crops and protecting the crops from the inserts 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

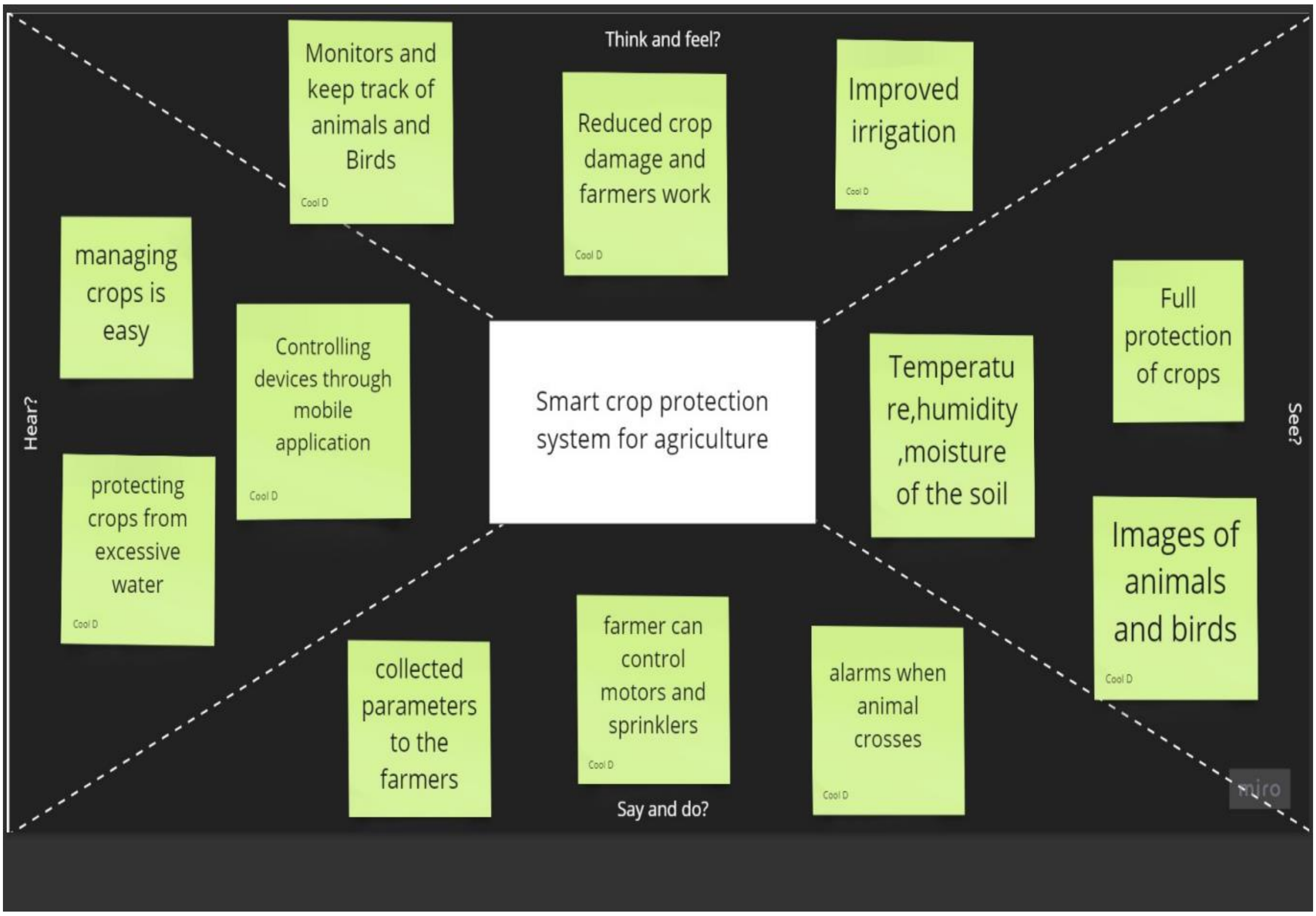
Satellites (for imagery and positioning) are few key technologies characterizing the precision Agriculture trend.Precision agriculture is one of the most famous applications of IoT in the agricultural sector And numerous organizations are leveraging this technique around the world.IoT has been making deep inroads into sectors such as manufacturing, health-care and Automotive. When it comes to food production, transport and storage, it offers a breadth of Options that can improve India’s per capita food availability. Sensors that offer information On soil nutrient status, pest infestation, moisture conditions etc. which can be used to improve Crop yields over time.In Vidarbha region, Main Cash Crops such as Pigeon Pea, GreenGram, Black Gram, Jowar, Cotton, Soybean etc. present and are Badly affected by wild animals like Deer, Rohi (Neel Gai), wild Pigs,Peacock etc. In few districts in Vidarbha crop loss is more than 35%. Main Wild animals attacking crops in region are Akola, BuldhanaWashim etc.

In spite of economic development agriculture is the backbone of the economy. Crops in forms are many times ravaged by local animals like buffaloes, cows, goats, birds and fire etc. this leads to huge loss for the farmers.it is not possible for farmers to blockade to entire fields or stay 24 hours and guard it. Agriculture meetsfood requirements of the people and produces several raw materialsfor industries. But because of animal interference and fire in agricultural lands, there will be huge loss of crops.Crops will be totally getting destroyed

CHAPTER-3

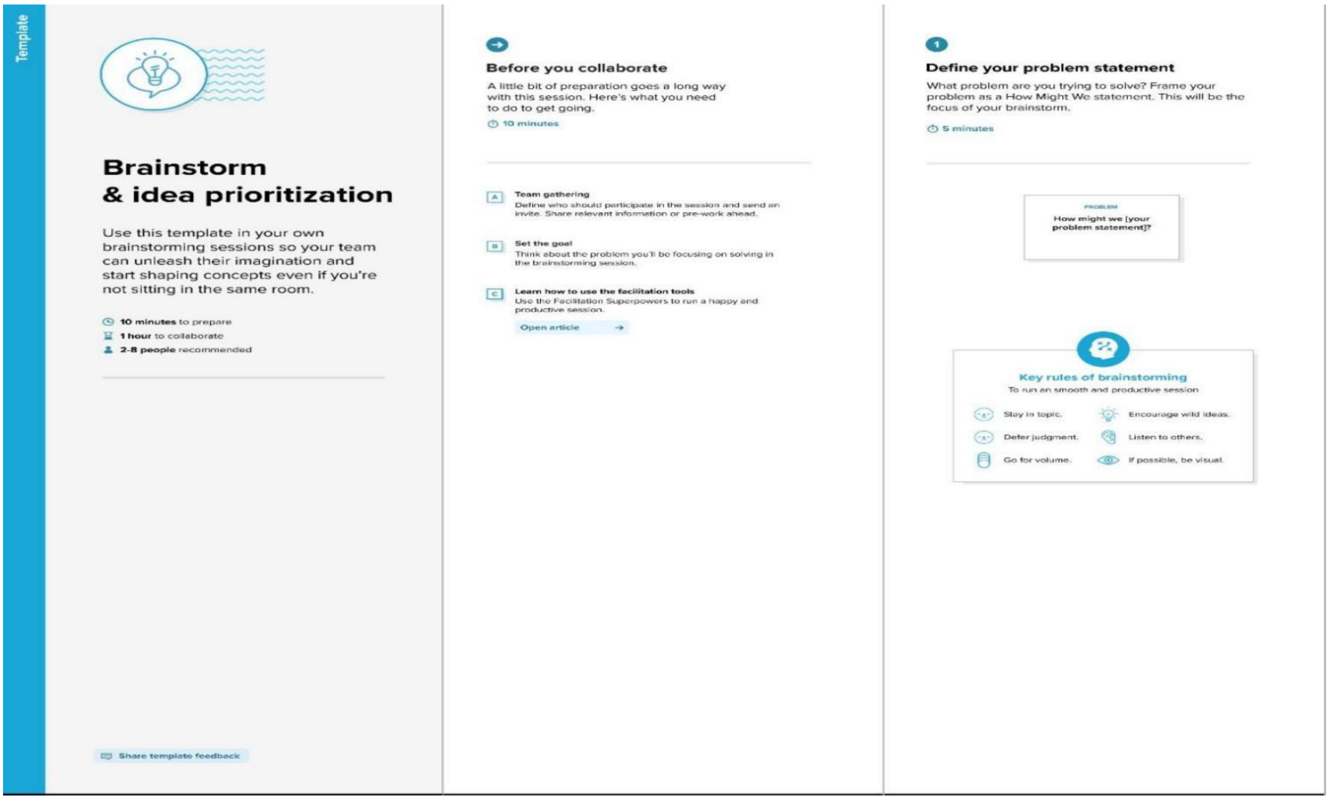
IDEATION AND PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS



Think and feel?
Monitors and keep track of animals and Birds
Cool D
Reduced crop damage and farmers work
Cool D
Improved irrigation
Cool D
Hear?
managing crops is easy
Controlling devices through mobile application
Cool D
protecting crops from excessive water
Cool D
Smart crop protection system for agriculture
Temperatu re,humidity ,moisture of the soil
Full protection of crops
See?
Images of animals and birds
Cool D
collected parameters to the farmers
farmer can control motors and sprinklers
Cool D
alarms when animal crosses
Cool D
Say and do?
miro

3.2 IDEATION AND BRAINSTORMING



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

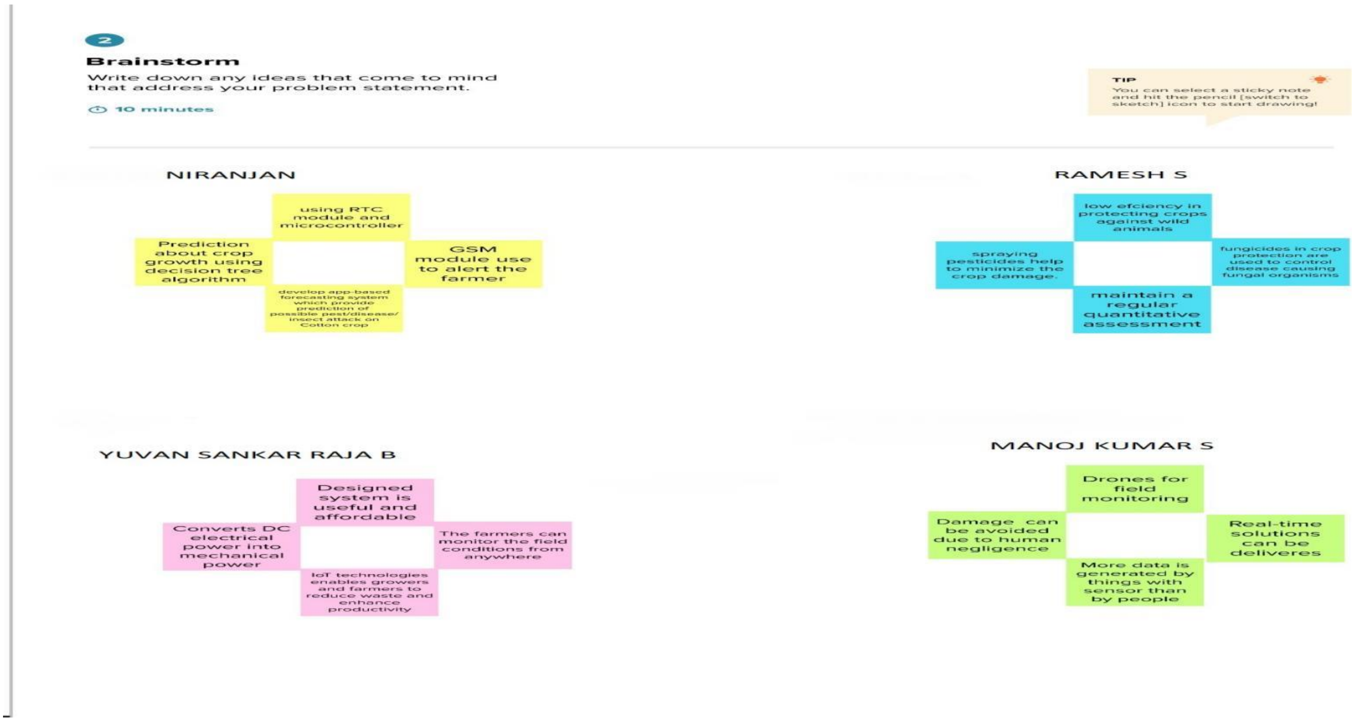
PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

NIRANJAN

- using RTC module and microcontroller
- Prediction about crop growth using decision tree algorithm
- GSM module use to alert the farmer
- develop app-based forecasting system which provide prediction of possible pest/disease/ insect attack on Cotton crop

RAMESH S

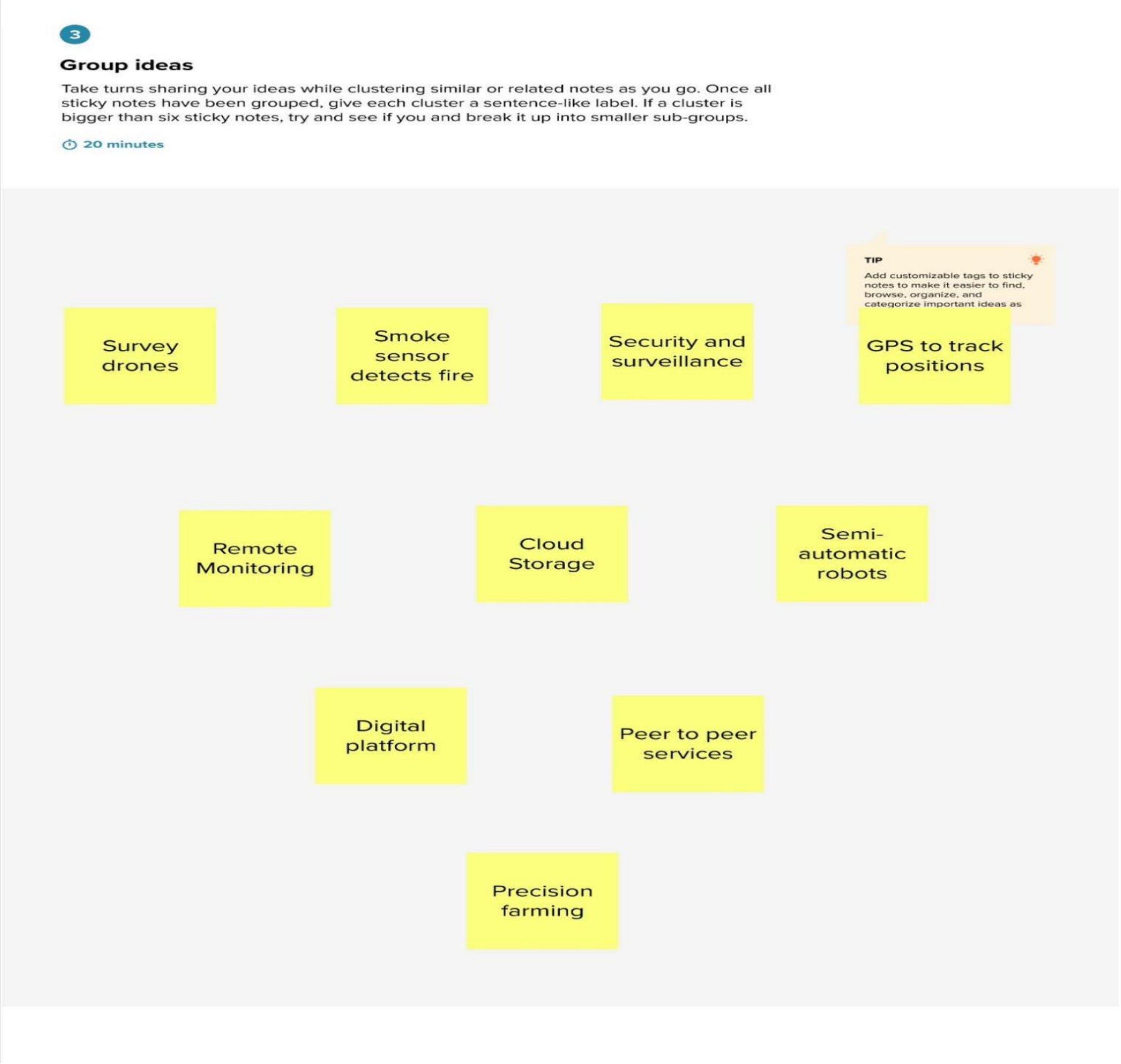
- low efciency in protecting crops against wild animals
- spraying pesticides help to minimize the crop damage.
- fungicides in crop protection are used to control disease causing fungal organisms
- maintain a regular quantitative assessment

YUVAN SANKAR RAJA B

- Designed system is useful and affordable
- Converts DC electrical power into mechanical power
- The farmers can monitor the field conditions from anywhere
- IoT technologies enables growers and farmers to reduce waste and enhance productivity

MANOJ KUMAR S

- Drones for field monitoring
- Damage can be avoided due to human negligence
- Real-time solutions can be deliveres
- More data is generated by things with sensor than by people



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as

Survey drones

Smoke sensor detects fire

Security and surveillance

GPS to track positions

Remote Monitoring

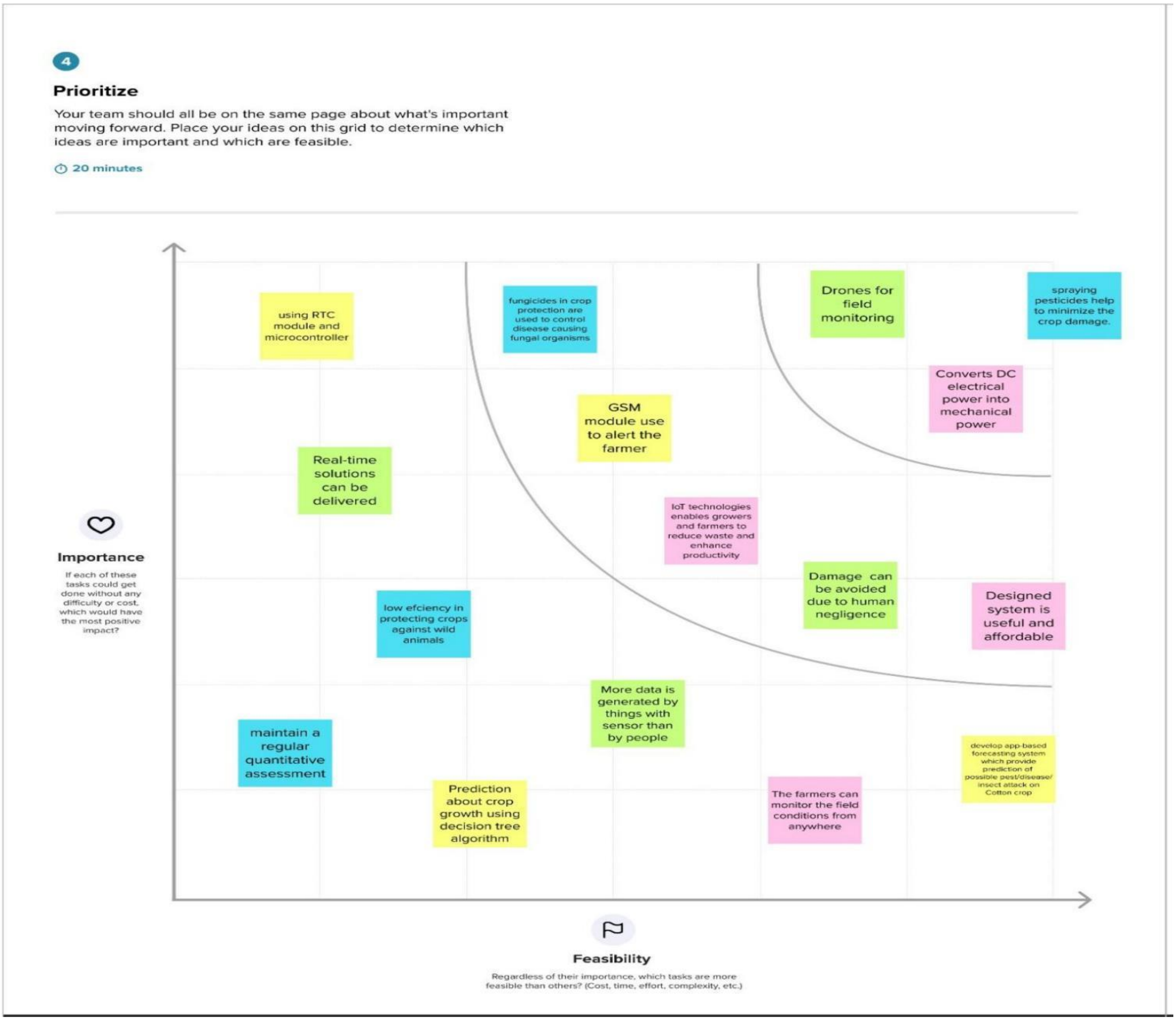
Cloud Storage

Semi-automatic robots

Digital platform

Peer to peer services

Precision farming



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

using RTC module and microcontroller

fungicides in crop protection are used to control disease causing fungal organisms

Drones for field monitoring

spraying pesticides help to minimize the crop damage.

Converts DC electrical power into mechanical power

GSM module use to alert the farmer

Real-time solutions can be delivered

IoT technologies enables growers and farmers to reduce waste and enhance productivity

Damage can be avoided due to human negligence

Designed system is useful and affordable

low efciency in protecting crops against wild animals

More data is generated by things with sensor than by people

maintain a regular quantitative assessment

develop app-based forecasting system which provide prediction of possible pest/disease/ insect attack on Cotton crop

Prediction about crop growth using decision tree algorithm

The farmers can monitor the field conditions from anywhere

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

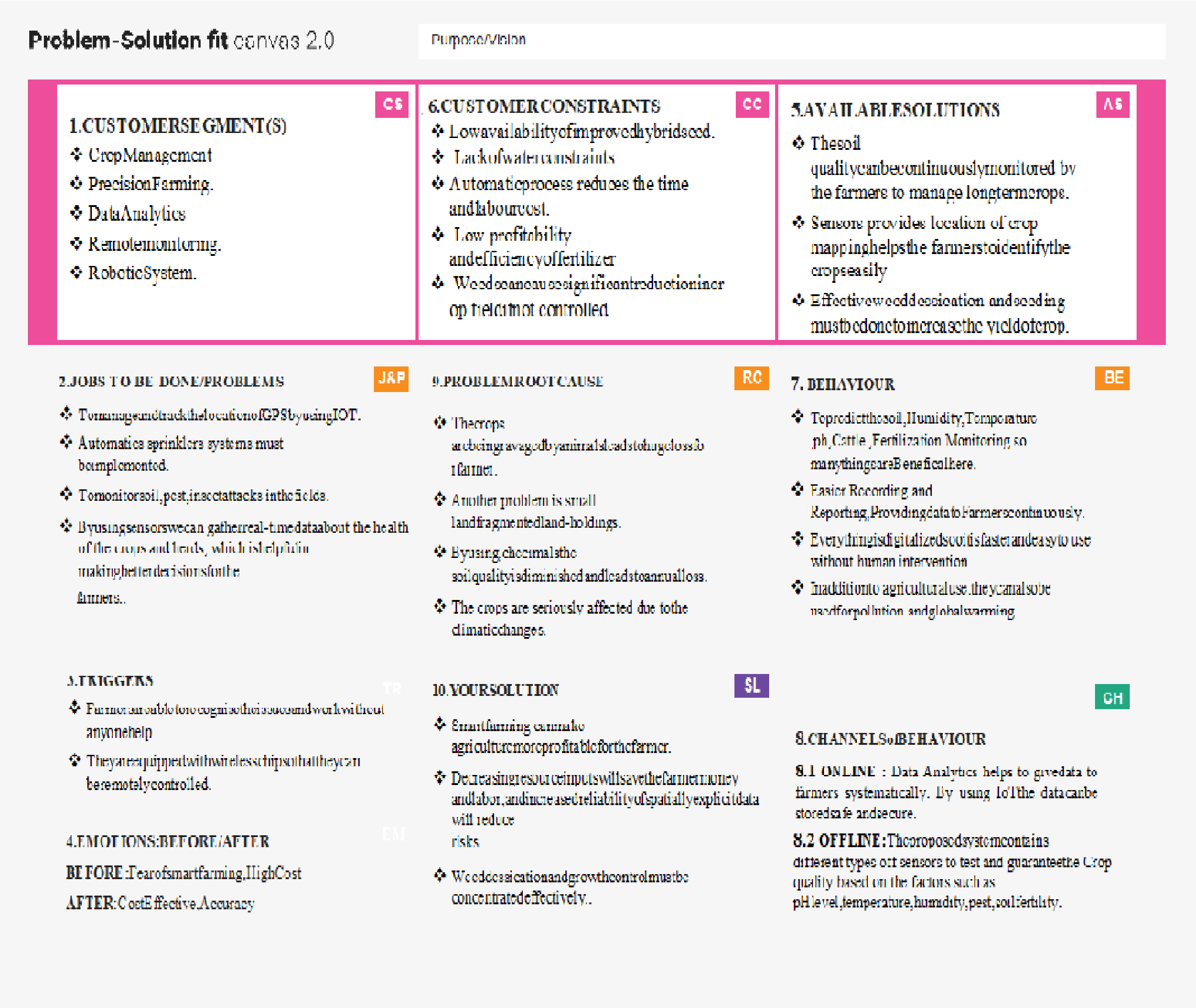
Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	ProblemStatement (Problemtobe solved)	Develop an efficient system & an application that can monitor and alert the users(farmers)
2.	Idea/Solution description	➢ This product helps the field in monitoring the animals other disturbance ➢ In several areas, the temperature sensors will be integrated to monitor the temperature & humidity ➢ If in any area feel dry or wet is detected by admins, will be notified along with the location in the web application
3.	Novelty/Uniqueness	➢ Fastest alerts to the farmers ➢ The increasing demand for quality food □User friendly
4.	Social Impact/Customer Satisfaction	➢ Easy installation and provide efficient results ➢ Can work with irrespective of fear
5.	Business Model(Revenue Model)	➢ As the product usage can be understood by everyone, it is easy for them to use it properly for their safest organization ➢ The product is advertised all over the platforms. Since it is economical, even helps small scale farming land from disasters.
6.	Scalability of the Solution	□Even when the interruption is more, the product sense the accurate location and alerts the farmers effectively

3.4 PROBLEM SOLUTION FIT


CS
CC
AS
J&P
RC
BE
TR
EM
SL
CH

Problem-Solution fit canvas 2.0

Purpose/Vision

1.CUSTOMERSEGMENT(S)

- CropManagement
- PrecisionFarming.
- DataAnalytics
- Remotemonitoring.
- RoboticSystem.

6.CUSTOMERCONSTRAINTS

- Lowavailabilityofimprovedhybridseed.
- Lackofwaterconstraints
- Automaticprocess reduces the time andlabourcost.
- Low profitability andefficiencyoffertilizer
- Weedcancausesignificantreductionincrop fieldnot controlled

5.AVAILABLESOLUTIONS

- Thesoil qualitycanbecontinuouslymonitored by the farmers to manage longtermcrops.
- Sensors provides location of crop mappinghelpsthe farmerstoidentifythe cropseasily
- Effectiveweeddessication andseeding mustbedonetoincreasethe yieldofcrop.

2.JOBS TO BE DONE/PROBLEMS

- TomanageandtrackthelocationofGPSbyusingIOT.
- Automatics sprinklers systems must beimplemented.
- Tomonitorsoil,pest,insectattacks inthe fields.
- Byusingsensorswecan gatherreal-timedataabout the health of the crops and lands, which ishelpfulin makingbetterdecisionsforthe farmers..

9.PROBLEMROOTCAUSE

- Thecrops arebeingravagedbyanimalsleadstohugelossfo rfarmer.
- Another problem is small landfragmentedland-holdings.
- Byusing,chemicalsthe soilqualityisdiminishedandleadstoannualloss.
- The crops are seriously affected due tothe climaticchanges.

7. BEHAVIOUR

- Topredictthesoil,Humidity,Temperature ,ph,Cattle ,Fertilization Monitoring so manythingsareBeneficalhere.
- Easier Recording and Reporting,ProvidingdatatoFarmerscontinuously.
- Everythingisdigitalizedsoitisfasterandeasyto use without human intervention
- Inadditionto agriculturaluse,theycanalsobe usedforpollution andglobalwarming

3.TRIGGERS

- Farmersareabletorecognisetheissueandworkwithout anyonehelp
- Theyareequippedwithwirelesschipsothattheycan beremotelycontrolled.

4.EMOTIONS:BEFORE/AFTER

BEFORE:Fearofsmartfarming,HighCost

AFTER:CostEffective,Accuracy

10.YOURSOLUTION

- Smartfarming canmake agriculturemoreprofitableforthefarmer.
- Decreasingresourceinputswillsavethefarmermoney andlabor,andincreasedreliabilityofspatiallyexplicitdata will reduce risks
- Weeddessicationandgrowthcontrolmustbe concentratedeffectively..

8.CHANNELSofBEHAVIOUR

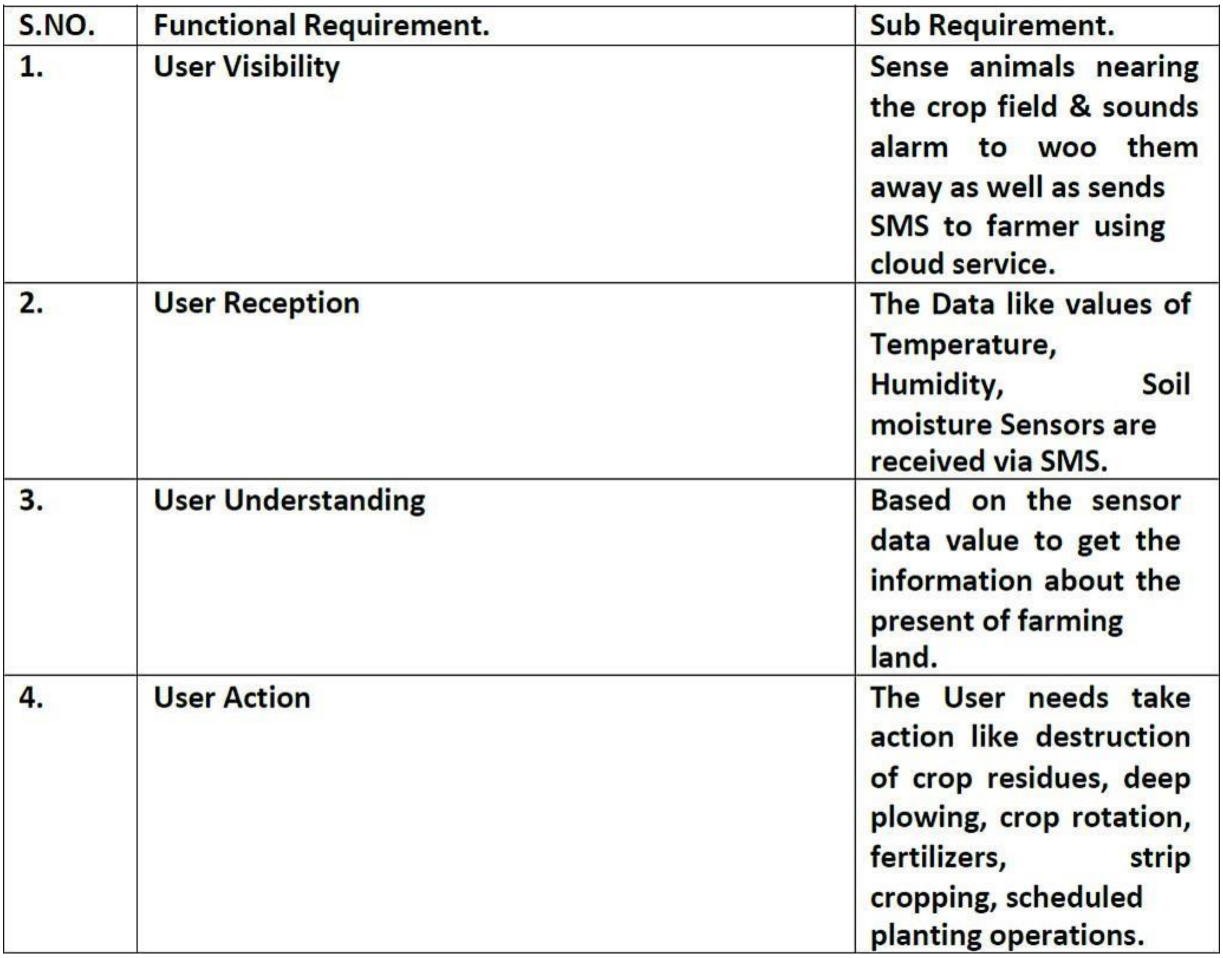
8.1 ONLINE : Data Analytics helps to givedata to farmers systematically. By using IoTthe datacanbe storedsafe andsecure.

8.2 OFFLINE: Theproposedsystemcontains different types off sensors to test and guaranteethe Crop quality based on the factors such as pHlevel,temperature,humidity,pest,soilfertility.

CHAPTER-4

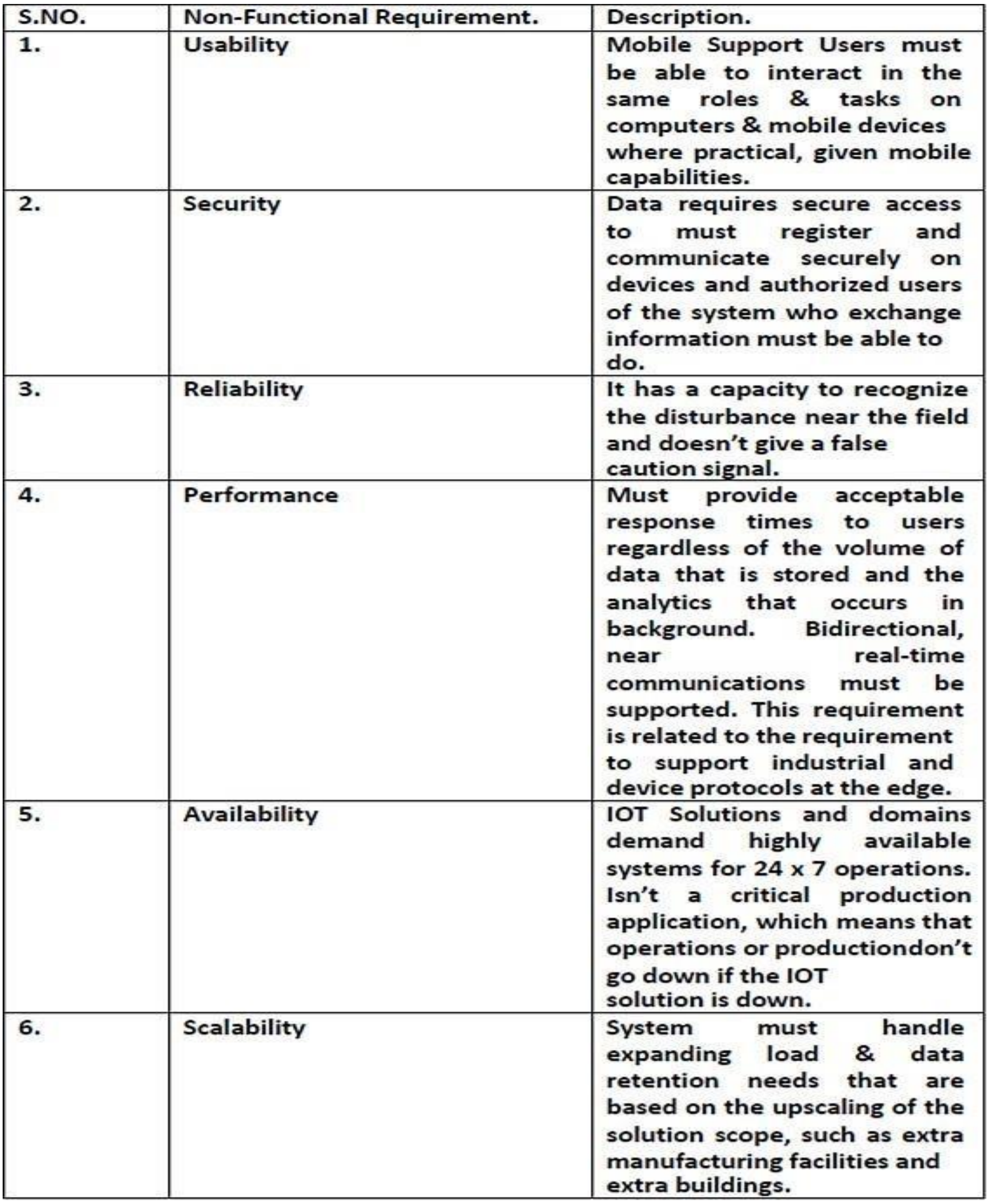
REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT



S.NO.	Functional Requirement.	Sub Requirement.
1.	User Visibility	Sense animals nearing the crop field & sounds alarm to woo them away as well as sends SMS to farmer using cloud service.
2.	User Reception	The Data like values of Temperature, Humidity, Soil moisture Sensors are received via SMS.
3.	User Understanding	Based on the sensor data value to get the information about the present of farming land.
4.	User Action	The User needs take action like destruction of crop residues, deep plowing, crop rotation, fertilizers, strip cropping, scheduled planting operations.

4.2 NON-FUNCTIONAL REQUIREMENT

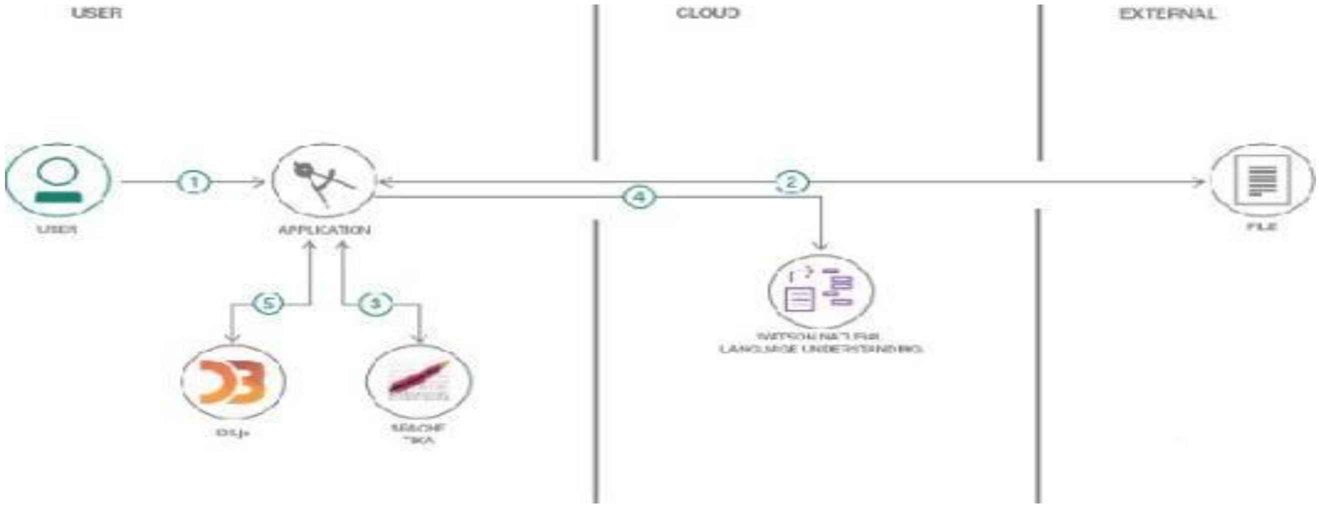


S.NO.	Non-Functional Requirement.	Description.
1.	Usability	Mobile Support Users must be able to interact in the same roles & tasks on computers & mobile devices where practical, given mobile capabilities.
2.	Security	Data requires secure access to must register and communicate securely on devices and authorized users of the system who exchange information must be able to do.
3.	Reliability	It has a capacity to recognize the disturbance near the field and doesn't give a false caution signal.
4.	Performance	Must provide acceptable response times to users regardless of the volume of data that is stored and the analytics that occurs in background. Bidirectional, near real-time communications must be supported. This requirement is related to the requirement to support industrial and device protocols at the edge.
5.	Availability	IOT Solutions and domains demand highly available systems for 24 x 7 operations. Isn't a critical production application, which means that operations or productiondon't go down if the IOT solution is down.
6.	Scalability	System must handle expanding load & data retention needs that are based on the upscaling of the solution scope, such as extra manufacturing facilities and extra buildings.

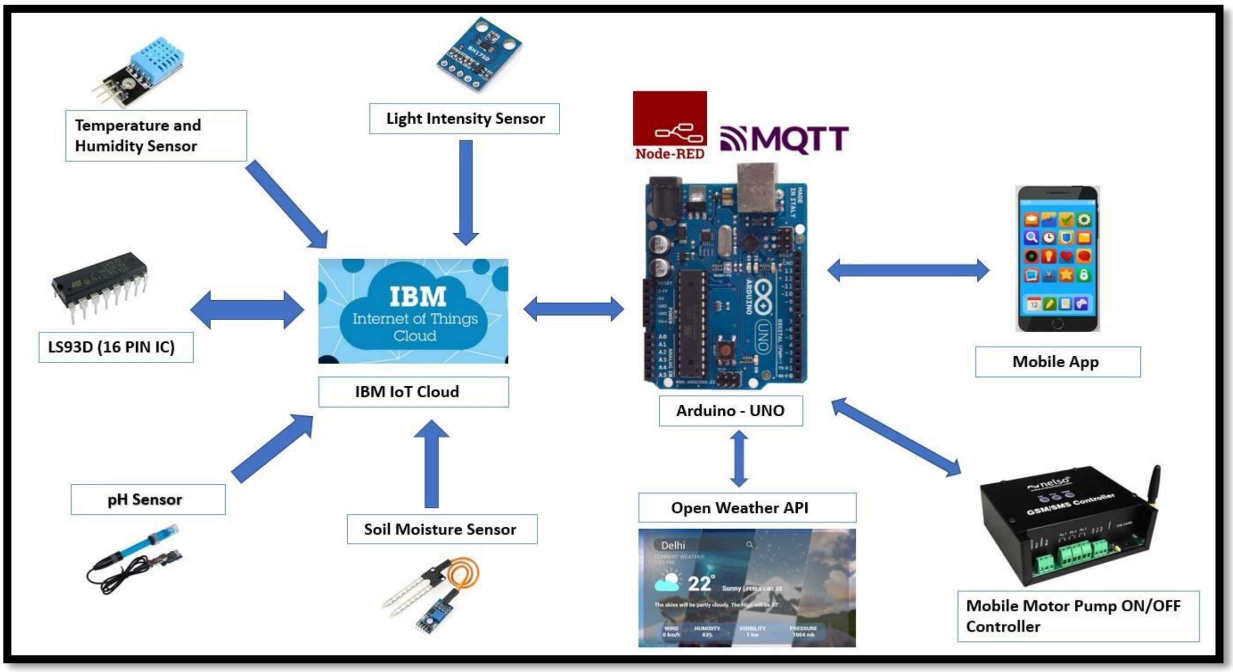
CHAPTER-5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3USER STORIES

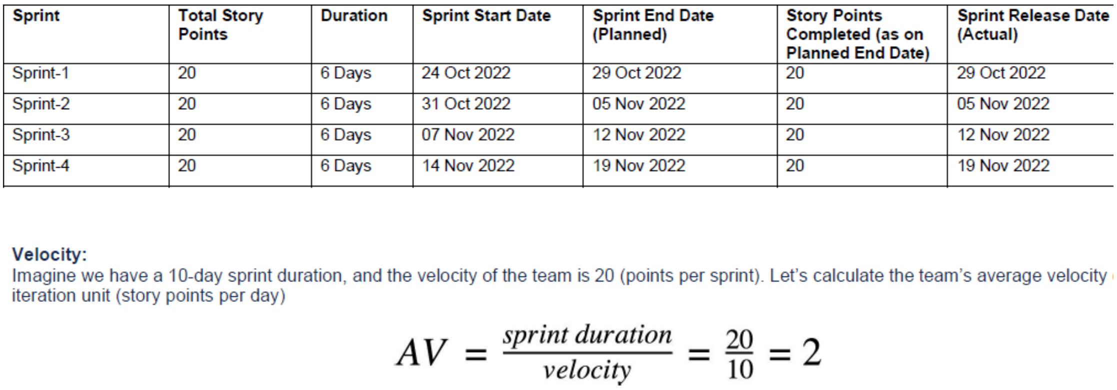
User Type	Functional Requirement (Epic)	User Story Number	User Story I Task	Acceptance criteria	Priority	Release
		USN-4	As a user, I can register for the application through Gmail	I can increase or decrease weather	Medium	Sprint-I
		USN-5	As a user, I can log into the application by entering email & password	I can access my weather status ahead in my field	High	Sprint-I
	Dashboard	USN-6	As a user, I can log into the open weather map by entering email & password	I can access the application through my Gmail login		Spint-2
Customer (Web user)	Interface	USN-7	As a user the interface should be simple and easily accessible	I can access the interface easily	Hligh	Spint-l

Customer Care Executive	Data generation	USN-8	As a user open weather application to access the data regarding the weather changes	I can access the data regarding the weather through the application		Spint-l
Administrator	Problem Solving/ Fault clearance	USN-9	As an official who is in charge for the proper fumnctioning of the sign boards have to maintain it through periodic monitoring.	Officials can monitor the sign boards for proper functioning	Medium	Spint-2

CHAPTER-6

PROJECT PLANNING AND SCHEDULING:

6.1 SPRINT PLANNING AND ESTIMATION

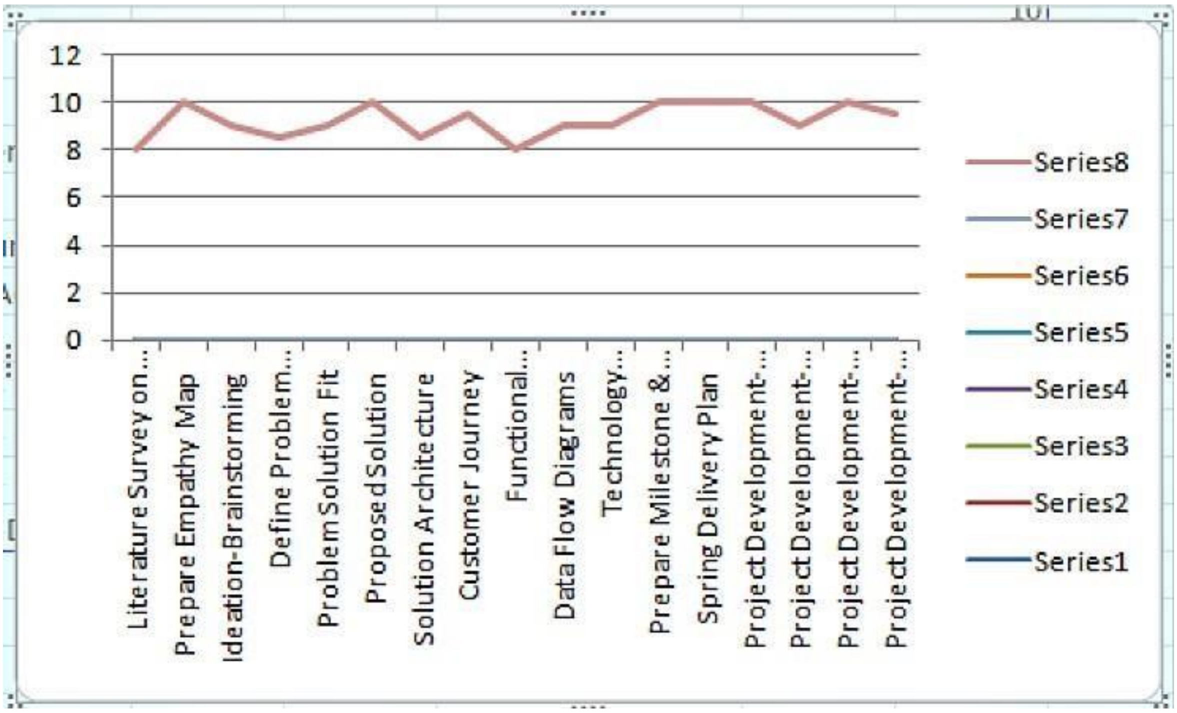


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	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)

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



6.2 SPRINT DELIVERY

Sprint	FunctionalRequirement(Epic)	UserStoryNumber	UserStory/ Task	Story Points	Priority	TeamMembers

Sprint-1		US-1	Create the IBMCloud serviceswhich are beingused in thisproject.	6	High	Manoj kumar S Ramesh S Niranjan N Yuvan sankar raja B
Sprint-1		US-2	Configure theIBM Cloudservices whicharebeingus edincompleting thisproject.	4	Medium	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B
Sprint-2		US-3	IBMWatsonIoTp latform acts asthe mediator toconnect the webapplication toIoTdevices,so createtheIBM	5	Medium	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B

Sprint	**FunctionalR equirement(Epic)**	**UserSto ryNumb er**	**UserStory/ Task**	**Story Points**	**Priority**	**TeamMembers**

Sprint-2		US-4	In order toconnect the IoTdevice to theIBM cloud,createadeviceinthe IBM WatsonIoT platform andget the devicecredentials.	5	High	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B
Sprint-3		US-1	Configure theconnections ecurity andcreateAPIk eysthat are used inthe Node-REDservice foraccessing theIBMIoT Platform.	10	High	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B
Sprint-3		US-2	CreateaNode-REDservice.	10	High	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B
Sprint-3		US-1	Developa	7	High	Niranjan N

Sprint	**FunctionalRequirement(Epic)**	**UserStoryNumber**	**UserStory/ Task**	**Story Points**	**Priority**	**TeamMembers**
Sprint -2		US-2	python script topublish randomsensordatasuchas temperature,moisture, soiland humidity tothe IBM IoTplatform			Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B
Sprint-3		US-2	After developingpython code,commands arereceived justprint thestatements whichrepresent thecontrol of thedevices.	5	Medium	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B
Sprint-4		US-3	Publish Data toTheIBMCloud	8	High	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B

Sprint-4		US-1	Create Web UIinNode-Red	10	High	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B

Sprint	FunctionalRequirement(Epic)	UserStoryNumber	UserStory/ Task	Story Points	Priority	TeamMembers
Sprint-4		US-2	Configure theNode-RED flowto receive datafrom the IBMIoT platform andalsouseCloudant DBnodestostorethe receivedsensordataintheCloudantDB	10	High	Manoj kumar s Ramesh s Niranjan N Yuvan sankar raja B

CHAPTER-7

CODING AND SOLUTION

7.1 FEATURE 1

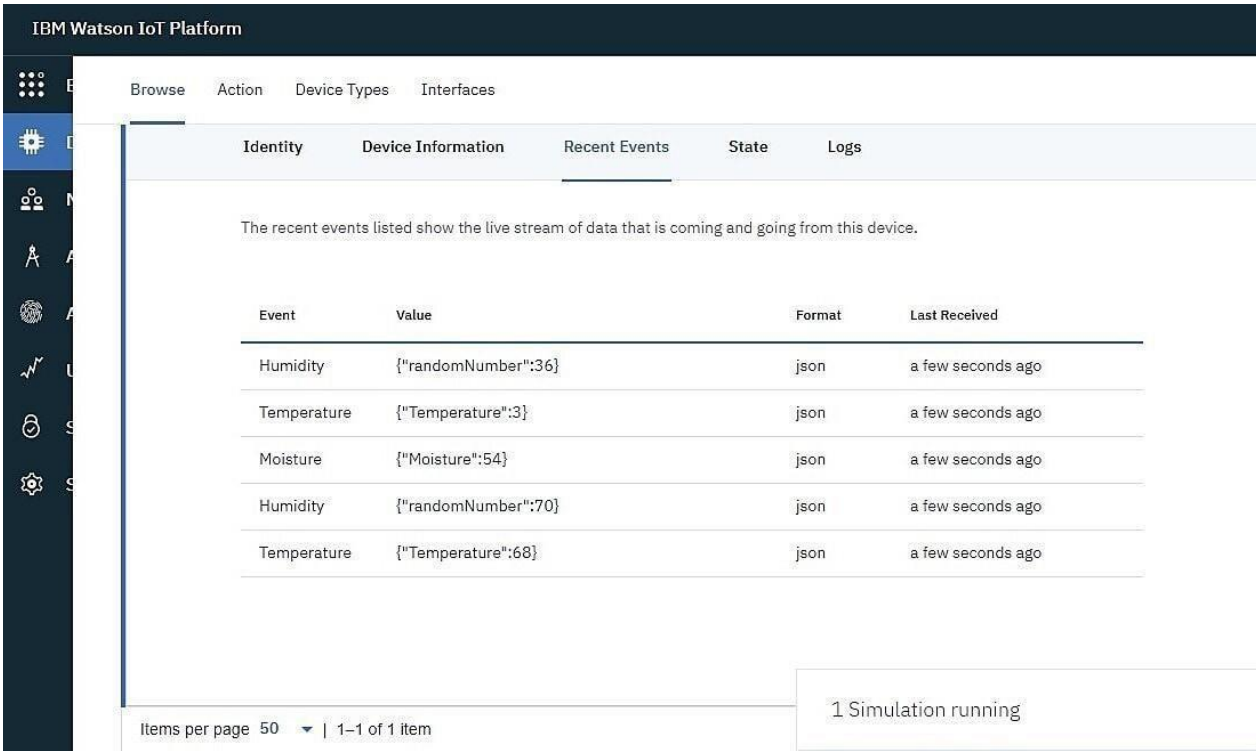
```
import time import
sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device
#Provide your IBM Watson Device Credentials organization
= "hrodmj" #replace the ORG ID deviceType =
"NODEMCU1"#replace the Device type wi deviceId =
"12345"#replace Device ID
authMethod = "token"
authToken = "kp1234" #Replace the authtoken
def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("Motor On IS RECEIVED")
    elif cmd.data['command']=='motoroff':
        print("Motor Off IS RECEIVED")
    if cmd.command == "setInterval":
        if 'interval' not in cmd.data:
print("Error - command is missing required information: 'interval'")else:
            interval = cmd.data['interval'] elif
    cmd.command == "print":
        if 'message' not in cmd.data:

            print("Error - command is missing required information: 'message'")
        else:
            output=cmd.data['message']
            print(output)
        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 as an event of type
"greeting" 10 times

deviceCli.connect()  while
True:
       deviceCli.commandCallback = myCommandCallback #
Disconnect the device and application from the cloud
deviceCli.disconnect()
```



The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Humidity	{"randomNumber":36}	json	a few seconds ago
Temperature	{"Temperature":3}	json	a few seconds ago
Moisture	{"Moisture":54}	json	a few seconds ago
Humidity	{"randomNumber":70}	json	a few seconds ago
Temperature	{"Temperature":68}	json	a few seconds ago

7.2 FEATURE 2

i. Good sensitivity to Combustible gas in wide range .

ii. High sensitivity to LPG, Propane and Hydrogen .

iii. Long life and low cost.

iv. Simple drive circuit.

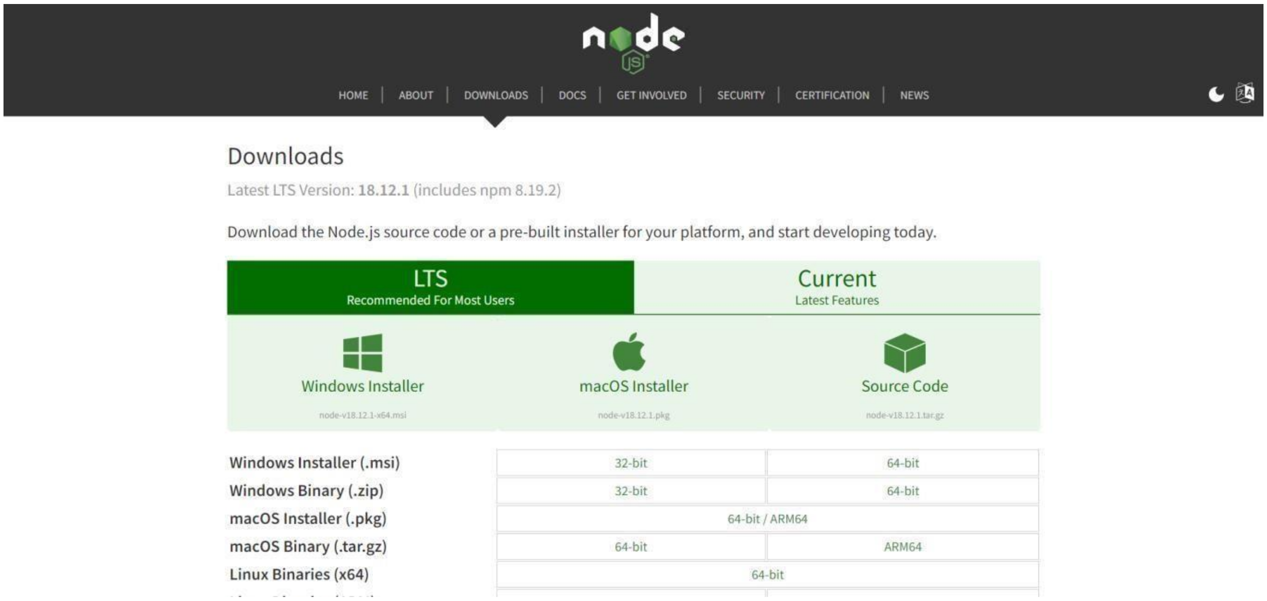
CHAPTER-8

TESTING

8.1 TEST CASES

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

8.2 USER ACCEPTANCE TESTING



Downloads

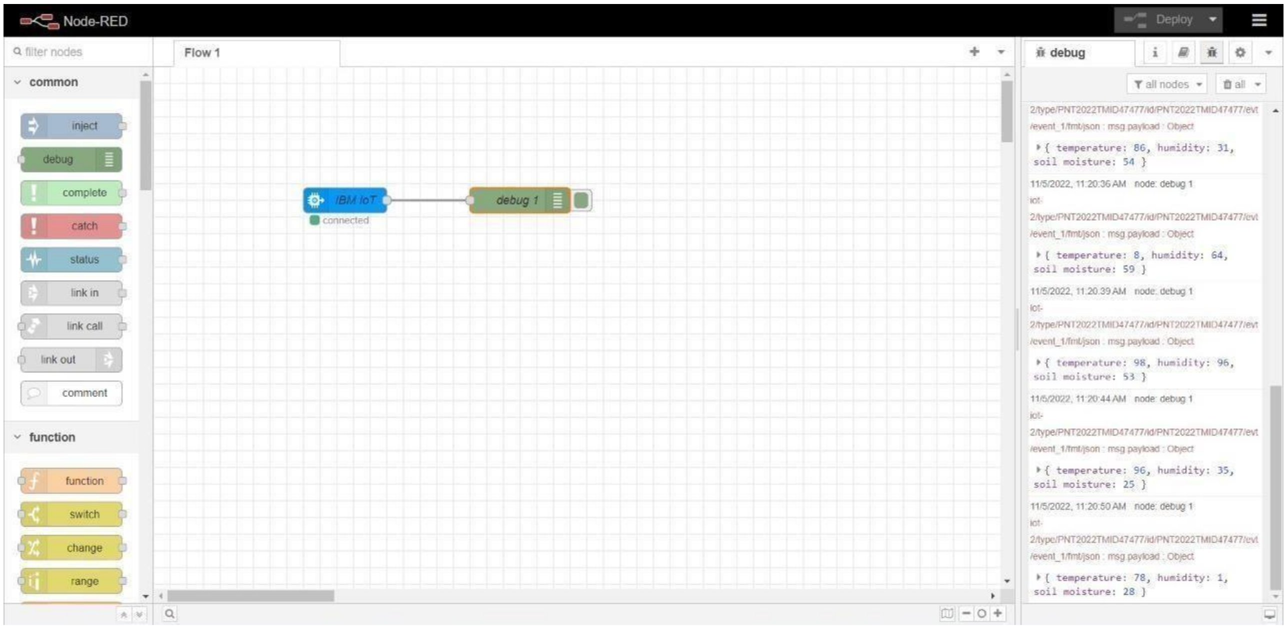
Latest LTS Version: **18.12.1** (includes npm 8.19.2)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

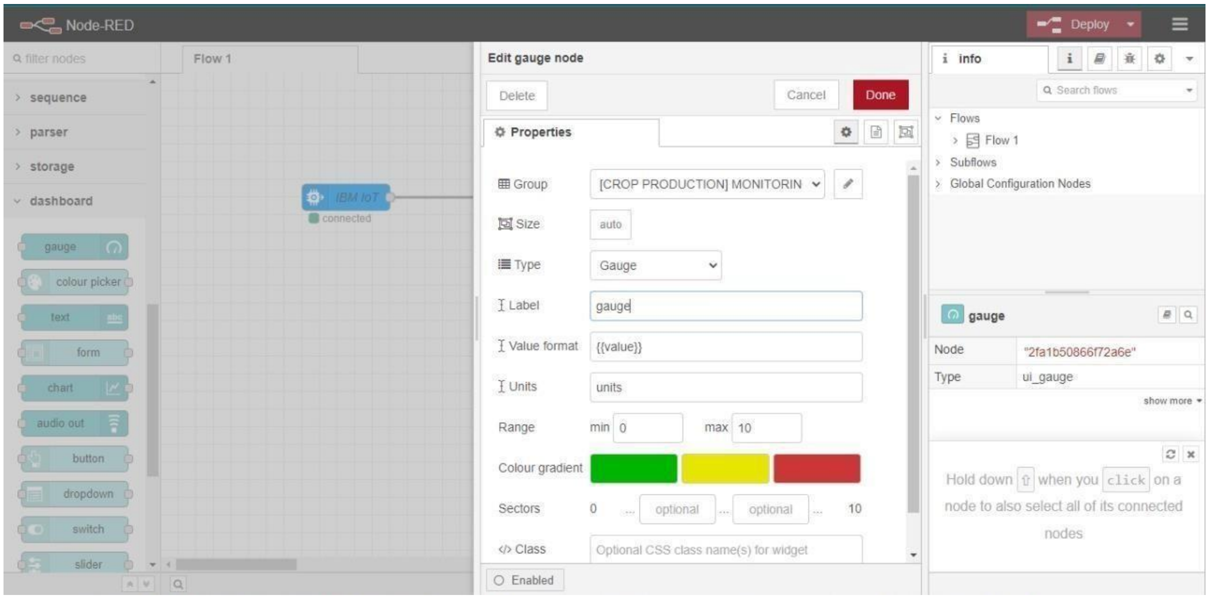
LTS Recommended For Most Users	Current Latest Features

Windows Installer	macOS Installer	Source Code
node-v18.12.1-x64.msi	node-v18.12.1.pkg	node-v18.12.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	



```
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
 ▸{ temperature: 86, humidity: 31,
 soil moisture: 54 }
11/5/2022, 11:20:36 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
 ▸{ temperature: 8, humidity: 64,
 soil moisture: 59 }
11/5/2022, 11:20:39 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
 ▸{ temperature: 98, humidity: 96,
 soil moisture: 53 }
11/5/2022, 11:20:44 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
 ▸{ temperature: 96, humidity: 35,
 soil moisture: 25 }
11/5/2022, 11:20:50 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
 ▸{ temperature: 78, humidity: 1,
 soil moisture: 28 }
```



Group [CROP PRODUCTION] MONITORIN

Size auto

Type Gauge

Label gauge

Value format {{value}}

Units units

Range min 0 max 10

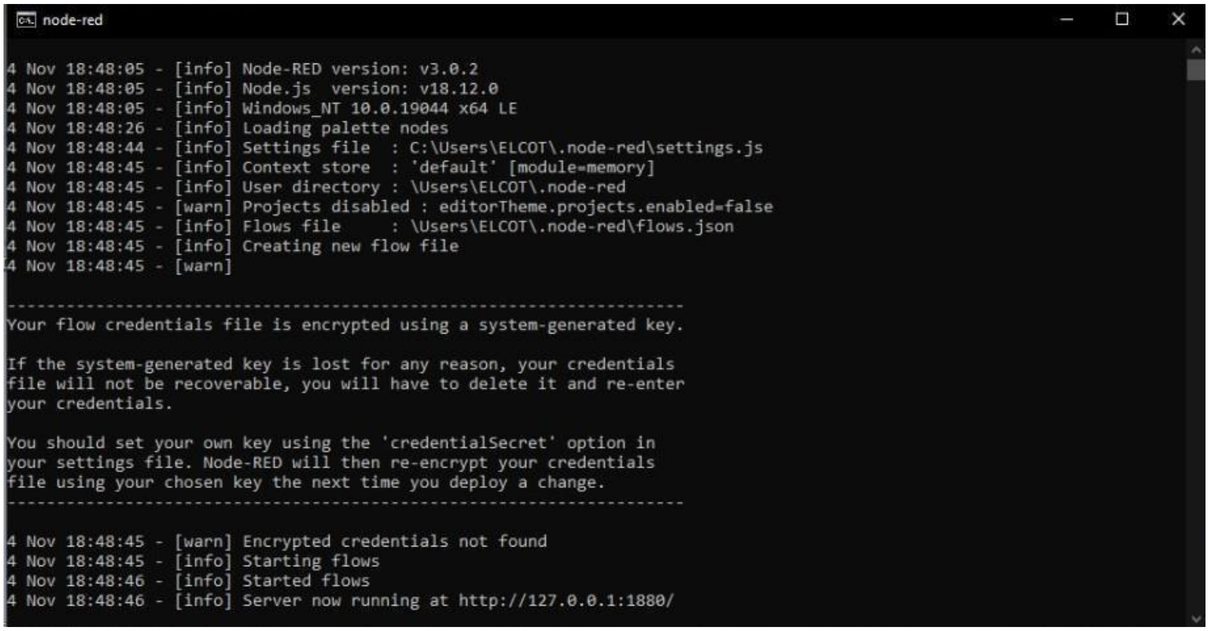
Colour gradient 🟩 🟨 🟥

Sectors 0 ... optional ... optional ... 10

Class Optional CSS class name(s) for widget

Node	"2fa1b50866f72a6e"
Type	ui_gauge

CHAPTER-09



```
4 Nov 18:48:05 - [info] Node-RED version: v3.0.2
4 Nov 18:48:05 - [info] Node.js  version: v18.12.0
4 Nov 18:48:05 - [info] Windows_NT 10.0.19044 x64 LE
4 Nov 18:48:26 - [info] Loading palette nodes
4 Nov 18:48:44 - [info] Settings file  : C:\Users\ELCOT\.node-red\settings.js
4 Nov 18:48:45 - [info] Context store  : 'default' [module=memory]
4 Nov 18:48:45 - [info] User directory : \Users\ELCOT\.node-red
4 Nov 18:48:45 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Nov 18:48:45 - [info] Flows file     : \Users\ELCOT\.node-red\flows.json
4 Nov 18:48:45 - [info] Creating new flow file
4 Nov 18:48:45 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

4 Nov 18:48:45 - [warn] Encrypted credentials not found
4 Nov 18:48:45 - [info] Starting flows
4 Nov 18:48:46 - [info] Started flows
4 Nov 18:48:46 - [info] Server now running at http://127.0.0.1:1880/
```

RESULT

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

It requires urgent attention as no effective solution existstill date for this problem. Thus this project carries a greatsocial relevance as it aims to address this problem. This project will help farmers in protecting their orchards and fields and save them from significant financial losses andwill save them from the unproductive efforts that they endure for the protection their fields. This will also help them in achieving better crop yields thus leading to their economic wellbeing.

CHAPTER-10

ADVANTAGES AND DISADVANTAGES

Advantage:

Controllable food supply. you might have droughts or floods, but ifyou are growing the crops and breeding them to be hardier, you have a better chance of 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 then environment of the planet.

CHAPTER-11

CONCLUSION

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

CHAPTER-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 detectedby cameras and if it comes towards form then 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 sensingthis laser or sensor’s security system will be activated.

CHAPTER-13
APPENDIX

SOURCE CODE

A. MOTOR.PY

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf import
ibmiotf.device

# Provide your IBM Watson Device Credentials
organization = "8gyz7t" # replace the ORG ID
deviceType = "weather_monitor" # replace the Device typedeviceId =
"b827ebd607b5" # replace Device ID authMethod = "token"
authToken = "LWVpQPaVQ166HWN48f" # Replace the authtoken

 def myCommandCallback(cmd): # function for Callback if
    cmd.data['command'] == 'motoron':
        print("MOTOR ON IS RECEIVED")

    elif cmd.data['command'] == 'motoroff':
        print("MOTOR OFF IS RECEIVED")
    if cmd.command == "setInterval": if
        'interval' not in cmd.data:
            print("Error - command is missing required information: 'interval'")
        else:
            interval = cmd.data['interval'] elif
    cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
        else:
            output    =    cmd.data['message']
            print(output)
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 as an event oftype "greeting" 10
times
deviceCli.connect()

while True:
    deviceCli.commandCallback = myCommandCallback

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

SENSOR.PY

```
import    time
import sys
import       ibmiotf.application
import  ibmiotf.device  import
random
#  Provide  your  IBM  Watson  Device  Credentials
organization = "8gyz7t"  # replace the ORG ID
deviceType = "weather_monitor" # replace the Device typedeviceId =
"b827ebd607b5" # replace Device ID authMethod = "token"
```

```
authToken = "LWVpQPaVQ166HWN48f"  # Replace the authtoken

def myCommandCallback(cmd):

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

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 as an event oftype "greeting" 10
times
deviceCli.connect()

while True:
       temp=random.randint(0,100)
       pulse=random.randint(0,100)
       soil=random.randint(0,100)
       data = { 'temp' : temp, 'pulse': pulse ,'soil':soil}
       #print data
       def myOnPublishCallback():
         print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %pulse,"Soil Moisture =
%s %%" % soil,"to IBM Watson")

      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()
```

A. Node-RED FLOW :

```
[
{ "id":"625574ead9839b34",
"type":"ibmiotout", "z":"630c8601c5ac3295",
"authentication":"apiKey",
"apiKey":"ef745d48e395ccc0",
"outputType":"cmd",
"deviceId":"b827ebd607b5",
"deviceType":"weather_monitor",
"eventCommandType":"data",
"format":"json",
"data":"data",      "qos":0,
"name":"IBM          IoT",
"service":"registered",
"x":680,
"y":220,
"wires":[]
},
{
"id":"4cff18c3274cccc4",  "type":"ui_button",
"z":"630c8601c5ac3295",
"name":"",          "group":"716e956.00eed6c",
"order":2,
"width":"0",
"height":"0",
"passthru":false,
"label":"MotorON",
"tooltip":"",
"color":"",
"bgcolor":"",
"className":"",
"icon":"",             "payload":"{\"command\":\"motoron\"}",
"payloadType":"str",
"topic":"motoron",
"topicType":"str",
"x":360,
"y":160, "wires":[["625574ead9839b34"]]},
{
"id":"659589baceb4e0b0",
"type":"ui_button",
"z":"630c8601c5ac3295",
"name":"",          "group":"716e956.00eed6c",
"order":3,
"width":"0",
"height":"0",     "passthru":true,
"label":"MotorOFF",
"tooltip":"",
"color":"",
"bgcolor":"",
"className":"",
"icon":"",             "payload":"{\"command\":\"motoroff\"}",
"payloadType":"str",
"topic":"motoroff",
"topicType":"str",
"x":350,
"y":220, "wires":[["625574ead9839b34"]]},
{"id":"ef745d48e395ccc0",
"type":"ibmiot",
"name":"weather_monitor",
"keepalive":"60",
"serverName":"",
"cleansession":true,
"appId":"",
"shared":false},
{"id":"716e956.00eed6c",
"type":"ui_group",
"name":"Form",
"tab":"7e62365e.b7e6b8",
"order":1,
"disp":true,
"width":"6",
"collapse":false},
{"id":"7e62365e.b7e6b8",
"type":"ui_tab",
"name":"contorl",
"icon":"dashboard",
"order":1,
"disabled":false,
"hidden":false}
]
[
{
"id":"b42b5519fee73ee2",
"type":"ibmiotin",
"z":"03acb6ae05a0c712",
"authentication":"apiKey",
"apiKey":"ef745d48e395ccc0",
"inputType":"evt",
"logicalInterface":"",          "ruleId":"",
"deviceId":"b827ebd607b5",
"applicationId":"",
"deviceType":"weather_monitor",
"eventType":"+",
"commandType":"",
"format":"json",
"name":"IBMIoT",
"service":"registered",
"allDevices":"",
"allApplications":"",
"allDeviceTypes":"",
"allLogicalInterfaces":"",
"allEvents":true,
"allCommands":"",
"allFormats":"",
"qos":0,
"x":270,
"y":180,
"wires":[["50b13e02170d73fc","d7da6c2f5302ffaf","a949797028158f3f","a71f164bc3 78bcf1"]]
},
{ "id":"50b13e02170d73fc",
"type":"function",   "z":"03acb6ae05a0c712",
"name":"Soil Moisture",
"func":"msg.payload = msg.payload.soil;\nglobal.set('s',msg.payload);\nreturn msg;","outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":490,
"y":120,
"wires":[["a949797028158f3f","ba98e701f55f04fe"]]
},
{
"id":"d7da6c2f5302ffaf",    "type":"function",
"z":"03acb6ae05a0c712", "name":"Humidity",
"func":"msg.payload = msg.payload.pulse;\nglobal.set('p',msg.payload)\nreturn msg;","outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":480,
"y":260, "wires":[["a949797028158f3f","70a5b076eeb80b70"]]
```

```
},
{ "id":"a949797028158f3f",
"type":"debug",
"z":"03acb6ae05a0c712",
"name":"IBMo/p",
"active":true,   "tosidebar":true,
"console":false,
"tostatus":false,
"complete":"payload",
"targetType":"msg",
"statusVal":"",
"statusType":"auto", "x":780,
"y":180,
"wires":[]
},
{
"id":"70a5b076eeb80b70",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",   "name":"",
"group":"f4cb8513b95c98a4",
"order":6,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Humidity",
"label":"Percentage(%)",
"format":"{{value}}",
"min":0, "max":"100",
"colors":["#00b500","#e6e600","#ca3838"],
"seg1":"",
"seg2":"",
"className":"",
"x":860,
"y":260,
"wires":[]
},
{
"id":"a71f164bc378bcf1",
"type":"function",
"z":"03acb6ae05a0c712",
"name":"Temperature",
"func":"msg.payload=msg.payload.temp;\nglobal.set('t',msg.payload);\nreturn msg;","outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":490,
"y":360,
"wires":[["8e8b63b110c5ec2d","a949797028158f3f"]]
},
{
"id":"8e8b63b110c5ec2d",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",    "name":"",
"group":"f4cb8513b95c98a4",
"order":11,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Temperature",
"label":"DegreeCelcius",
"format":"{{value}}",
"min":0, "max":"100",
"colors":["#00b500","#e6e600","#ca3838"], "seg1":"",
"seg2":"",
"className":"",
"x":790,
"y":360,
"wires":[]
},
{
"id":"ba98e701f55f04fe",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",    "name":"",
"group":"f4cb8513b95c98a4",
"order":1,
"width":"0",
"height":"0",
"gtype":"gage",    "title":"Soil
Moisture",
"label":"Percentage(%)",
"format":"{{value}}",
"min":0, "max":"100",
"colors":["#00b500","#e6e600","#ca3838"], "seg1":"",
"seg2":"",
"className":"",
"x":790,
"y":120,
"wires":[]
},
{
"id":"a259673baf5f0f98",
"type":"httpin",
"z":"03acb6ae05a0c712",
"name":"",
"url":"/sensor",
"method":"get",
"upload":false,
"swaggerDoc":"",
"x":370,
"y":500,
"wires":[["18a8cdbf7943d27a"]]
},
{
"id":"18a8cdbf7943d27a",     "type":"function",
"z":"03acb6ae05a0c712",
"name":"httpfunction",
"func":"msg.payload{\"pulse\":global.get('p'),\"temp\":global.get('t'),\"soil\":global.get( 's')};\nreturn msg;",
"outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":630,
"y":500, "wires":[["5c7996d53a445412"]]
},
{ "id":"5c7996d53a445412",
"type":"httpresponse",
"z":"03acb6ae05a0c712",
"name":"",
"statusCode":"",
"headers":{
},"x":870,
"y":500,
"wires":[]
},
{
"id":"ef745d48e395ccc0",
"type":"ibmiot",
"name":"weather_monitor",
"keepalive":"60",
"serverName":"",
"cleansession":true
,"appId":"",
"shared":false},
{
"id":"f4cb8513b95c98a4",
"type":"ui_group",
"name":"monitor",
"tab":"1f4cb829.2fdee8",
"order":2,
"disp":true,
"width":"6",
"collapse":false
,
"className":"
"
},
{
"id":"1f4cb829.2fdee8",
"type":"ui_tab",

"name":"Home",

"icon":"dashboard"
,             "order":3,
"disabled":false,
"hidden":false }
```

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-7336-1658852955

DEMO VEDIO LINK

https://github.com/IBM-EPBL/IBM-Project-7336-1658852955/blob/main/Final%20deliverables/VID-20221118-WA0013.mp4