PROJECT REPORT

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

PROJECT TITLE	IoT Based Smart Crop Protection System for Agriculture
TEAM ID	PNT2022TMID21685
TEAM MEMBERS	Rahul Raj Kanuparthi Saranya D.Manoranjani Kasireddy Prakash D.N.Karthikeyan
BRANCH	Computer Science and Engineering

1. INTRODUCTION:

PROJECT OVERVIEW:

The problem of crop vandalization by wild animals has become a major social problem. Crops in farms are frequently devastated by local animals such as buffaloes, cows, goats, birds, and so on. Various types of species such as birds and animals come to the cultivation field according to the crop that is being cultivated and also according to the season of cultivation. Some wild animals enter the field during night times when the field is near a forest region or when the farm cultivates some fruits and other crops that attract animals. This results massive losses for the farmer.

However, according to nature's rules, every living creature on this planet plays an important role in the eco-system. Elephants and other animals that come into touch with humans have a negative impact in a variety of ways, including agricultural depredation, damage to grain stores, water supply, dwellings and other assets, and human injury and death.

PURPOSE:

The main objective of this project was to design a small scale irrigated system that would use water in more well-organized way in order to prevent excess water loss and minimize the cost of labor. The following aspects were considered in the choice of design solution - Installation cost , Water saving , Human intervention , Reliability , Power consumption ,Maintenance, Expandability A critical Consideration in the segment costs, since cost define the viability and feasibility of a project.

Before Electric fences were used to control livestock in the United States in the early 1930s, and electric fencing technology developed in both the United States and New Zealand. An early application of the electric fence for livestock control was developed in 1936–1937 by New Zealand inventor Bill Gallagher. One of the major disadvantagesof having an electric fence installed is that it requires regular maintenance.
Scarecrow:

Scarecrow genealogy is rooted in a rural life style. The Egyptians used the first scarecrows in recorded history to use to protect wheat fields along the Nile River fromflocks of quail. Egyptian farmers installed wooden frames in their fields and covered them with nets. While traditional, motionless scarecrows do work against “pest birds” (e.g., crows and blackbirds), the effect is almost always temporary. Over time, the birdsget used to stationary dummies and resume their destructive habits.

2. LITERATURE SURVEY:

AUTHOR NAME	YEAR	TOPIC
Artur Frankiewicz; Rafal Cupek	2013 IECON 2013-39th annual conference of the IEEE industrial electronics society pages:7543-7547	Smart passive infrared sensor-Hardware platform
Discant, A. Rogozan, C. Rusuand A. Bensrhair	2007-30th International SpringSeminar on Electronics Technology (ISSE), Cluj- Napoca, 2007, pp. 100-105. doi: 10.1109/ISSE.2007.4432828 Volume:01 Pages:859-862, DOI:10.1109/ICCSNT.2015.7490876, IEEE Conference Publications.	Sensors for Obstacle Detection
Mustapha, Baharuddin, AladinZayegh, and Rezaul K. Begg.	Artificial Intelligence, Modelling and Simulation (AIMS), 2013 1st International Conference on. IEEE, 2013	Ultrasonic And Infrared Sensors Performance in A Wireless Obstacle Detection System
Padmashree S. Dhake, Sumedha S. Borde	International Journal of Advanced Technology in Engineering and Science, www.ijates.com Volume No.02, Issue No. 03, March 2014.	Embedded Surveillance System Using PIR Sensor
DR. R. Bulli Babu, CH. JonathanSoumith, T. Cherishma Sri Lakshmi & R. Keshav Rao Kluniversity	India Global Journal of Computer Science and Technology: A Hardware & Computation Volume 15 Issue 2 Version 1.0 Year 2015 Type	GSM based Agriculture Monitoring and Controlling System
Dugyala Karthik, R.Ramesh Babu	International Journal of Advanced Information Science and Technology (IJAIST), ISSN:	Smart Crop Protection System with Image Captureover IOT

	2319:2682 Vol.6, No.11, November2017	

EXISTING PROBLEM:

Farmers use a variety of conventional techniques, such as scarecrows, electric fences, etc. In some places, farmers burn elephant dung or other items that produce thick smoke to keep their farmland from being destroyed. However, theyare not very good at keeping animals away from farms. Consequently, we created this affordable system to surveillance and to protect the farm effectively.

1. IOT Based Smart Agriculture System:

In today materialistic society, smart agriculture systems are a hot topic. This essay explains the idea of showcasing and maintaining an online agribusiness platform. The most crucial aspect of human life is agriculture, which may be improved utilizing IoT technology. IoT technology makes it possible to increase the effectiveness of agricultural automation systems. Smart agriculture system that makes use of the benefits of cutting-edge technology like Wireless Sensor Network and Arduino. The construction of a system that can track temperature, humidity, moisture, and even the movement of animals that can destroy crops in agricultural fields using sensors and an Arduino board is a feature of this study. The device has the potential to be helpful in water-scarce, remote places thanks to its low cost and energy independence.

ADVANTAGES:

- Efficiency, Expansion, Reduced resources.
- Clean process, Agility, Improved product quality**.**

2. Smart Crop protection system from living objects and fireusingArduino:

Farmers can no longer block entire fields or prepare a field for 24 hours of protection. Therefore, we are presenting this computerised crop safety system against fire and animals. This is a microcontroller- based device that is mostly based on the Arduino Uno. This method uses a motion sensor to find animals approaching the sphere and a smoke sensor to find the hearth. The sensor informs the microcontroller to take action in such a situation. The microcontroller now sounds an alert to further entice the animals away from the area while also calling the farmer and sending an SMS so that he can understandthe situation and visit the scene in case the animals don't go despite the noise. If smoke is detected, it quickly turns the motor ON.

3. Development of IOT based Smart Security and Monitoring Devices for Agriculture:

Since agriculture is the foundation of the Indian economy, it demands protection. Agriculture products need protection and safety at a very early stage, such as protection from rodent or insect attacks in fields or grain storage, and security is no longer just a matter of sources. Even so, these difficulties should be taken into account. Today's security systems don't seem to be intelligent enough to send out real-time notifications when they detect a problem. Agriculture can become more modernised by combining traditional methods with current technologies like wireless sensor networks and the internet of things. With this situation in mind, we developed, tested, and examined a "Internet of Things"-based device that can analyse the sensed data before transferring it to the user. This study aims to improve ways for resolving issues such rodent identification, crop hazards, and turning in real-time notifications backed records evaluation and processing in addition to human intervention. The sensors and digital units used in this gadget are integrated using Python program. With support from attempted test cases, we were successfulin 84.8% of test cases.

REFERENCE:

1. Damini Kalra, Praveen Kumar, K. Singh, Apurva Soni "Sensor Based Crop Protection System with IoT monitored Automatic Irrigation" 2nd International conference on Advances in Computing,Communication Control and Networking, 2020.

2. Padmashree S. Dhake, Sumedha S. Borde, "Embedded Surveillance System UsingPIR Sensor", International Journal of Advanced Technology in Engineering and Science, www.ijates.com Volume No.02, Issue No. 03, March 2014

3. Wang, Z., Wang, H., Liu, L., Song, W., & Lu, J. (2015, December). Community alarm system design based on MCU and GSM. In 2015 4th International Conference on Computer Science and Network Technology (ICCSNT) (Vol. 1, pp. 859-862). IEEE.

4. Shende, P. Y., Raut, S. M., Ingale, P. S., Nagose, A. K., Katakpur, P. S., & Kathane,

S. S. Solar Electric Fencing for Irrigation of Animal Man Conflict

5.Mohammad, T. (2009). Using ultrasonic and infrared sensors for distance measurement. World Academy of Science, Engineering and Technology, 51, 293-299

6.Volume:01 Pages:859-862, DOI:10.1109/ICCSNT.2015.7490876, IEEE

Conference Publications.

7.T. Day and R. Mac Gibbon, "Multiple-Species Exclusion Fencing and Technology for Mainland Sites.",

Project Report published by National Wildlife Research Centre, 2007.

8. R. Padula and W. Head, "Fencing System" Project Report published by University of Minnesota, 2003.

PROBLEM STATEMENT DEFINITION:

Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds etc. This leads to huge losses for the farmer. Due to over population, it occurs a deforestation this results in shortage of food, water and shelter in forest areas.

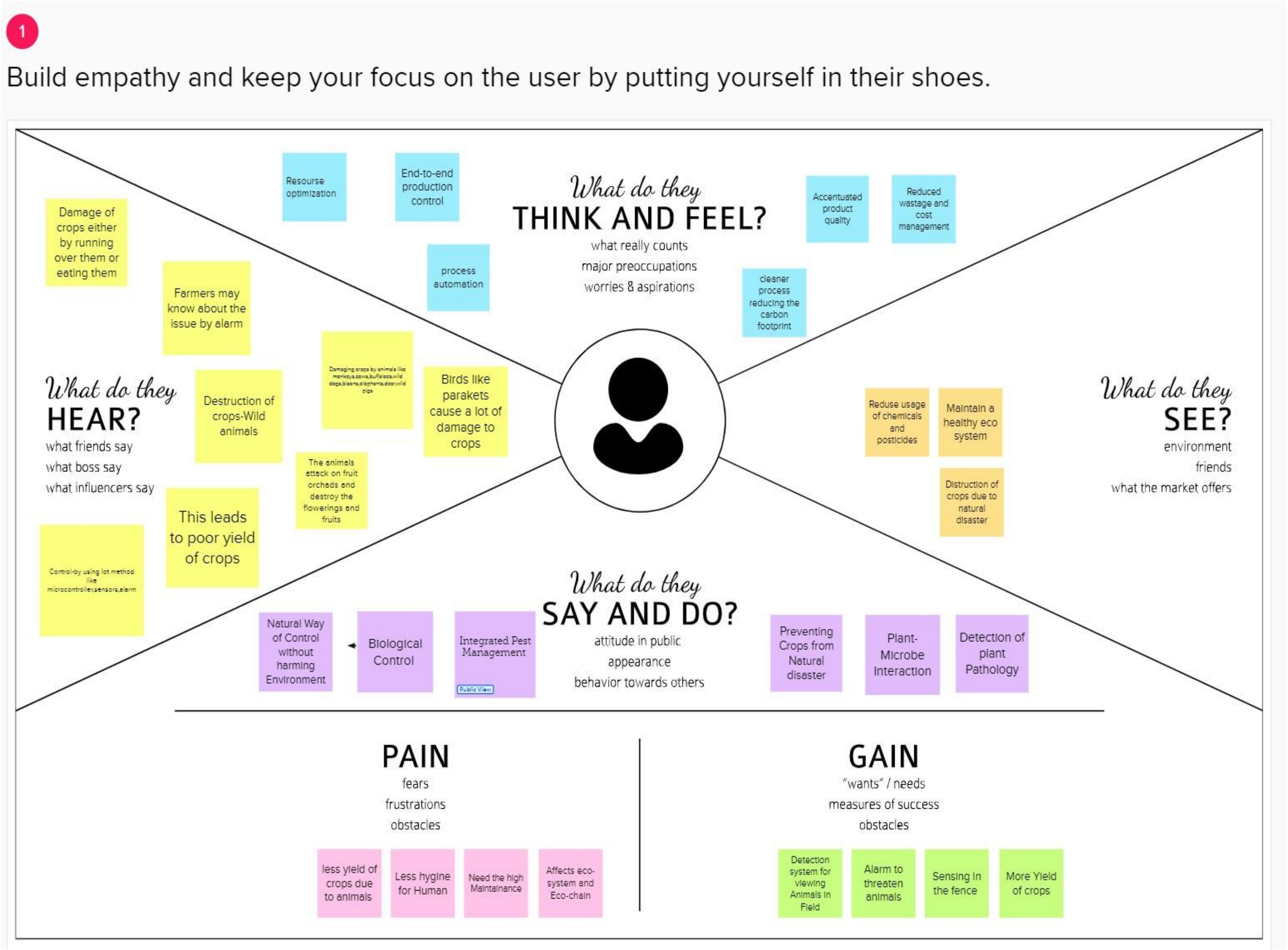
So, to avoid this we proposed Smart crop protection system from animals. This is a system which uses a motion sensor to detect wild animals approaching near the field. Here the sensor signals the microcontroller to take action. The microcontroller triggers an alarm in the field by detecting the presence of animals so that farmer may know about the issue and come to the spot.

Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**Because**	**Which makesmefeel**
PS-1 Develop a system for predicting cotton crop production and probable pest, disease and insect attacks (before atleas t 15 days and more). (Technology Bucket: Satellite images, Cloud computing. BigData ,In-field sensor data, drone imagery, and other Iot data.	Cost effective, small size	To create an app- based forecastin g system that canforecast potential pest, disease,and insect attacks on cotton crops.	Involvesmore requirements	Instead of using general recommendations, forecast the cotton crop yield production for farmers in the Vidarbha region using farm historical data, local terrain, weather scenarios, and numerous sensor inputs

PS-2 Wild animals like elephant , wild pigs, deer, wild dogs ,bison may enter into the field which in turn destroy the field which in turn destroy the crop and reduce the farming.	Eco-friendly,customer satisfaction	1.Identify and evaluate risk possessed by wild and domestic animals. 2. Monitor and document animal activity on the farm. 3.Conduct field assessment before harvest	Coverage area is larger	Crop protection needs particularly cautious approach

3. IDEATION AND PREPOSED SOLUTION

EMPATHY MAP CANVAS:



1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

- Resourse optimization
- End-to-end production control
- process automation
- Accentuated product quality
- Reduced wastage and cost management
- cleaner process reducing the carbon footprint

What do they HEAR?
what friends say
what boss say
what influencers say

- Damage of crops either by running over them or eating them
- Farmers may know about the issue by alarm
- Destruction of crops-Wild animals
- Damaging crops by animals like monkeys,cows,buffaloes,wild dogs,bisons,elephants,deer,wild pigs
- Birds like parakets cause a lot of damage to crops
- The animals attack on fruit orchads and destroy the flowerings and fruits
- This leads to poor yield of crops
- Control-by using Iot method like microcontroller,sensors,alarm

What do they SEE?
environment
friends
what the market offers

- Reduse usage of chemicals and posticides
- Maintain a healthy eco system
- Distruction of crops due to natural disaster

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

- Natural Way of Control without harming Environment
- Biological Control
- Integrated Pest Management
- Preventing Crops from Natural disaster
- Plant-Microbe Interaction
- Detection of plant Pathology

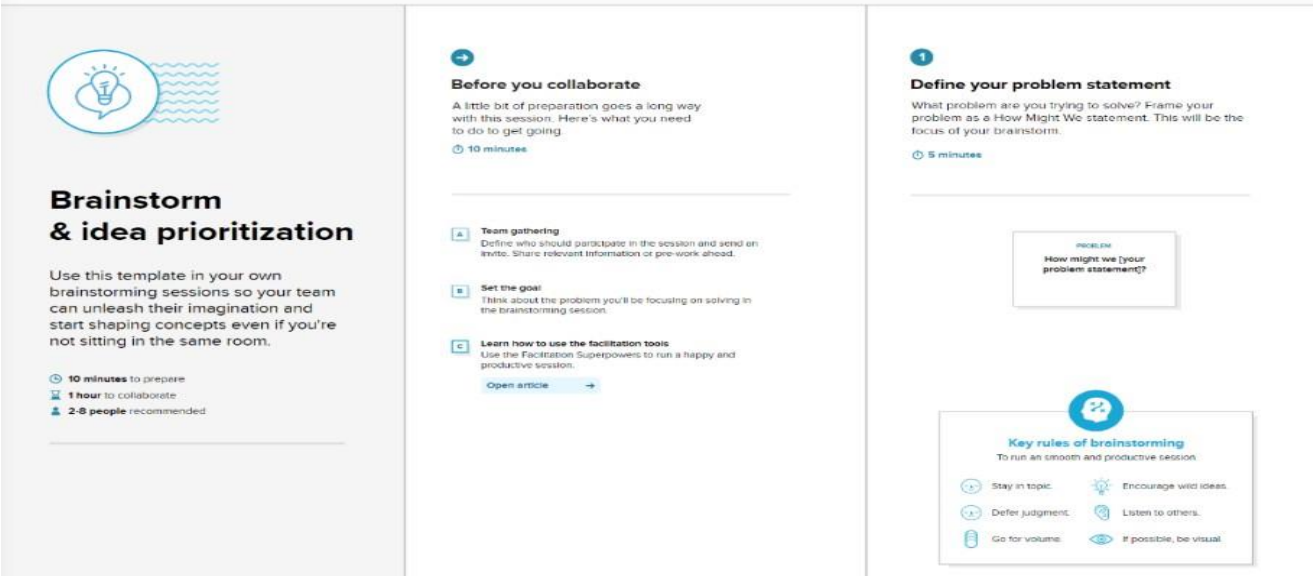
PAIN
fears
frustrations
obstacles

- less yield of crops due to animals
- Less hygine for Human
- Need the high Maintainance
- Affects eco-system and Eco-chain

GAIN
"wants" / needs
measures of success
obstacles

- Detection system for viewing Animals in Field
- Alarm to threaten animals
- Sensing in the fence
- More Yield of crops

IDEATION AND BRAINSTROMING:



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

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.

1 Define your problem statement

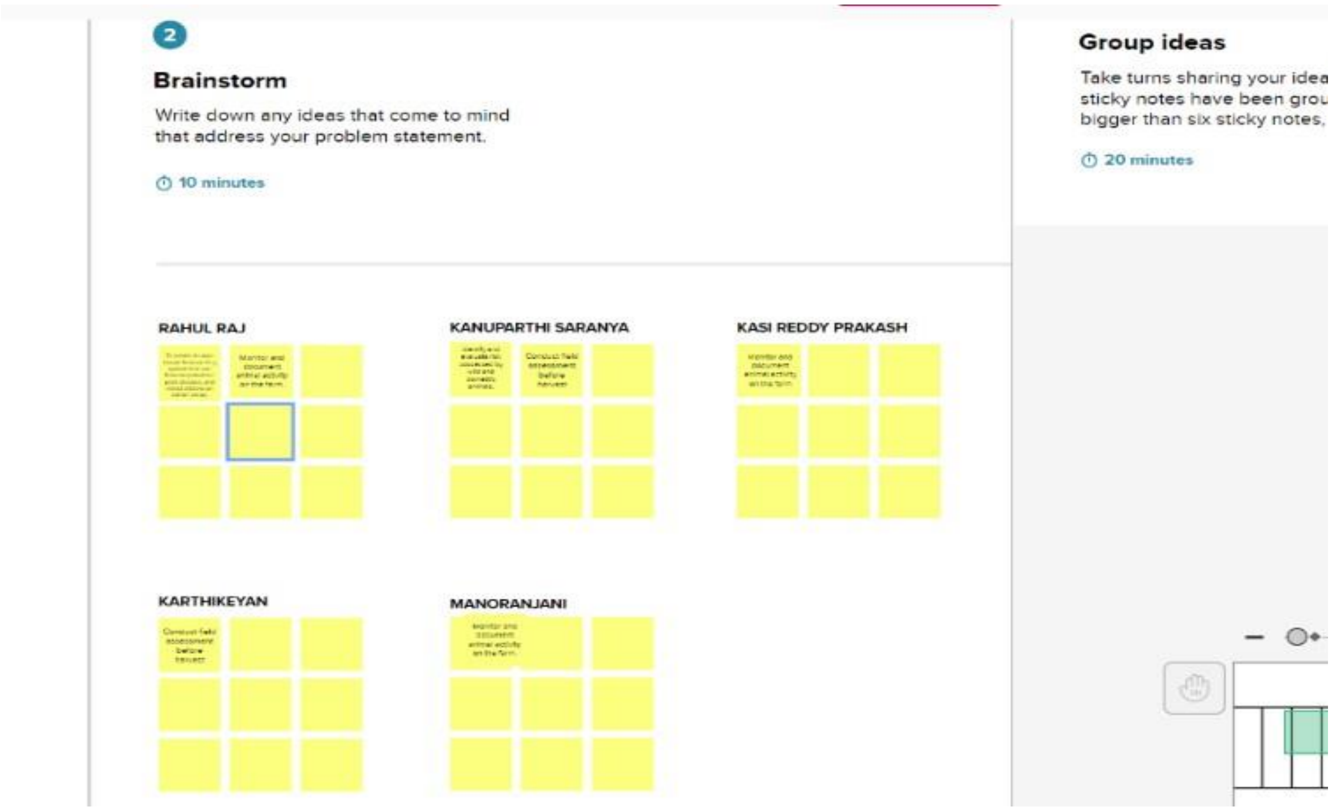
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

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

Group ideas

Take turns sharing your idea
sticky notes have been grou
bigger than six sticky notes,

20 minutes

RAHUL RAJ

[illegible]

Monitor and document animal activity on the farm.

KANUPARTHI SARANYA

Identify and evaluate risk associated by wild and domestic animals.

Conduct field assessment before harvest

KASI REDDY PRAKASH

Monitor and document animal activity on the farm.

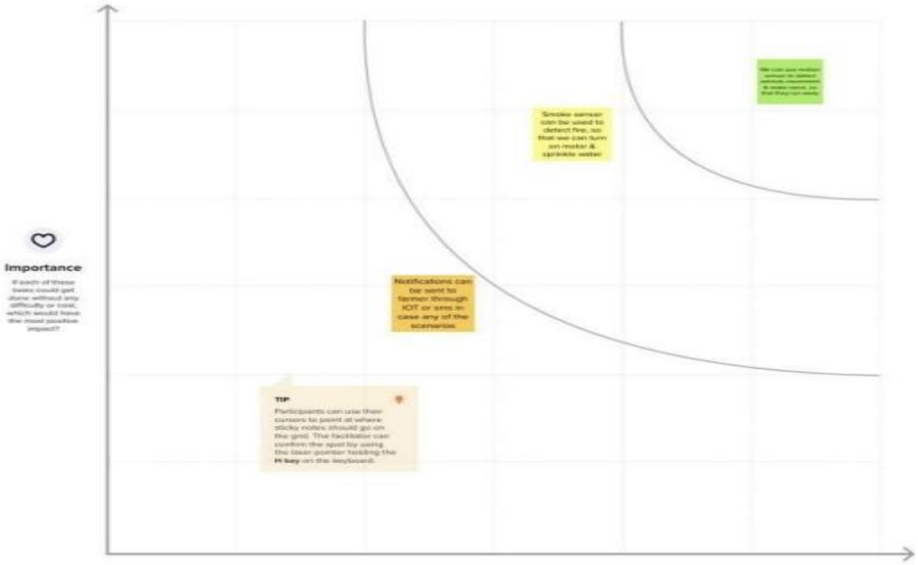
KARTHIKEYAN

Conduct field assessment before harvest

MANORANJANI

Monitor and document animal activity on the farm.

PRIORITIZATION:



Importance

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

We can use motion sensor to detect animals movement & make noise, so that they run away

Smoke sensor can be used to detect fire, so that we can turn on motor & sprinkle water

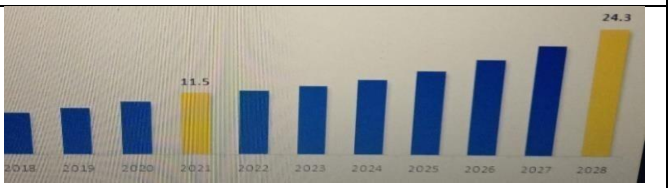
Notifications can be sent to farmer through IOT or sms in case any of the scenarios

TIP

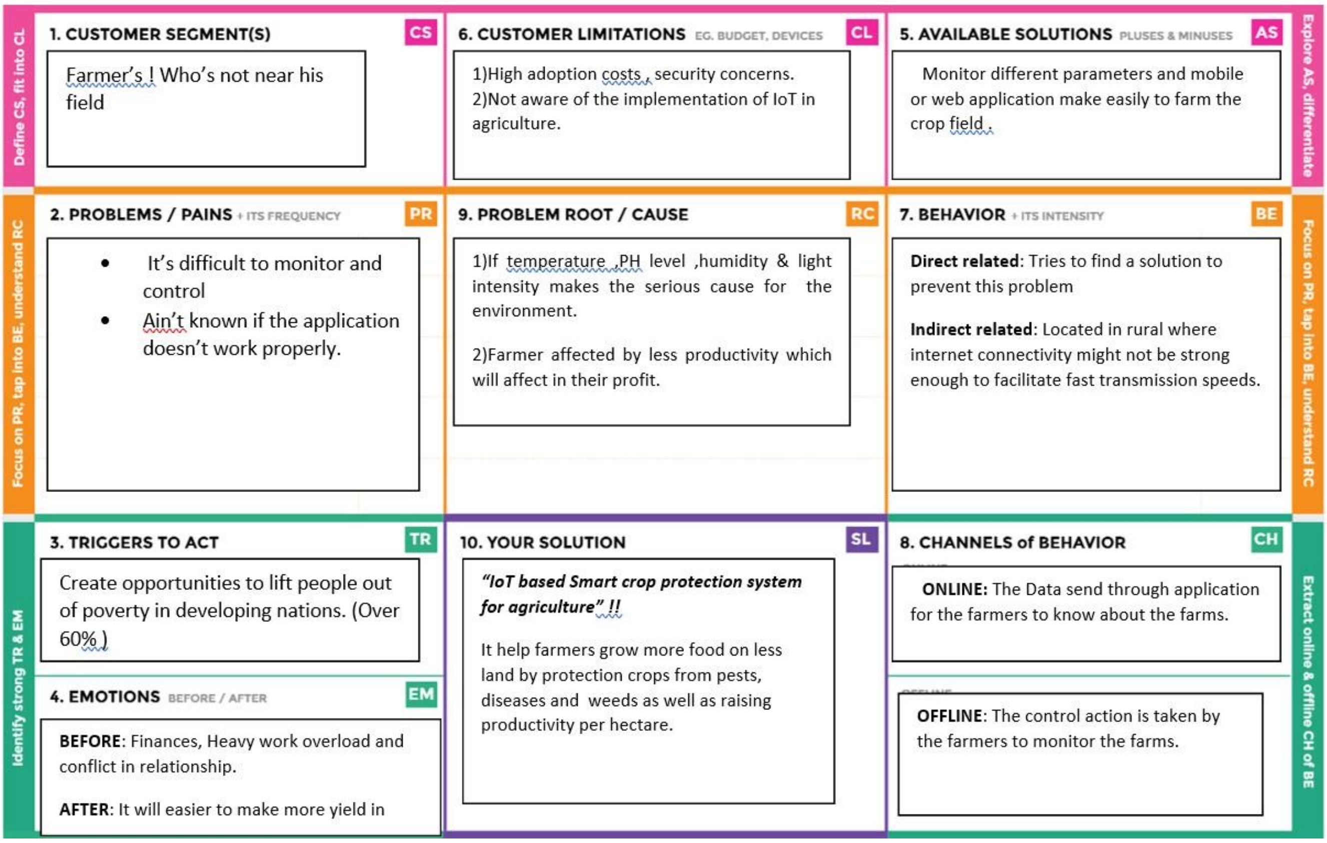
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Proposed Solution:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Usually crops in the fields are protected against birds and other unknown disturbances by humans. This take an enormous amount of time. Creating a smart automatic system will benefit the farmers in many different ways.
2.	Idea / Solution description	Smart Farming has enabled farmers to reduce waste and enhance productivity with the help of sensors (light, humidity, temperature, soil moisture, etc..) . Further with the help of these sensors, farmers can monitor the field conditions from anywhere.
3.	Novelty / Uniqueness	Role of SENSORS: IOT smart agriculture products are designed to help monitor crop fields using sensors and by automating irrigation systems. As a result, farmers and associated brands can easily monitor the field conditions from anywhere without any hassle.
4.	Social Impact / Customer Satisfaction	Water conservation. Saves lot of time. Increased quality of production. Real time data and production insight. Remote monitoring.
5.	Business Model (Revenue Model)	
6.	Scalability of the Solution	Scalability in smart farming refers to the adaptability of a system to increase the capacity, the number of technology devices such as sensors and fluctuators .

PROBLEM SOLUTION FIT:



Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Farmer's ! Who's not near his field

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

1)High adoption costs , security concerns.
2)Not aware of the implementation of IoT in agriculture.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

Monitor different parameters and mobile or web application make easily to farm the crop field .

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- It's difficult to monitor and control
- Ain't known if the application doesn't work properly.

9. PROBLEM ROOT / CAUSE RC

1)If temperature ,PH level ,humidity & light intensity makes the serious cause for the environment.

2)Farmer affected by less productivity which will affect in their profit.

7. BEHAVIOR + ITS INTENSITY BE

Direct related: Tries to find a solution to prevent this problem

Indirect related: Located in rural where internet connectivity might not be strong enough to facilitate fast transmission speeds.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

Create opportunities to lift people out of poverty in developing nations. (Over 60%)

4. EMOTIONS BEFORE / AFTER EM

BEFORE: Finances, Heavy work overload and conflict in relationship.

AFTER: It will easier to make more yield in

10. YOUR SOLUTION SL

"IoT based Smart crop protection system for agriculture" !!

It help farmers grow more food on less land by protection crops from pests, diseases and weeds as well as raising productivity per hectare.

8. CHANNELS of BEHAVIOR CH

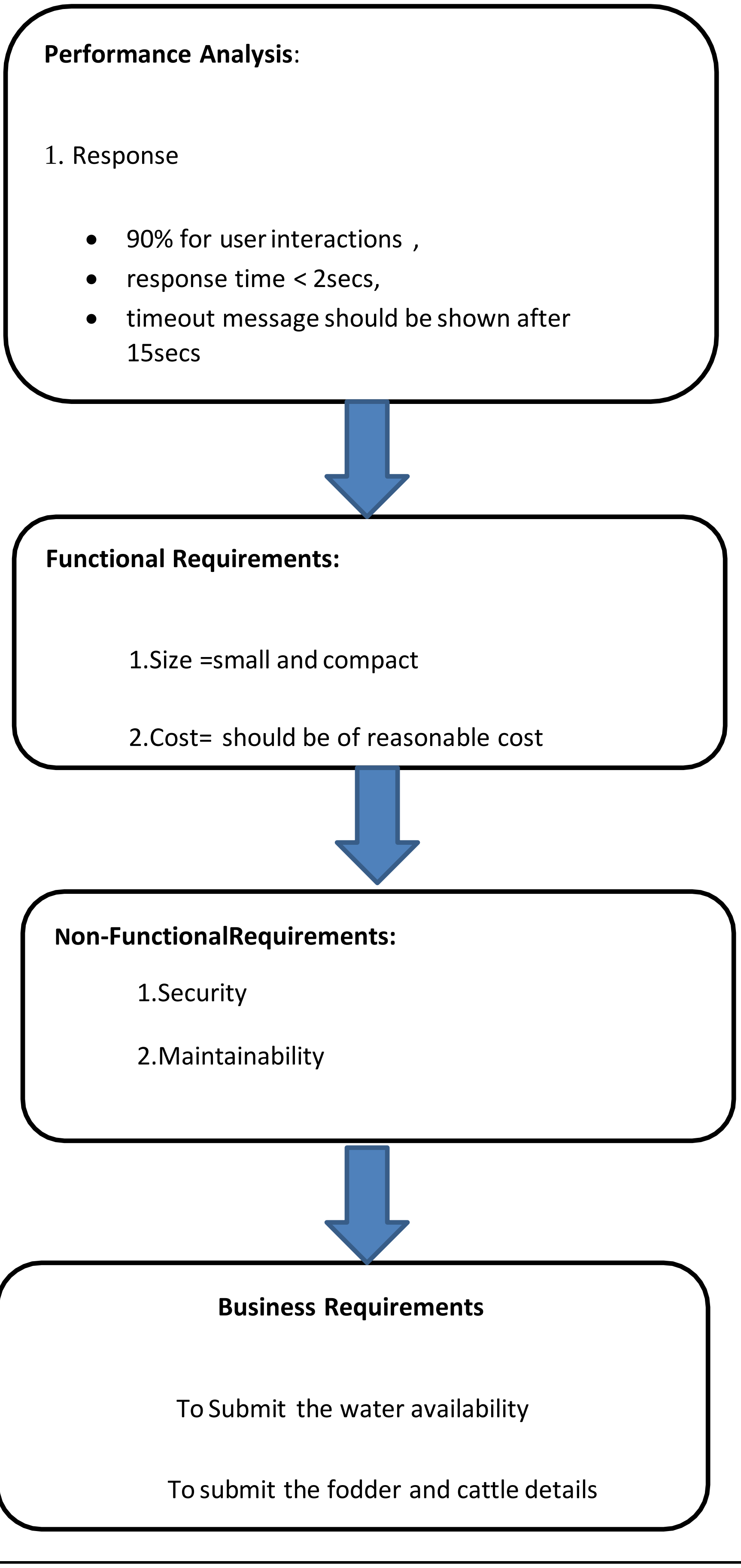
ONLINE: The Data send through application for the farmers to know about the farms.

OFFLINE: The control action is taken by the farmers to monitor the farms.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS:



Performance Analysis:

1. Response

- 90% for user interactions ,
- response time < 2secs,
- timeout message should be shown after 15secs

Functional Requirements:

1.Size =small and compact

2.Cost= should be of reasonable cost

Non-FunctionalRequirements:

1.Security

2.Maintainability

Business Requirements

To Submit the water availability

To submit the fodder and cattle details

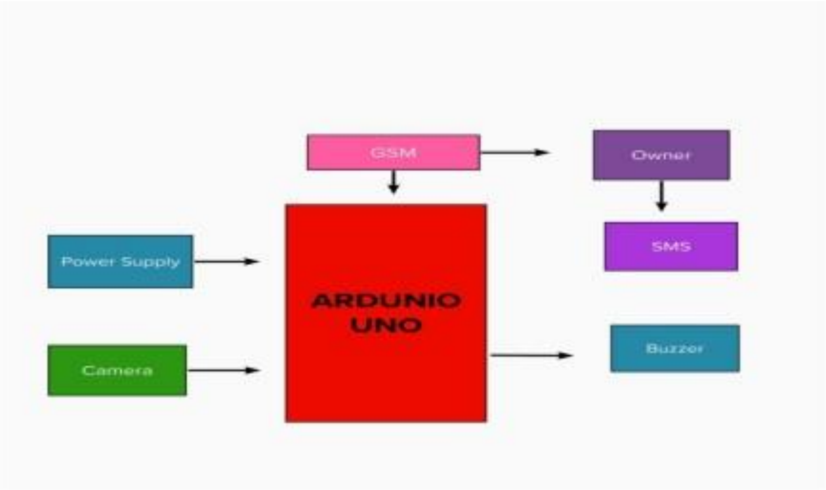
NON-FUNCTIONAL REQUIREMENTS:

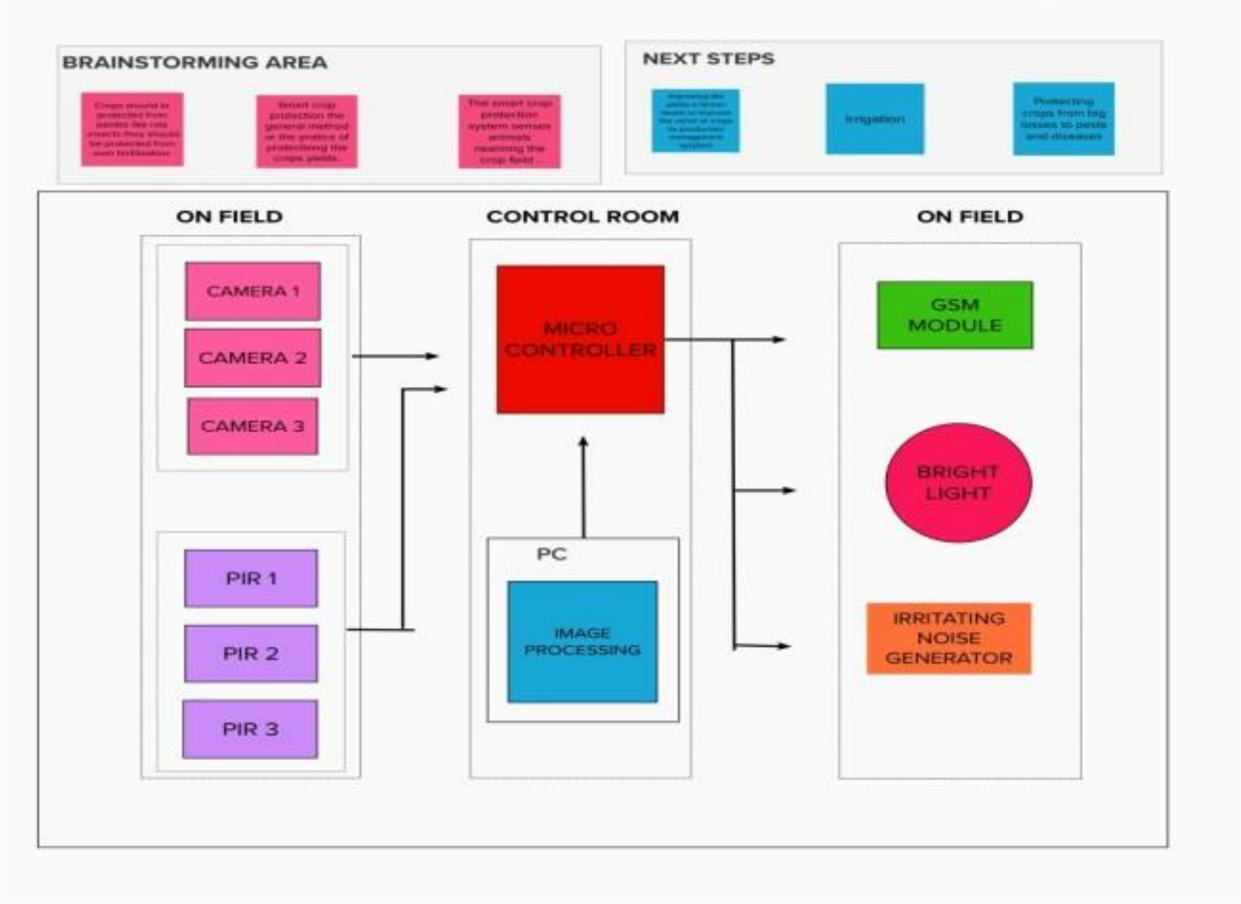
the non-functional requirements of the proposed solution.

FR No	Non-Functional Requirement	Description
NFR-1	Usability	Mobile support. Users must be able to interact in the same roles & tasks on computers & mobile devices where practical, given mobile capabilities.
NFR-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.
NFR-3	Reliability	It has a capacity to recognize the disturbance Near the field and doesn't give a false caution signal.
NFR-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.
NFR-5	Availability	IoT solutions and domains demand highly available systems for 24x7 operations. Isn't a *critical production* application, which means that operations or production don't go down if the IoT solution is down.
NFR-6	Scalability	System must handle expanding load and data retention needs that are based on the upscaling of the solution scope, such as extra manufacturing facilities and extra buildings.

5. PROJECT DESIGN

Data Flow Diagrams:

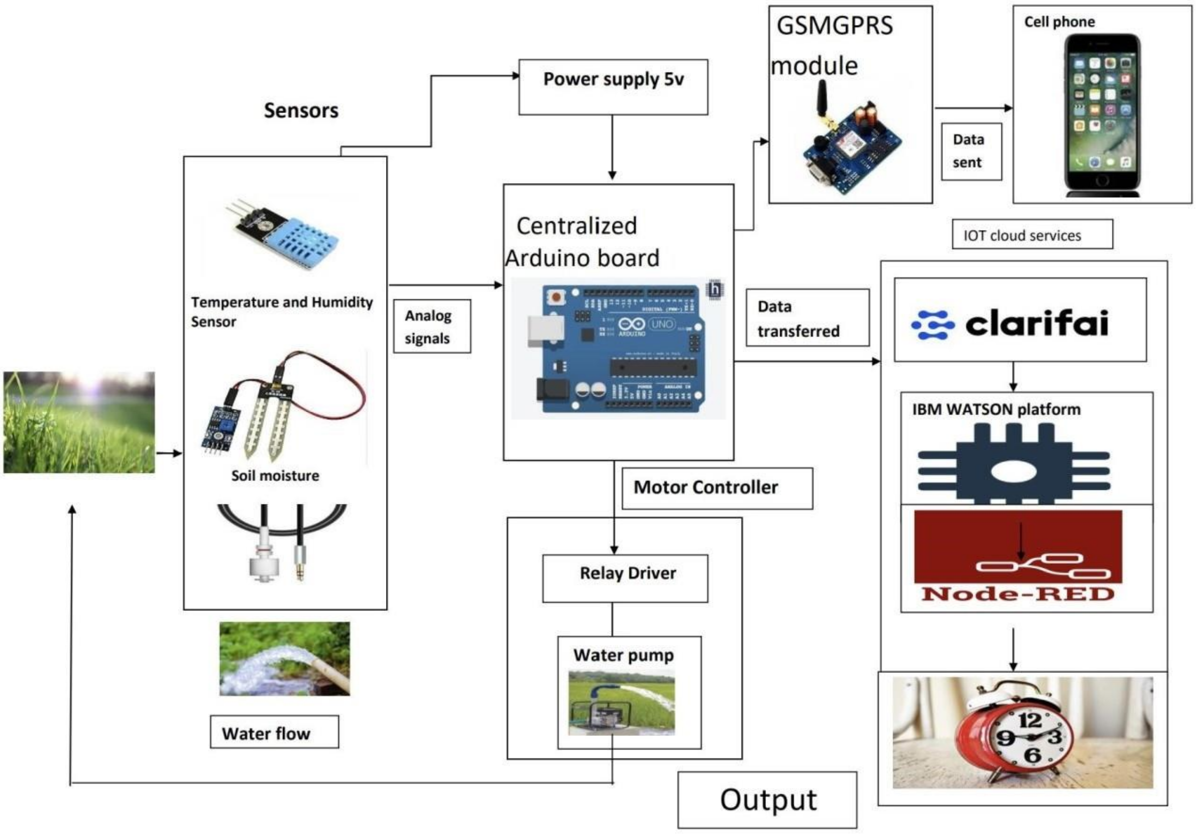



BRAINSTORMING AREA
Smart crop protection the general method or the pratice of protectiong the crops yields.
The smart crop protection system senses animals nearning the crop field.
NEXT STEPS
Irrigation
Protecting crops from big losses to pests and diseases
ON FIELD
CAMERA 1
CAMERA 2
CAMERA 3
PIR 1
PIR 2
PIR 3
CONTROL ROOM
MICRO CONTROLLER
PC
IMAGE PROCESSING
ON FIELD
GSM MODULE
BRIGHT LIGHT
IRRITATING NOISE GENERATOR

Solution & Technical Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.



Explanation for the Architecture Diagram:

- The device will detect the animals and birds using the Clarifai service.
- If any animal or bird is detected the image will be captured and stored in the IBM Cloud object storage.
- It also generates an alarm and avoid animals from destroying the crop.

- It also generates an alarm and avoid animals from destroying the crop.

- The image URL will be stored in the IBM Cloudant DB service.
- The device will also monitor the soil moisture levels, temperature, and humidity values and send them to the IBM IoT Platform.
- The image will be retrieved from Object storage and displayed in the web application.
- A web application is developed to visualize the soil moisture, temperature, and humidity values.
- Users can also control the motors through web applications.

User Stories:

User type	Functional Requirement	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
User	Data collecting	USN-1	Smart farming based IOT	Sensing of Agriculture data and storing.	High	Sprint-4
		USN-2	As an user, I will inform the farmer to protect the crops.	I can inform the farmer.	Medium	Sprint-2
		USN-3	Extract data from source.	Management of data through expert and investigation method	High	Sprint-2
User 2	Login	USN-4	As an co-user, I can send the alert message to the farmers.	I can alert farmers.	High	Sprint-1
Farmer	Login	USN-5	As a farmer, I will follow the route to the crop which can avoid are detect animal intrusion.	I can reach the crops.	High	Sprint-2
Crop Protector		USN-6	A an crop protector.	In can protect the crop.	Medium	Sprint-2
Farmer	Login	USN-7	As a supervisior, I can Supervise the crop an ensure the hygiene proces	I can manage all these sprocess going good.	High	Spirit-1
Crop yielder	Register	USN-8	As a crop yielder,I can yield more crop.	I can register smart crop.	Medium	Spirit-3
Crop Monitor		USN-9	As a crop monitor,I check the quality of IIOTdevice's quality.	I can check the IOT device.	Medium	Spirit-3

CHAPTER – 6

PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1		US-1	Create the IBM Cloud services which are being used in this project.	6	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-2		US-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	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-2		US-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	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-3		US-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	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-3		US-2	Create a Node-RED service.	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R

Sprint-3		US-1	Develop a python script to publish random sensor data such as temperature, moisture, soil and humidity to the IBM IoT platform	7	High	Arshad Parvez G Lavanya Kishore Kumar K Mathuprakas R
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R
Sprint-4		US-2	Configure the Node-RED flow to receive data from the IBM IoT platform and also use Cloudant DB nodes to store the received sensor data in the cloudant DB	10	High	Arshad Parvez G Lavanya M Kishore Kumar K Mathuprakas R

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

Sprint Delivery Schedule:

TITLE	DESCRIPTION	DATE
Literature Survey on The Selected Project and Information Gathering	A Literature Survey is a compilat ion summary of research done previously in the given topic. Literature survey can be taken from books, research paper online or from any source.	19 September 2022
Prepare Empathy Map	Empathy Map is a visualization tool which can be used to get a better insight of the customer	19 September 2022
Ideation-Brainstorming	Brainstorming is a group problem solving session where ideas are shared, discussed and organized among the team members.	19 September 2022
Define Problem Statement	A Problem Statement is a concise description of the problem or issues a project seeks to address. The problem statement identifies the current state, the desired future state and any gaps between the two.	19 September 2022
Problem Solution Fit	This helps us to understand the thoughts of the customer their likes, behaviour, emotions etc.	12 October 2022
Proposed Solution	Proposed solution shows the current solution and it helps is going towards the desired result until it is achieved.	12 October 2022
Solution Architecture	Solution Architecture is a very complex process I.e it has a lot of sub-processes and branches. It helps in understand ing the components and features to complete our project.	12 October 2022
Customer Journey	It helps us to analyse from the perspective of a customer, who uses our project.	15 October 2022
Functional Requirement	Here functional and nonfunctio na l requirements are briefed. It has specific features like usability, security, reliabil it y, performance, availability and scalability.	15 October 2022
Data Flow Diagrams	Data Flow Diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.	15 October 2022

Technology Architecture	Technology Architecture is a more well defined version of solution architecture. It helps us analyze and understand various technologies that needs to be impleme nted in the project.	15 October 2022
Prepare Milestone & Activity List	It helps us to understand and evaluate our own progress and accuracy so far.	29 October 2022
Spring Delivery Plan	Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.	In Progress

CHAPTER -7

CODING & SOLUTIONING

Feature 1:

Coding for Animals or pests enter into the field from pygame

```
import mixer

class SoundPlayer:

     def init (self, sound_file):
    mixer.init(44100, -16, 2, 2048)
    self.sound =mixer.Sound(sound_file)
    def play(self): self.sound.pla y()
    import time
      class FPS:
     def init
     (self):
          self.frame_count = 0
          self.elapsed_time = 0
          def start(self):
          self.start_time = time.time()    def stop(self):
           self.stop_time     =     time.time()
           self.frame_count += 1
           self.elapsed_time += (self.stop_time-self.start_time)
            def count(self):
return
    self.frame_count
     def elapsed(self):
return
     self.elapsed_time  def fps(self):
     if
    self.elapsed_time==0:
return 0
else:
     return
self.frame_count/self.elapsed_time
```

Feature 2: coding for moisture level checking

```
Import RPi.GPIO as GPIO

Import time channel=21
       GPIO.setmode(GPIO.BCM) GPIO.setup(channel,GPIO.IN)
       def callback(channel):

 if
 GPIO.input(channel):
      print(“no water detected”)
else:
     print(“water detected”)

GPIO.add_event_detect(channel,GPIO.BOTH,bouncetime=300)

GPIO.add_event_callback(channel,callback)
 while True:

               time.sleep(1)
```

Feature 3: Detect The PH Level of Crops

```
import io # used to create file streams

import fcntl # used to access I2C parameters like addresses

import time # used for sleep delay and timestamps class Ezo:

long_timeout = 1.5

# the timeout needed to query readings and #calibrations short_timeout = .5

# timeout for regular commands default_bus = 1

# the default bus for I2C on the newer Raspberry Pis,

# certain older boards use bus 0 default_address = 99

# the default address for the pH sensor

def init (self, address=default_address, bus=default_bus):

# open two file streams, one for reading and one for writing

# the specific I2C channel is selected with bus

# it is usually 1, except for older revisions where its 0

# wb and rb indicate binary read and write self.file_read = io.open("/dev/i2c-" + str(bus),
"rb", buffering=0) self.file_write = io.open("/dev/i2c-" + str(bus), "wb", buffering=0)

# initializes I2C to either a user specified or default address self.set_i2c_address(address)

def set_i2c_address(self, addr):

# set the I2C communications to the slave specified by the address

# The commands for I2C dev using the ioctl functions are specified in

# the i2c-dev.h file from i2c-tools I2C_SLAVE = 0x703
```

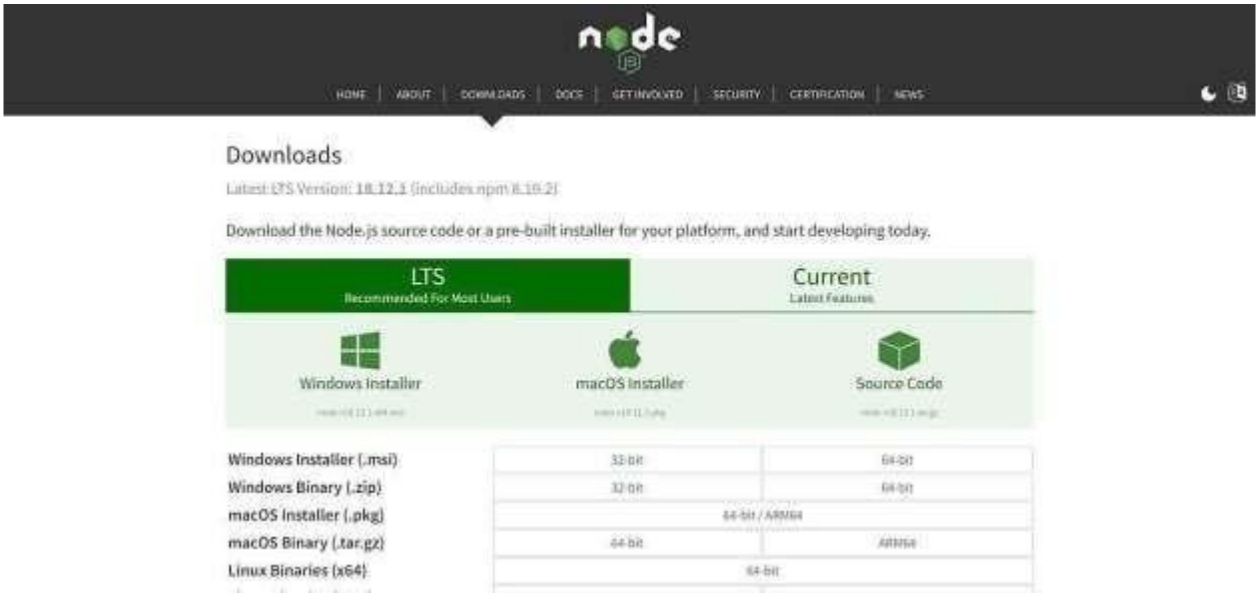
```
fcntl.ioctl(self.file_read, I2C_SLAVE, addr) fcntl.ioctl(self.file_write, I2C_SLAVE, addr)

def write(self, string):

# appends the null character and sends the string over I2C  string += "\00"
self.file_write.write(bytes(string, 'UTF-8'))

def read(self, num_of_bytes=31):

# reads a specified number of bytes from I2C,

# then parses and displays the result res = self.file_read.read(num_of_bytes)

 # read from the board

 # remove the null characters to get the response response = [x for x in res if x != ' \x00'] if
response[0] == 1:

 # if the response isnt an error

 # change MSB to 0 for all received characters except the first

 # and get a list of characters

char_list = [chr(x & ~0x80) for x in list(response[1:])]

# NOTE: having to change the MSB to 0 is a glitch in the

 # raspberry pi, and you shouldn't have to do this!

 # convert the char list to a string and returns it

#return "Command succeeded " +

return''.join(char_list)

else:

    return "Error " + str(response[0])

def query(self, string):

# write a command to the board, wait the correct timeout,

# and read the response self.write(string)
# the read and calibration commands require a longer timeout
if((string.upper().startswith("R"))
or
      (string.upper().startswith("CAL"))):
time.sleep(self.long_timeout)
elif((string.upper().startswith("SLEEP"))):
return
    "sleep mode"
else:
time.sleep(self.short_timeout)
return self.read() def close(self):
self.file_read.close()
self.file_write.close()


#ph = Ezo()
#phvalue = ph.query('R')
 #ph1 = str(phvalue)
 #ph2 = round(phvalue)
 #print (ph.query('R'))
#print (round(ph.query('R'),2))
```

8. TESTING:

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%	

USER ACCEPTANCE TESTING:


HOME | ABOUT | DOWNLOADS | DOCS | GET INVOLVED | SECURITY | CERTIFICATION | NEWS

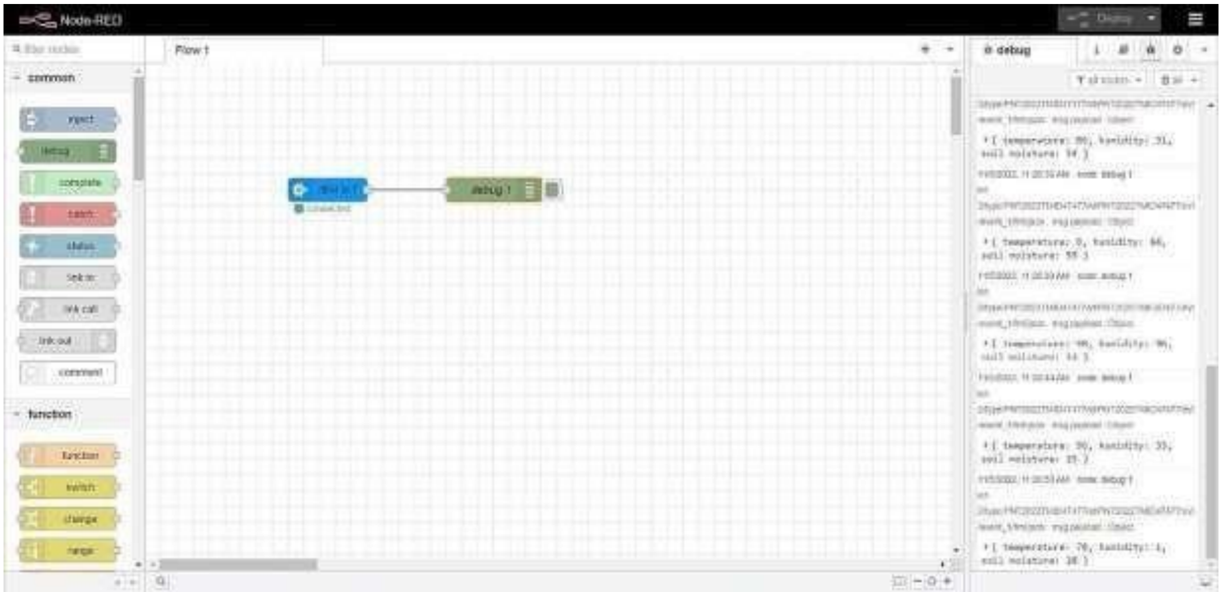
Downloads

Latest LTS Version: 18.12.1 (includes npm 8.19.2)

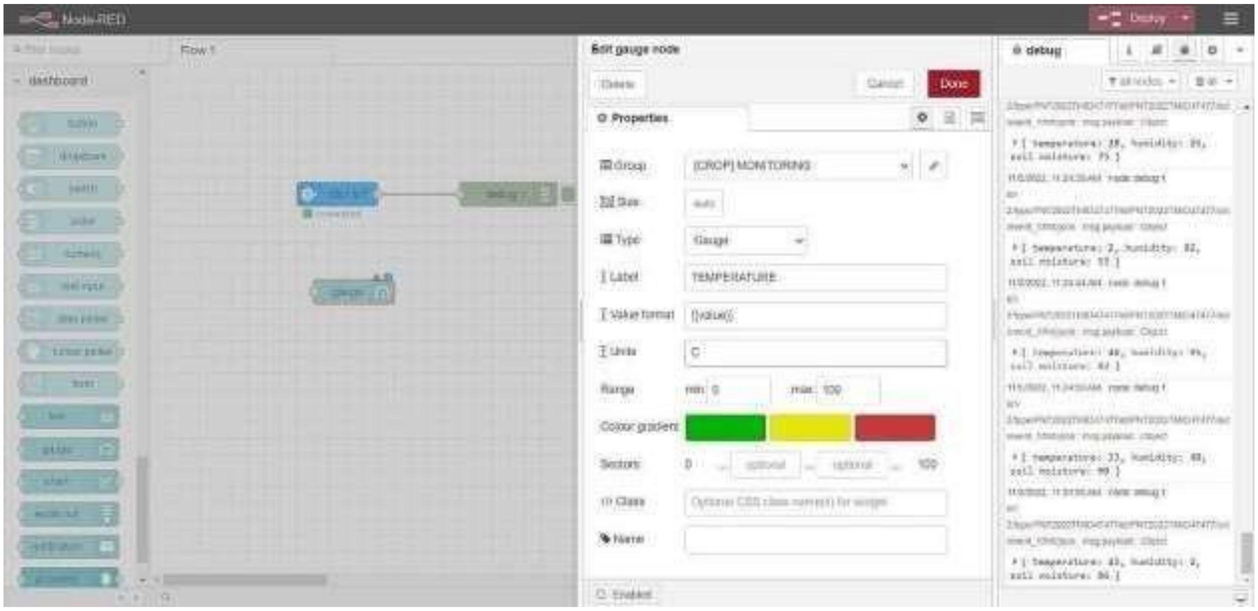
Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS
Recommended For Most Users
Current
Latest Features
Windows Installer
macOS Installer
Source Code

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	

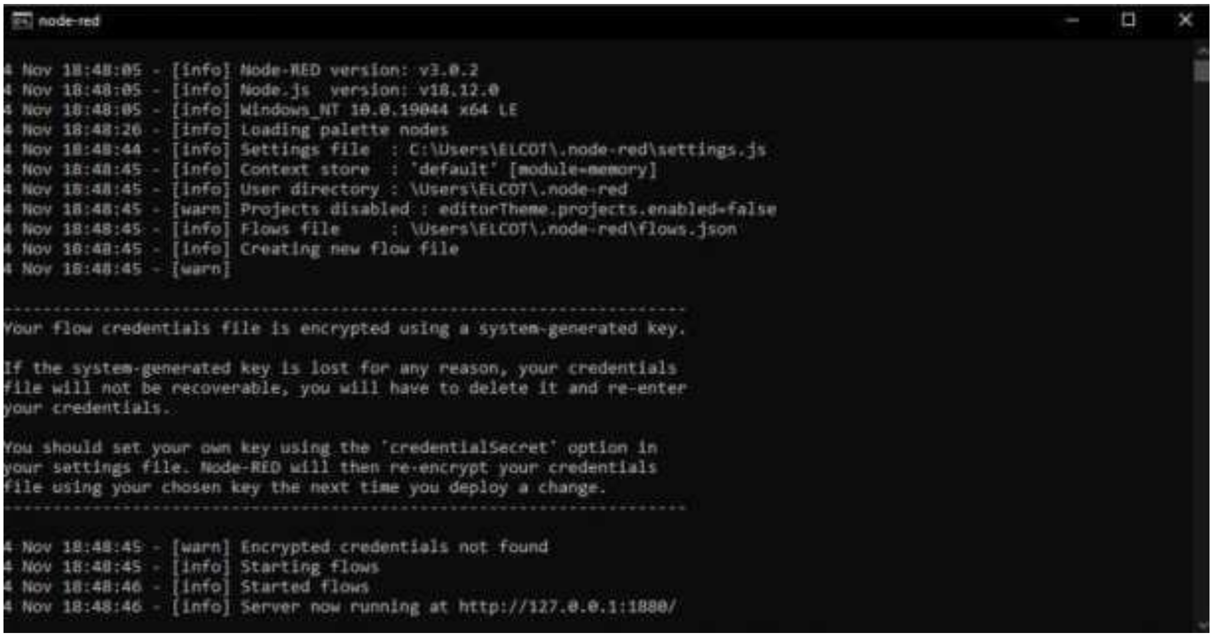


```
▸{ temperature: 80, humidity: 31, soil moisture: 14 }
[illegible] node: debug 1
▸{ temperature: 0, humidity: 66, soil moisture: 55 }
[illegible] node: debug 1
▸{ temperature: 40, humidity: 96, soil moisture: 14 }
[illegible] node: debug 1
▸{ temperature: 50, humidity: 33, soil moisture: 15 }
[illegible] node: debug 1
▸{ temperature: 70, humidity: 1, soil moisture: 38 }
```



Group: [CROP] MONITORING
Size: auto
Type: Gauge
Label: TEMPERATURE
Value format: {{value}}
Units: C
Range: min: 0 max: 100
Colour gradient: 🟩 🟨 🟥
Sectors: 0 ... optional ... optional ... 100
Class: Optional CSS class name(s) for widget
Name:

```
▸{ temperature: 28, humidity: 39, soil moisture: 75 }
11/5/2022, 11:24:35.444 node: debug 1
▸{ temperature: 2, humidity: 82, soil moisture: 55 }
11/5/2022, 11:24:44.444 node: debug 1
▸{ temperature: 48, humidity: 96, soil moisture: 82 }
11/5/2022, 11:24:55.444 node: debug 1
▸{ temperature: 33, humidity: 48, soil moisture: 99 }
11/5/2022, 11:25:05.444 node: debug 1
▸{ temperature: 43, humidity: 2, soil moisture: 84 }
```



```
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/
```

9. RESULTS:

Crop vandalism caused by wild animals and fire is currently a significant social issue.

Given that there is currently no working remedy for this issue, it needs urgent attention. As a result, this project has significant social significance because it seeks to solve this issue. This project will assist farmers in safeguarding their orchards and fields, save them from suffering major financial losses, and spare them from making futile efforts to safeguard their fields. They will also benefit from higher crop yields, which will improve their economic situation.

10. ADVANTAGES & DISADVANTAGES:

ADVANTAGE:

The main benefit is that it keeps animals away from the crops on farmland. The ultrasonic sensor primarily picks up on temperature, humidity, and soil moisture. It also picks up on birds entering fields. This technology will constantly monitor the field's soil properties.

DISADVANTAGE:

controlled access to food. If you are cultivating the crops and breeding them to be more resilient, you have a better chance of avoiding droughts or floods. It enables farmers to increase yields while utilising the least amount of water and fertiliser.

11. CONCLUSION:

In India's rural areas, farmers face serious dangers like animal and bird damage. Therefore, in order to solve this problem, a system that plays frightening noises to make animals and birds flee automatically was created. As a result, the developed system is cost-effective and beneficial to farmers. The system is safe for people and animals to use, and also safeguards farmland.

12. FUTURE SCOPE:

The application of this system will have a broad range in the future. Information is collected using IR and ultrasonic sensors, which are then communicated over GSM. Wi-Fi sensor networks further improve this idea. The kind of sensors that can measure the soil's moisture content, a crop's growth, and its nutritional value. These sensors collect information that helps farmers and allow them to monitor farmland from anywhere in the world.

13.APPENDIX:

SOURCE CODE:

```
import time importsys import ibmiotf.application # toinstallpipinstall ibmiotf
importibmiotf.device

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

def myCommandCallback(cmd): # function for Callbackif

cm.data['command'] == 'motoron':

print("MOTOR ON IS RECEIVED")

elif cmd.data['command'] == 'motoroff':print("MOTOR OFF IS RECEIVED")if
cmd.command == "setInterval":

else:

if 'interval' not in cmd.data:

print("Error - command is missing requiredinformation:

'interval'") interval = cmd.data['interval']elif

cmd.command == "print":

if 'message' not in cmd.data:
```

```
print("Error - commandis missing requiredinformation: 'message'")else:output =
cmd.data['message']
```

```
print(output)
try:

        deviceOptions = {"org": organization, "type": deviceType, "id":
        deviceId,"authmethod":authMethod, "auth-token": authToken}        deviceCli
        = ibmiotf.device.Client(deviceOptions)#
        ..............................................


        exceptException 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 cloud deviceCli.disconnect()
```

SENSOR.PY

```
import time import
sysimport
ibmiotf.application
importibmiotf.device
```

```
import random

# Provide your IBM Watson Device Credentials organization = "8gyz7t" # replace the ORG
ID deviceType = "weather_monitor" #replace the Device type deviceId = "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) #.............................................

exceptException 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

                                                            de

fmyOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %pulse,"Soil
Moisture = %s %%" % soil,"to IBMWatson")


success = deviceCli.publishEvent("IoTSensor", "json", data,
qos=0,on_publish=myOnPublishCallback)                                                if
notsuccess: print("Not connected to

IoTF")time.sleep(1)


deviceCli.commandCallback = myCommandCallback



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

Node-RED FLOW :

[

{

"type":"ibmiotout", "z":"630c8601c5ac3295","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","o
rder":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"]]},
{
"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"]]},

```
"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}
]
```

```
[
{
"type":"ibmiotin","z":"03acb6ae05a0c712",
"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"]]
},
{
"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"]]
},
{
"name":"Humidity",
"func":"msg.payload =
msg.payload.pulse;\nglobal.set('p',msg.payload)\nreturn msg;", "outputs":1,
"noerr":
0,
"initialize":"",
"finalize"
:"",
"libs":[
],
"x
":
48
0,
"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"
:86 0,"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":

49

0, "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":fals e, "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\":glob
al.get( 's')};\nreturn msg;",
```

"outputs":1,

"noerr":0,

"initialize":"",

"finalize":"

","libs

":[

],

"x

":

63

0,

"y":500, "wires":[["5c7996d53a445412"]]

},

{ "id":"5c7996d53a445412",

"type":"httpresponse",

"z":"03acb6ae05a0c712","na

me":"","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 & Project Demo Link

https://drive.google.com/file/d/1zfnoiLrGQFc5gkl4Ruko19t3mhN7wpka/view?usp=sharing

IBM-EPBL/**IBM-Project-11977-1659364387**