P.S.R.R COLLEGE OF ENGINEERING

IOT BASED SMART CROP PROTECTION SYSTEMS FOR AGRICULTURE

AN IBM REPORT

TEAM ID: PNT2022TMID50334

SUBMITTED BY,

R.SEEMA MARAGTHAM (952019106009)
R.KEERTHI (952019106004)
V.KARISHMA KRUTHIK (952019106003)
C.HARISTHA (952019106002)

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The main aim of our project is to protect the crops from damage caused by animal as well as divert the animal without any harm. 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. Animal detection system is designed to detect the presence of animal and offer a warning. In this project we used PIR and ultrasonic sensors to detect the movement of the animal and send signal to the controller. It diverts the animal by producing sound and signal further, this signal is transmitted to GSM and which gives an alert to farmers and forest department immediately.

Wildlife tracking involves acquiring information about the behavior of animals in their natural habitat. This information is used both for scientific and conservation purposes. The primary form of information that needs to be obtained is the location of the animal at certain points in time and this is generally referred to as tracking or radio–tracking. However, due to the similarities in obtaining the information, the terms are frequently used interchangeably. There are remote methods that can be used to track and identify animals visually and through acoustic signals. It is meaningful to design a strategy to roughly localize mobile phones without a GPS by exploiting existing conditions and devices especially in environments without GPS availability. The availability of Bluetooth devices for most phones and the existence of a number of GPS equipped phones in a crowd of phone users enable us to design a Bluetooth aided mobile phone localization strategy. With the position of GPS equipped phones as beacons, and with the Bluetooth connection between neighbor phones as proximity constraints, we formulate the problem into an inequality problem defined on the Bluetooth network. The convergence of the neural network and the solution feasibility to the defined problem are both theoretically proven.

The hardware implementation architecture of the proposed neural network is also given in this article. As applications, rough localizations of drivers in a tunnel and localization of customers in a supermarket are explored and simulated. Simulations demonstrate the effectiveness of the proposed method

1.2 PURPOSE

Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds, and fire 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 and fire. This is a arduino Uno based system using microcontroller. 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. If there is a smoke, it immediately turns ON the motor. This ensures complete safety of crops from animals and from fire thus protecting the farmer's loss. This is a arduino Uno based system using microcontroller. 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.

In the proposed system, Crop monitoring is done where sensors are used to collect information in the agricultural field. In our proposed work, PIR, Smoke sensor and GSM is used. When animals come near to the PIR sensor and it detects the animal movement. After getting that initial input signal, it is passed for further processing. Then it will be given to the microcontroller. Our system will be activated, immediately buzzer will be on, at the same time it sends an SMS and makes call to the owner. Microcontroller Block is used for reading the inputs from PIR and Smoke sensor. Whole process is controlled by microcontroller. The GSM module is used for sending SMS and making call to farmer when movement or smoke is detected. It also turns ON the motor, when smoke is detected. It alerts the farmer that some animals try to enter into the farm. Our LCD data will be display for SMS sending.

Crops are usually destroyed by these wild animals and result in large amount of less to the farmers. A constant manual guarding of the fields is not possible. To tackle this problem, in our proposed work, we shall design a system to prevent the cutry of animals into the farm and alert the farmer at the same time via a phone call.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Crops are vulnerable to attack, damage, and competition.insects, plant diseases nematodes, rodents, weeds, and air pollution are among the many enemies

Insects, for example, can destroy a crop in a relatively short time . Control measures for many years have engaged the attention of farmer and scientist, yet full success has not been achieved, and the battle continues. The problem is further complicated by the fact that control measures not only kill unwanted insects, but also may harm honey bees as well as the parasites and predators that destroy insect pests.

At least 10,000 species of insects are classed as unwanted. Of these, several hundred species are particularly destructive and require some degree of control. They destroy food as well as the forage, pasture, and grain needed to produce livestock; and, in addition, they carry and transmit many diseases of plants and animals.

Due to over population it occurs a deforestation this results in shortage of food, water and shelter in forest areas. So, Animals interference in residential areas is increasing day by day which affects human life and property causes human animal conflict but as per nature's rule every living creature on this earth has important role in eco-system. Agriculture is the backbone of the economy but because of animal interference in agricultural lands, there will be huge loss of crops. Elephants and other animals coming in to contact with humans, impact negatively in various means such as by depredation of crops, damaging grain stores, water supplies, houses and other assets, injuring and death of humans.

2.2 PROBLEM STATEMENT DEFINITION

Animal attacks in India are a common story nowadays. Due to the unavailability of any detection system these attacks kill villagers and also destroy their crops. Due to lack of proper safety measures, these villagers are left helpless to their fate. Therefore a proper detection system could help save their lives and also to the preservation of crops. Also the crops of villagers are destroyed due to frequent interference of animals. The increasing rate of decrease in forests and encroaching agriculture land is leading to an up rise in animal invasion of fields which has leads to a drastic change in farmers perception towards them. The harmony between a farmer and wild animals seems to be a next impossible thing.

Insecticides generally are effective, cheap, and safe if handled correctly; the good derived from them, however, can be partly offset by adverse effects. Chlorinated hydrocarbon insecticides such as DDT, for example, may leave residues toxic to beneficial insects, fish, and other wildlife; the insecticides may be found in meat and milk, or they may persist in the soil. Another problem is that some species of insects build up resistance to chlorinated hydrocarbon, organic phosphate, and carbamate insecticides. These disadvantages can be overcome only by persistent search for new and safer insecticides accompanied by wide use of nonchemical insect control.

Certain insects that attack cotton, vegetables, and forage crops may be controlled by chemicals absorbed by the plant. Called systemics, they are placed with the seed at planting time. The chemical is taken up by the plant, and insects die when they attempt to feed on the leaf or stem. Beneficial insects that do not feed on the plant remain unharmed.

REFERENCES

[1]. Mriganka Gogoi and Savio Raj Philip, “Protection of crops from animals using intelligent surviellance”2015 by Journal of Applied and Fundamental Sciences.

[2]. S. Santhiya, Y. Dhamodharan, N E. KaviPriya, C S. Santhosh , “A smart farmland using raspberry pi crop prevention and animal intrusion detection system” presented in International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 03 | Mar-2018

[3]. Stefano Giordano, IliasSeitanidis and Mike Ojo, Davide Adami ,“IoT Solutions for Crop Protection against Wild Animal Attacks” 2018 IEEE International Conference on Environmental Engineering (EE) in march 2018

[4]. Krishnamurthy b, International Journal of Latest Engineering Research and Applications (IJLERA) ISSN:2455-7137, Volume – 02, Issue – 05, May – 2017, PP –128-135.

[5]. Bindu D , International Journal of Engineering, Basic sciences, Management & Social studies, Volume 1, Issue 1, May 2017.

IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION AND BRAIN STORMING

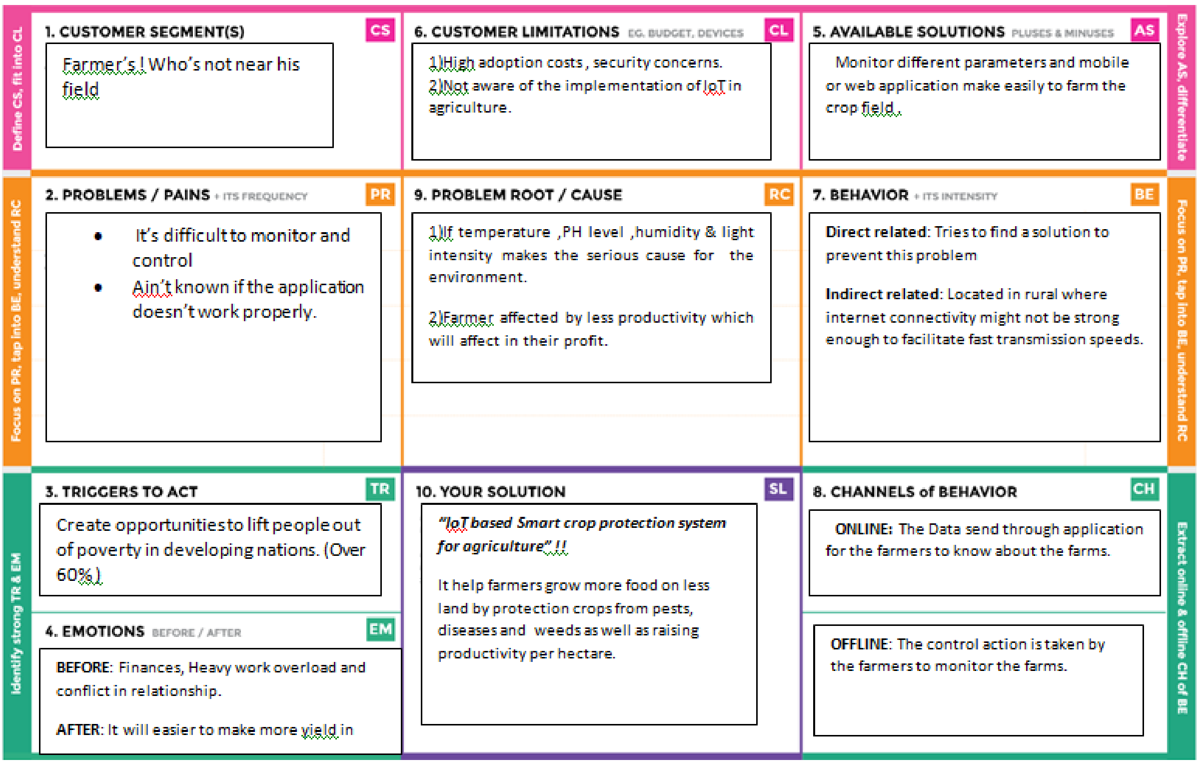
3.3 PROPOSED SOLUTION

Proposed solution template

S.no	Parameter	Description
1.	Problem statement (problem to be solved)	Agriculture assumes a significant job for advancement in nourishment creation and crop protection in India. Here, agriculture relies upon disproportionate rain which thereby affects India's agriculture. There arises a need for effective irrigation for the agricultural production. The protection is done through the voice detection and movement detection methods to enable high frequency sound, hence protecting the crops from insects, pests and small animals.
2.	Idea / Solution description	To control over how much water is to be supplied and when it is to be applied determines the uniformity which is key to maximizing the irrigation efforts. The proper irrigation management takes careful consideration and vigilant observations. Irrigation framework would automatically begin/stop water siphons, on the agricultural site depending upon the dampness content obtained by the moisture sensor as soon as it senses the level of water in the reservoir. The deliberate sensor estimates are sent to the Arduino Uno microcontroller for arranging the controlled calculation.

3.	Novelty / Uniqueness	It help formers grow more food on less land by protecting crops from pests ,diseases and weeds as well as raising productivity per hectare.
4.	Social impact / Customer Satisfaction	In rural parts of India, farmers encounter severe threats such as damage done by animals. Hence, to overcome this issue we have designed a system in which sound is played and by using LDR it detects light intensity, if it is less, it will focus the light. So that wild animals will not enter into the farm. It will run away.
5.	Business mode (revenue model)	Behaviour model: Wildlife tracking involves acquiring information about the behavior of animals in their natural habitat. This information is used both for scientific and conservation purposes. Proposition of model: The primary form of information that needs to be obtained is the location of the animal at certain points in time and this is generally referred to as tracking or radio–tracking. However, due to the similarities in obtaining the information, the terms are frequently used interchangeably. There are remote methods that can be used to track and identify animals visually and through acoustic signals.
6.	Scalability of the solution	Agriculture is the main livehood for majority of people living in rural areas .It contributes 17-18% of GDP at present. India is a majorproducer or wheat,rice and barley however in comparison with china the yield is low

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Install the app. Signing up with Gmail Creating a profile. Understand the instructions.
FR-2	User Confirmation	Email verification required via OTP or gmail.
FR-3	Accessing datasets	Data's are obtained by cloud.
FR-4	Interface sensor	Connect the sensor ,controller connected with server and the application. When animals enter the field , the alarm is generated.
FR-5	Mobile application	It is used to monitor the crops from insects ,animals, bird and field sprinklers.

4.2 Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	By using smart protection systems the crops are protected from animals and birds.
NFR-2	**Security**	It was constructed for protecting the crops.
NFR-3	**Reliability**	It help farmers grow more food on less land by protecting crops from pests, diseases and weeds as well as raising productivity per hectare.
NFR-4	**Performance**	The iot devices, sensors which are connected as the systems will monitor the crops and the data will send via messages to farmers when the animals are entered in the field .
NFR-5	**Availability**	We can defend the crops against wild animals by creating and implementing hardware and software.
NFR-6	**Scalability**	This systems is more scalable which has an efficient way to retrieve the data from ibm cloud services.

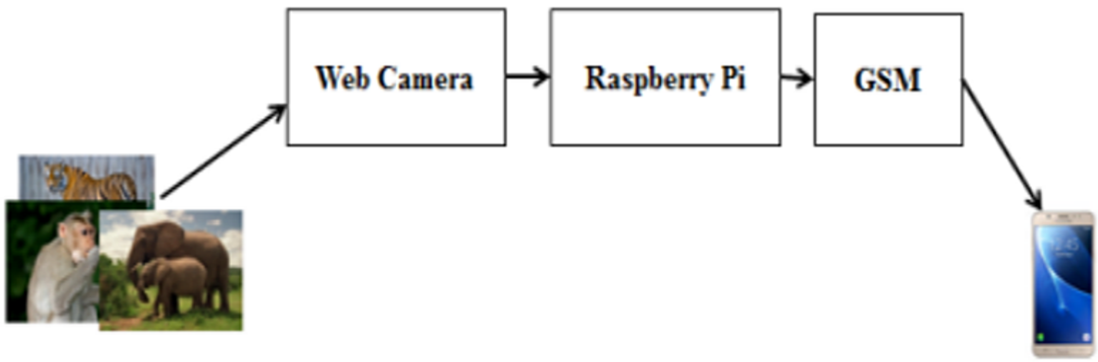
5. PROJECT DESIGN

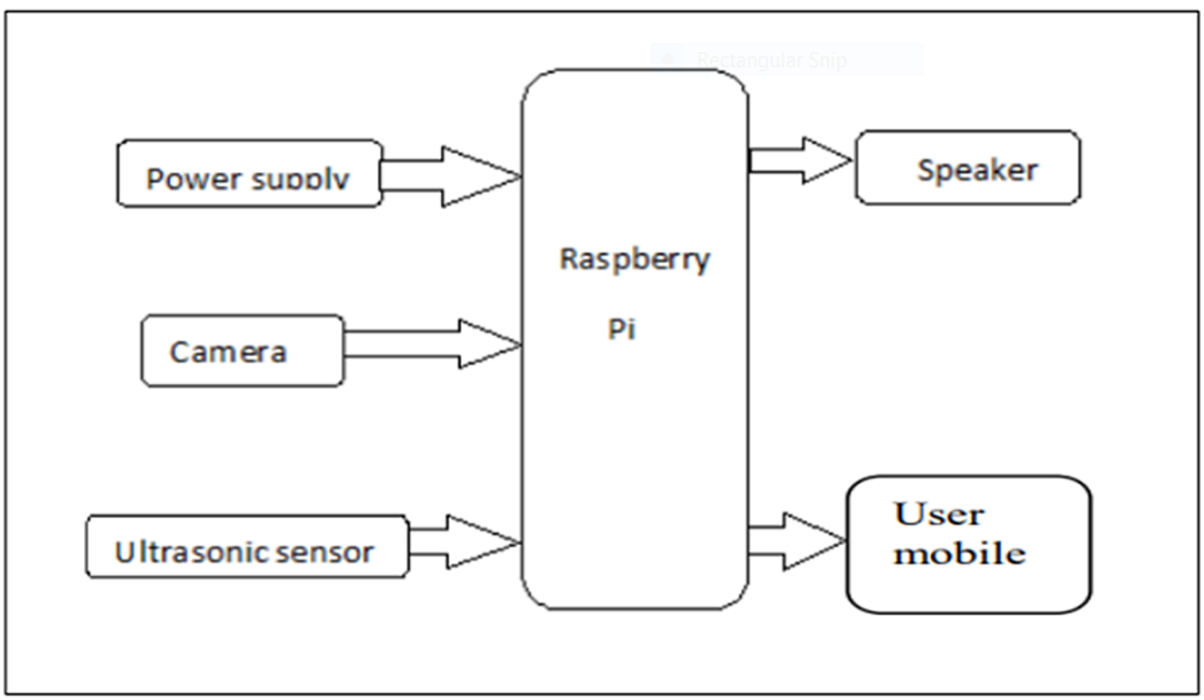
5.1 DATA FLOW DIAGRAM

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

FLOW CHART





CROP PROTECTION FLOW DIAGRAM

5.2 user stories:

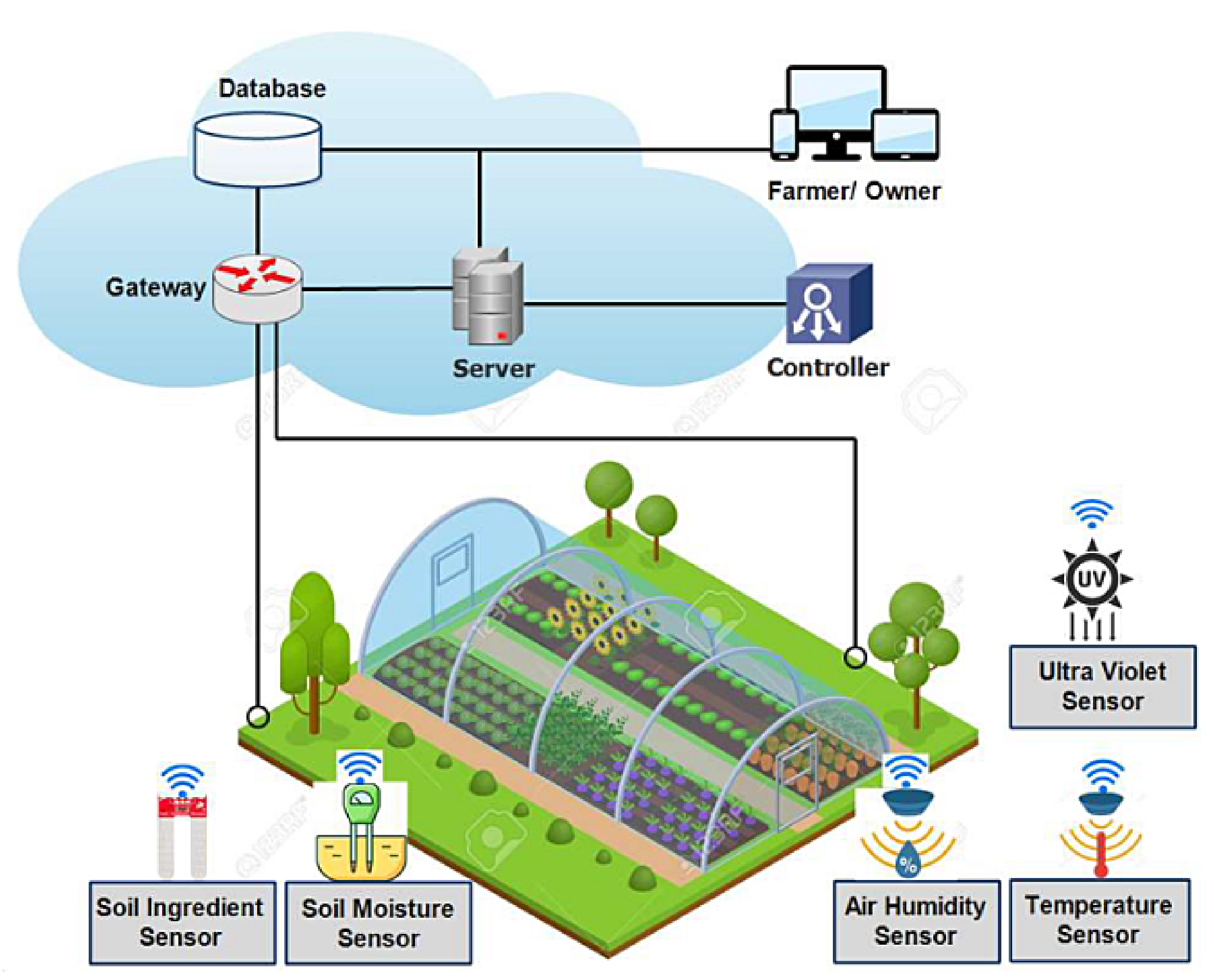
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)						
Customer Care						
Executive						

5.3 SOLUTION AND TECHNICAL ARCHITECTURE

5.3.1 SOLUTION ARCHITECTURE

A greenhouse consists of walls and a roof, which are usually made from transparent materials, such as plastic or glass. In a greenhouse, plants are grown in a controlled environment, including controlling for moisture, nutrient ingredients of the soil, light, temperature, etc. Consequently, greenhouse technology makes it possible for humans to grow any plant, at any time, by providing suitable environmental conditions. the below figure, a smart agriculture IoT system for monitoring greenhouse farming factors based on IoT ecosystems.

An introduced IoT-based greenhouse environmental monitoring system for multipoint monitoring in large greenhouses. Instead of using multiple sensors at different locations, this solution involves a drive system that allows the sensor system to move to different locations in the greenhouse. The experimental results show that the proposed system can effectively monitor multiple points in large greenhouses. An energy-saving temperature control technology for smart greenhouses. This study proposed two intelligent control methods: active disturbance rejection control and fuzzy active disturbance rejection control. The experimental results demonstrate that the proposed technology saves over 15% of the total energy consumption of the greenhouse. We designed an intelligent IoT system to monitor and control greenhouse temperature for energy efficiency and improve crop productivity.



SOLUTION ARCHITECTURE

5.3.2 TECHNICAL ARCHITECTURE

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interact with the application via mobile phone or messages.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	java
3.	Application Logic-2	Logic for a process in the application	IBM Watson/node red
4.	Application Logic-3	Logic for a process in the	IBM Watson/node red

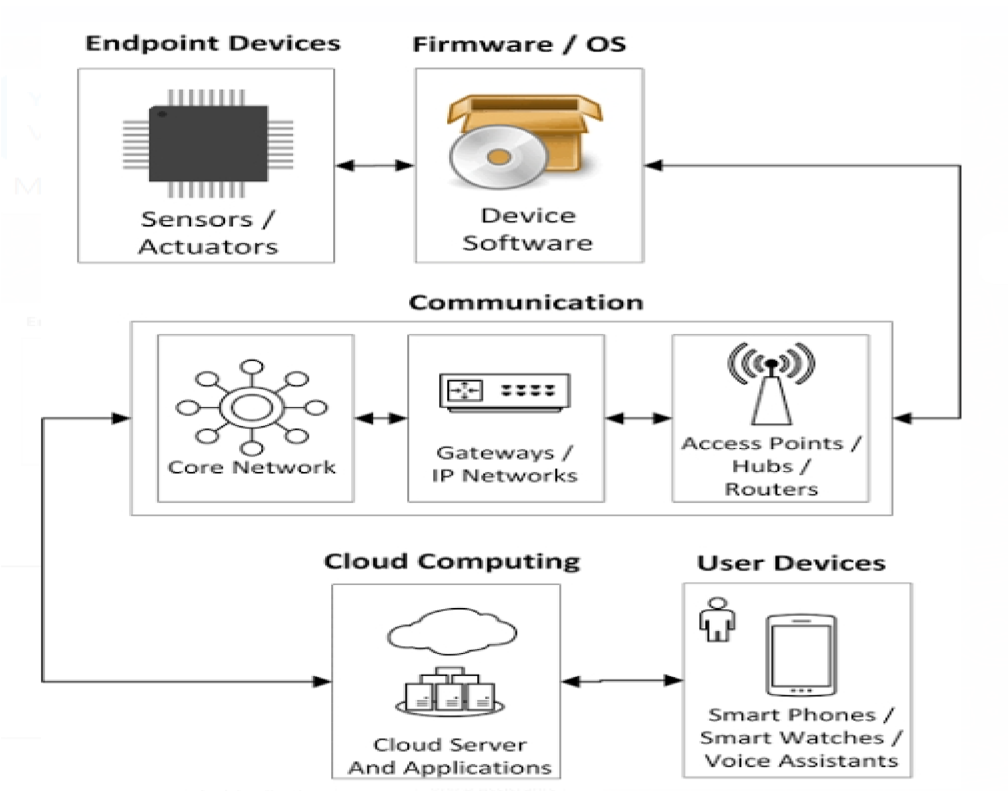
		application	
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM Cloudant, Microsoft azure.
7.	Temperature sensor	Monitor the temperature	TMP36
8.	Humidity sensor	Monitor the humidity	DHT11
9.	Soil moisture sensor	Measure the amount of water in the soil	Soil moisture sensor
10.	Weather monitoring	Monitor the weather	Temperature sensor

Table-2: Application Characteristics:

S.No	**Characteristics**	**Description**	**Technology**
1.	Open-Source Frameworks	Node red, clarifi	Software
2.	Security Implementations	Firewalls provide protection to the network by filtering insecure services.	Encryption process
3.	Scalable Architecture	Scalability is a major concern for IOT platform handle multiple IOT devices our solution due to the fact that the IOT devices do not access control information.	Software
4.	Availability	Automatic adjustment of farming equipment made devices as the ability of weather humidity, temperature throughout their service lifecycle.	Software

5.	Performance	The ideas of sensor networks agriculture can be create seamless environment among farmers and crops regardless of their geographical boundaries.	Software

TECHNOLOGY STACK DIAGRAM



6. PROJECT PLANNING AND ESTIMATION

6.1 SPRINT PLANNING , ESTIMATION AND DELIVERY SCHEDULE

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	Seema Keerthi Karishma kruthik harshitha
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	Seema Keerthi Karishma kruthik harshitha
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	Seema Keerthi Karishma kruthik harshitha
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	Seema Keerthi Karishma kruthik harshitha
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	Seema Keerthi Karishma kruthik harshitha
Sprint-3		US-2	Create a Node-RED	10	High	Seema

			service.			Keerthi Karishma kruthik harshitha
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	Seema Keerthi Karishma kruthik harshitha
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	Seema Keerthi Karishma kruthik harshitha
Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	Seema Keerthi Karishma kruthik harshitha
Sprint-4		US-1	Create Web UI in Node- Red	10	High	Seema Keerthi Karishma kruthik harshitha
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	Seema Keerthi Karishma kruthik harshitha

Project Tracker, Velocity & Burndown Chart:

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:

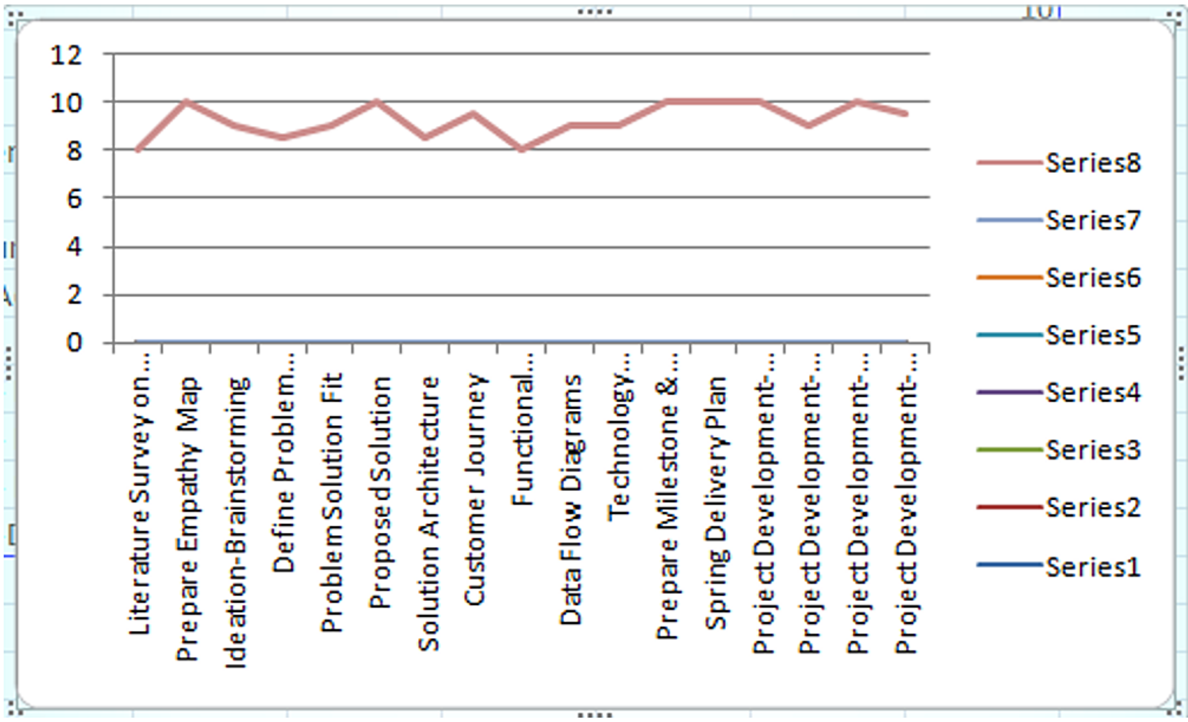
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

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

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time.It is often used in agile software development methodologies such as scrum . However, burndown charts can be applied to any project containing measurable progress overtime.

BURNDOWN CHART



7. CODING & SOLUTIONING

7.1 FEATURES 1

The first major threat to the farmers is drought. Crop vandalization by animals is the second major threat after drought. Crops are vulnerable to animals. Therefore, it is very important to monitor the nearby presence of animals.

The main aim of this project is to provide a better solution in order to resolve this problem. In this paper we proposed a method which could detect the presence of animal and offer a warning.

In this project we used microcontroller and camera to detect the movement of animals send signal to the controller. It diverts the animal by producing sound and signal further, transmitted to GSM and which gives an alert to the owner of the crop immediately. The proposed monitoring scheme is to provide an early warning about possible intrusion and damage by animals.

This system helps us to keep away such wild animals from the farmlandsand it is also an automated depending on the need so that there is no manual work, thereby saving time and also preventing the loss of crops.

ALGORITHM

Step 1. If any animals' movement was found

Step 2. Goto step 3, else go to step 9.

Step 3. The sensors capture the data and send it to the aurdino.

Step 4. Aurdino and GSM collects the data from the sensors.

Step 5. Send those data to the cloud server.

Step 6. Server process the data received from the aurdino.

Step 7. Send the signals to the atuaters, and mobiles of the users.

Step 8. The actuators perform the actions based on the data from the server.

Step 9. Go to step 1.

7.2 FEATURES 2

This project is smart crop protection system for protect the farm from animals as well as unknown person. This projects contents ardiuno UNO, Nodemcu, LCD display, PIR sensor,flame sensor,sd card module,solar panal,solar charges converter. This whole project is work on 12v dc supply from battery. We used solar panel to charge the battery.

1. Simple circuit and easy to operate.

2. Fast and simple installation and automatic operation.

3. Location flexibility.

4. Low cost maintainence.

8. TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	c for automation (Y/N	BUG ID	Executed By
LoginPage_TC_001	Home Page		Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	https://shopenzer.com/	Login/Signup popup should display	Working as expected	Pass				
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	https://shopenzer.com/	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass			BUG-1234	
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button	Username: chalam@gmail.com password: Testing123	User should navigate to user account homepage	Working as expected	pass				

					3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button								
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: chalam @gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: chalam @gmail.com password: Testing12367868678687 6876	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass				
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into applicati		1.Enter URL(https://shopenzer.com/) and	Username: chalam passwor	Application should show 'Incorrect email or password ' validation	Working as expected	Pass				

			on with InValid credentials		click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	d: Testing123678686786876876	message.						

8.2 User Acceptance Testing

UAT Execution & Report Submission

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	12	4	1	4	22

Duplicate	1	0	2	0	3
External	3	3	0	2	5
Fixed	14	2	5	23	33
Not Reproduced	0	0	1	0	1
Skipped	0	0	2	1	2
Won'tFix	0	5	3	3	4
Total	30	14	14	33	70

Test Case Analysis

This report shows the number of test cases that have passed ,failed ,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	12	0	0	12
Security	4	0	0	4
Outsource Shipping	2	0	0	2
Exception Reporting	5	0	0	5
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS

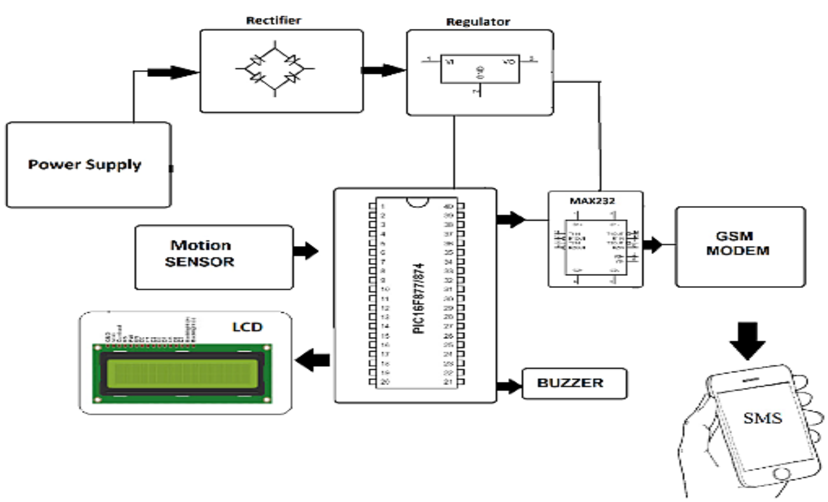
Agriculture has seen many revolutions, whether the domestication of animals and plants a few thousand years ago, the systematic use of crop rotations and other improvements in farming practice a few hundred years ago, or the "green revolution" with systematic breeding and the widespread use of man-made fertilizers and pesticides a few decades ago.

Wild animals are a special challenge for farmers throughout the world. Animals such as deer, wild boars, rabbits, moles, elephants, monkeys, and many others may cause serious damage to crops.

They can damage the plants by feeding on plant parts or simply by running over the field and trampling over the crops. Therefore, wild animals may easily cause significant yield losses and provoke additional financial problems.

Another aspect to consider is that wild animal crop protection requires a particularly cautious approach. In other words, while utilizing his crop production, every farmer should be aware and take into consideration the fact that animals are living beings and need to be protected from any potential suffering.

Block Diagram



S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Load/Voluem Changes	Risk Score	Justification
1	IOT based crop protection	New	Moderate	High	Moderate	>30 to 50 %	ORANGE	As we have seen the changes

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

1. Effective, accurate and adaptive
2. Requires no human supervision
3. Real time monitoring
4. Causes no harm to animals and humans.
5. Safeguards cultivated crops without manual observations.

10.2 DISADVANTAGES

1. System gets damage due to heavy rain and storms.
2. Network issues in remote areas.
3. Animals may cause damage to system.

11. CONCLUSION

The problem of crop vandalization by wild animals has become a major social problem in the current time. It requires urgent attention and an effective solution. Thus this project carries a great social relevance as it aims to address this problem. Hence we have designed a smart embedded farmland protection and surveillance based system which is low cost, and also consumes less energy. The main aim is to prevent the loss of crops and to protect the area from intruders and wild animals which pose a major threat to the agricultural areas. Such a system will be helpful to the farmers in protecting their orchards and fields and save them from significant financial losses and also saves them from unproductive efforts that they endure for the protection of their fields from unproductive efforts that they endure for the protection of their fields

12. FUTURE SCOPE

1.In addition to providing protection this system distinguishes between an intruder and an authorized person using RFID.

2.If the motion detection is due to an authorized person with a valid RFID, who is mostly a farm worker, his attendance gets recorded automatically.

3.We can design a IOT based application to provide an image and video feed to farmer on any smart device and farmer will be notified when there is an intrusion in the farm by animal along with additional information of humidity and temperature

13. APPENDIX

SOURCE CODE

```
import time

import sys
import ibmiotf.application
import ibmiotf.device
```

```
organization = "7oyue9"
deviceType = "nodeMCU"
deviceId = "BME280_Sensor"
authMethod = "use-token-auth"
authToken = "12345678"


def myCommandCallback(cmd):
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)


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()

deviceCli.connect()

while True:
```

```
    deviceCli.commandCallback = myCommandCallback

deviceCli.disconnect()
```

Node Red Flow.json


```
[{
    "id": "6a097760.653918",
    "type": "tab",
    "label": "IBMIOT(smart Agriculture)",
    "disabled": false,
    "info": ""
}, {
    "id": "4fdd8d20.76a9b4",
    "type": "ibmiot in",
    "z": "6a097760.653918",
    "authentication": "apiKey",
    "apiKey": "233183d6.16ba7c",
    "inputType": "evt",
    "logicalInterface": "",
    "ruleId": "",
    "deviceId": "BME280_Sensor",
    "applicationId": "",
    "deviceType": "ESP32_Controller",
    "eventType": "+",
    "commandType": "",
    "format": "json",
    "name": "IBM IoT",
    "service": "registered",
    "allDevices": "",
    "allApplications": "",
    "allDeviceTypes": "",
    "allLogicalInterfaces": "",
    "allEvents": true,
    "allCommands": "",
    "allFormats": "",
    "qos": 0,
    "x": 130,
    "y": 440,
    "wires": [
        ["8d0c40d3.848cd", "6aa78b4a.da3eb4", "5642999c.ed7868",
```

```
"c396573b.e8d738"]
    ]
}, {
    "id": "c396573b.e8d738",
    "type": "debug",
    "z": "6a097760.653918",
    "name": "",
    "active": false,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "x": 770,
    "y": 360,
    "wires": []
}, {
    "id": "8a49f2d5.e9d07",
    "type": "ui_gauge",
    "z": "6a097760.653918",
    "name": "",
    "group": "28e6141.0c047ec",
    "order": 0,
    "width": "6",
    "height": "4",
    "gtype": "gage",
    "title": "Humidity",
    "label": "%Percentage",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": ["#00b500", "#e6e600", "#ca3838"],
    "seg1": "",
    "seg2": "",
    "x": 800,
    "y": 540,
    "wires": []
}, {
    "id": "9e820fb2.1ded5",
    "type": "ui_gauge",
    "z": "6a097760.653918",
    "name": "",
    "group": "28e6141.0c047ec",
```

```
        "order": 0,
        "width": "6",
        "height": "4",
        "gtype": "gage",
        "title": "Temperature",
        "label": "°C Celcius",
        "format": "{{value}}",
        "min": 0,
        "max": "100",
        "colors": ["#00b500", "#e6e600", "#ca3838"],
        "seg1": "",
        "seg2": "",
        "x": 770,
        "y": 660,
        "wires": []
    }, {
        "id": "6aa78b4a.da3eb4",
        "type": "function",
        "z": "6a097760.653918",
        "name": "Temperature",
        "func": "msg.payload=msg.payload.d.temperature;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 410,
        "y": 560,
        "wires": [
            ["c396573b.e8d738", "9e820fb2.1ded5", "687d6f13.98f7c"]
        ]
    }, {
        "id": "8d0c40d3.848cd",
        "type": "function",
        "z": "6a097760.653918",
        "name": "Humidity",
        "func": "msg.payload=msg.payload.d.humidity;\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 420,
        "y": 500,
        "wires": [
            ["c396573b.e8d738", "8a49f2d5.e9d07", "a4f00796.520788"]
        ]
    }, {
        "id": "5642999c.ed7868",
```

```
    "type": "function",
    "z": "6a097760.653918",
    "name": "SoilMoisture",
    "func": "msg.payload=msg.payload.d.objectTemp;\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 430,
    "y": 440,
    "wires": [
        ["c396573b.e8d738", "dad1ab68.86f798", "9888ac53.4a285"]
    ]
}, {
    "id": "dad1ab68.86f798",
    "type": "ui_gauge",
    "z": "6a097760.653918",
    "name": "",
    "group": "28e6141.0c047ec",
    "order": 2,
    "width": "6",
    "height": "4",
    "gtype": "gage",
    "title": "Soil Moisture",
    "label": "% Percentage",
    "format": "{{value}}",
    "min": 0,
    "max": "100",
    "colors": ["#00b500", "#e6e600", "#ca3838"],
    "seg1": "",
    "seg2": "",
    "x": 810,
    "y": 420,
    "wires": []
}, {
    "id": "9de2a117.06e1d",
    "type": "http request",
    "z": "6a097760.653918",
    "name": "",
    "method": "GET",
    "ret": "obj",
    "paytoqs": false,
    "url":
"http://api.openweathermap.org/data/2.5/weather?q=Ponda,IN&appid=c17ea99bb
f41216723c2071ce90c3633",
```

```
        "tls": "",
        "persist": false,
        "proxy": "",
        "authType": "",
        "x": 510,
        "y": 240,
        "wires": [
            ["c396573b.e8d738", "91b4e81a.972888", "4bcf3c9.21fd4c4",
    "2c496973.5626d6", "3552343c.1a23ac"]
        ]
    }, {
        "id": "cbdf50d7.8bd57",
        "type": "inject",
        "z": "6a097760.653918",
        "name": "",
        "topic": "",
        "payload": "",
        "payloadType": "date",
        "repeat": "5",
        "crontab": "",
        "once": true,
        "onceDelay": "5",
        "x": 150,
        "y": 300,
        "wires": [
            ["9de2a117.06e1d"]
        ]
    }, {
        "id": "f8fb8426.88b758",
        "type": "ibmiot out",
        "z": "6a097760.653918",
        "authentication": "apiKey",
        "apiKey": "233183d6.16ba7c",
        "outputType": "cmd",
        "deviceId": "BME280_Sensor",
        "deviceType": "ESP32_Controller",
        "eventCommandType": "command",
        "format": "json",
        "data": "Data",
        "qos": 0,
        "name": "IBM IoT",
        "service": "registered",
        "x": 560,
```

```
        "y": 100,
        "wires": []
    }, {
        "id": "2deb666d.10728a",
        "type": "ui_button",
        "z": "6a097760.653918",
        "name": "",
        "group": "d251626d.10cec",
        "order": 2,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "Motor on",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "icon": "",
        "payload": "{\"command\":\"motoron\"}",
        "payloadType": "json",
        "topic": "",
        "x": 160,
        "y": 60,
        "wires": [
            ["f8fb8426.88b758", "c396573b.e8d738"]
        ]
    }, {
        "id": "154a1e0e.e80672",
        "type": "ui_button",
        "z": "6a097760.653918",
        "name": "",
        "group": "d251626d.10cec",
        "order": 3,
        "width": 0,
        "height": 0,
        "passthru": false,
        "label": "Motoroff",
        "tooltip": "",
        "color": "",
        "bgcolor": "",
        "icon": "",
        "payload": "{\"command\":\"motoroff\"}",
        "payloadType": "json",
        "topic": "",
```

```
    "x": 160,
    "y": 160,
    "wires": [
        ["f8fb8426.88b758", "c396573b.e8d738"]
    ]
}, {
    "id": "6329ceb0.9a74",
    "type": "ui_text",
    "z": "6a097760.653918",
    "group": "a9434212.30379",
    "order": 0,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Temperature",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 970,
    "y": 140,
    "wires": []
}, {
    "id": "5d4cb33b.861edc",
    "type": "ui_text",
    "z": "6a097760.653918",
    "group": "a9434212.30379",
    "order": 1,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Humidity",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 980,
    "y": 200,
    "wires": []
}, {
    "id": "d85fe3cc.9ca31",
    "type": "ui_text",
    "z": "6a097760.653918",
    "group": "a9434212.30379",
    "order": 0,
    "width": 0,
    "height": 0,
```

```
    "name": "",
    "label": "Region",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 980,
    "y": 260,
    "wires": []
}, {
    "id": "e00de3f6.29978",
    "type": "ui_text",
    "z": "6a097760.653918",
    "group": "a9434212.30379",
    "order": 3,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Weather Description",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 1020,
    "y": 320,
    "wires": []
}, {
    "id": "9888ac53.4a285",
    "type": "ui_chart",
    "z": "6a097760.653918",
    "name": "",
    "group": "309c8230.4f9bde",
    "order": 3,
    "width": 0,
    "height": 0,
    "label": "Soil moisture",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "",
    "ymax": "",
    "removeOlder": 1,
    "removeOlderPoints": "",
    "removeOlderUnit": "3600",
```

```
    "cutout": 0,
    "useOneColor": false,
    "useUTC": false,
    "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a",
"#d62728", "#ff9896", "#9467bd", "#c5b0d5"],
    "useOldStyle": false,
    "outputs": 1,
    "x": 820,
    "y": 460,
    "wires": [
        []
    ]
}, {
    "id": "a4f00796.520788",
    "type": "ui_chart",
    "z": "6a097760.653918",
    "name": "",
    "group": "309c8230.4f9bde",
    "order": 4,
    "width": 0,
    "height": 0,
    "label": "Humidity",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "",
    "ymax": "",
    "removeOlder": 1,
    "removeOlderPoints": "",
    "removeOlderUnit": "3600",
    "cutout": 0,
    "useOneColor": false,
    "useUTC": false,
    "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a",
"#d62728", "#ff9896", "#9467bd", "#c5b0d5"],
    "useOldStyle": false,
    "outputs": 1,
    "x": 800,
    "y": 580,
    "wires": [
```

```
        []
    ]
}, {
    "id": "687d6f13.98f7c",
    "type": "ui_chart",
    "z": "6a097760.653918",
    "name": "",
    "group": "309c8230.4f9bde",
    "order": 5,
    "width": 0,
    "height": 0,
    "label": "Temperature",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "",
    "ymax": "",
    "removeOlder": 1,
    "removeOlderPoints": "",
    "removeOlderUnit": "3600",
    "cutout": 0,
    "useOneColor": false,
    "useUTC": false,
    "colors": ["#1f77b4", "#aec7e8", "#ff7f0e", "#2ca02c", "#98df8a",
"#d62728", "#ff9896", "#9467bd", "#c5b0d5"],
    "useOldStyle": false,
    "outputs": 1,
    "x": 810,
    "y": 700,
    "wires": [
        []
    ]
}, {
    "id": "91b4e81a.972888",
    "type": "change",
    "z": "6a097760.653918",
    "name": "Temperature",
    "rules": [{
        "t": "set",
        "p": "payload",
```

```
            "pt": "msg",
            "to": "payload.main.temp",
            "tot": "msg"
        }],
        "action": "",
        "property": "",
        "from": "",
        "to": "",
        "reg": false,
        "x": 750,
        "y": 120,
        "wires": [
            ["6329ceb0.9a74"]
        ]
    }, {
        "id": "4bcf3c9.21fd4c4",
        "type": "change",
        "z": "6a097760.653918",
        "name": "Humidity",
        "rules": [{
            "t": "set",
            "p": "payload",
            "pt": "msg",
            "to": "payload.main.humidity",
            "tot": "msg"
        }],
        "action": "",
        "property": "",
        "from": "",
        "to": "",
        "reg": false,
        "x": 740,
        "y": 180,
        "wires": [
            ["5d4cb33b.861edc"]
        ]
    }, {
        "id": "2c496973.5626d6",
        "type": "change",
        "z": "6a097760.653918",
        "name": "Region",
        "rules": [{
            "t": "set",
```

```
            "p": "payload",
            "pt": "msg",
            "to": "payload.name",
            "tot": "msg"
        }],
        "action": "",
        "property": "",
        "from": "",
        "to": "",
        "reg": false,
        "x": 740,
        "y": 240,
        "wires": [
            ["d85fe3cc.9ca31"]
        ]
    }, {
        "id": "3552343c.1a23ac",
        "type": "change",
        "z": "6a097760.653918",
        "name": "Weather Description",
        "rules": [{
            "t": "set",
            "p": "payload",
            "pt": "msg",
            "to": "payload.weather.0.description",
            "tot": "msg"
        }],
        "action": "",
        "property": "",
        "from": "",
        "to": "",
        "reg": false,
        "x": 780,
        "y": 300,
        "wires": [
            ["e00de3f6.29978"]
        ]
    }, {
        "id": "233183d6.16ba7c",
        "type": "ibmiot",
        "z": "",
        "name": "",
        "keepalive": "60",
```

"serverName": "",
"cleansession": true,
"appId": "",
"shared": false
}, {
"id": "28e6141.0c047ec",
"type": "ui_group",
"z": "",
"name": "Smart Agriculture",
"tab": "d669ffca.1402d",
"order": 6,
"disp": true,
"width": "6",
"collapse": false
}, {
"id": "d251626d.10cec",
"type": "ui_group",
"z": "",
"name": "Motor Commands",
"tab": "d669ffca.1402d",
"order": 1,
"disp": true,
"width": "6",
"collapse": false
}, {
"id": "a9434212.30379",
"type": "ui_group",
"z": "",
"name": "Weather Forecast",
"tab": "d669ffca.1402d",
"order": 3,
"disp": true,
"width": "6",
"collapse": false
}, {
"id": "309c8230.4f9bde",
"type": "ui_group",
"z": "",
"name": "Graphical Representation",
"tab": "d669ffca.1402d",
"order": 5,
"disp": true,
"width": "6",

```
    "collapse": false
}, {
    "id": "d669ffca.1402d",
    "type": "ui_tab",
    "z": "",
    "name": "Smart Agriculture",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
}]
```

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-45136-1660728431

VIDEO DEMO LINK- https://youtu.be/NVrAd8u3D-U