IOT BASED SMART CROP PROTECTION FOR AGRICULTURE

TEAM ID : PNT2022TMID14113

TEAM MEMBERS:

1.M.GOKULAKANNAN
2.T.PARTHASARATHI
3.T.SUNILRAJ
4.R.S.SRIDHARAN

INTODUCTION

1.PROJECT OVERVIEW:

In this project, we have create a web application and mobile app for protect and monitor the crops by the farmers can monitor the corps from the animals and birds .

if animals and birds are detected by the IoT sensors , the image of the animals are capured and send the alert message to the farmers.

farmers also can monitor the soil moisture ,temperature and humidity by using the mobile app and also control the motors . click

2.PURPOSE:

An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop.

This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field.

The motors and sprinklers in the field can be controlled using the mobile application.

LITRATURE SURVEY

1.EXISTING PROBLEM:

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

2.REFERENCES:

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

[2] N.Penchalaiah, D.Pavithra, B.Bhargavi, D.P.Madhurai, K.Eliyas Shaik,S.Md.sohaib.Assitant Professor, Department of CSE,AITS, Rajampet,India UG Student,

Department of CSE,AITS,Rajampet, India.

[3] Mr.P.Venkateswara Rao, Mr.Ch Shiva Krishna ,MR M Samba Siva Reddy LBRCE,LBRCE,LBRCE.

[4] Mohit Korche,Sarthak Tokse, Shubham Shirbhate, Vaibhav Thakre, S. P. Jolhe(HOD). Students , Final Year,Dept.of Electrical engineering,Government College of engineering,Nagpur head of dept.,Electrical engineering,Government College of engineering,Nagpur

3.PROBLEM STATEMENT DEFINITION :

In the world economyof many Countrydependent upon the agriculture. In spite of economic development agriculture is the backbone of the economy. Crops in forms are many times ravaged by local animals like buffaloes, cows, goats, birds and fire etc. this leads to huge loss for the farmers.

it is not possible for farmers to blockade to entire fields or stay 24 hours and guard it. Agriculture meets food requirements of the people and produces several raw materialsfor industries. But because of animal interference and fire in agricultural lands, there will be huge loss of crops.Crops will be totally getting destroyed.

IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas

2. Ideation & Brainstorming

Project report

Template

Brainstorm & idea prioritization

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

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

Share template feedback

Before you collaborate

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

10 minutes

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

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

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

Open article →

Problem

1. Animals like wild boars, buffaloes, cows, elephant, monkeys, birds etc. damages the crop lot which results in loss of production and so of farmer. It is very difficult for farmer to keep an eye on the field every time. Therefore it is very important to monitor the nearby presence of animals. Our main aim to design a system that can help to farmer to protect his farm from, animals

2. This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field.

2

Brainstorm

M.GOKULAKANNAN

The speakers will be given the prerecorded audio input,which will give different kinds of sounds that can scare the bird and prevent the crop.it can also be accessed and turned off when not needed

the prpposed system uses a Raspberry pi board and the different sensors and cameras are interfaced with the puppet,As soon as the PIR sensors go High on detecting motion within a range of 10 meters,the camera will be turned ON which the first captures an image and then starts recording the video for about five to six minutes.

The crop field protection from the intrusion of animals using wireless technology which protect the crop from damage caused by animals and birds as well as divert the animal without any harm. the animal detection system is designed to detect the presnce to detect the presence of animals and offer a warning.

The APR board generate harsh sound to divert the path of ani8mlas.Due to this harsh sound animal will divert their path.The message and call will be given to the farm owner

P.PARTHASARATHI

The camera used in the ov7670 model which will take the picture of the intruder.once the movement is detected,the buzzers are activated.then the picture will be sent to the registered mobile number of the owner using GSm and the access is controlled

The picture which will be stored on boardas well as cloud. simultaneoustly a message will be genereayed automatically to the registered number using SIM900A modulle to inform abou the intrction along with the detail of the temperature and humidity obtained by interfacing the dht11 temperature and humidity sensor

It includes the various sensors connected to the ardunio board such as motion sensor,ultrasonic sensor,LDR,etc and other components are APR board,LcD,DC motor,etc. when any animal tries to enter the farm then the ultasonic sensor detects the presence of the animal and sends a signal to the ardunino board

In the message,there is a facility that,shows the direction of the intruded i.e.Animal is jus entered from the left side.also,if any animal tries to enter the farm then an electric fence is provided which will give a small amount of shock that will not cause any server harm to the animal.

T.SUNIL RAJ

once the sensor gets adapted to the surrounding,then any variation in the level of infrared radiations shall trigger thePIR sensor.

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.

then the agriculture sling starts to rotate to divert the path of birds.also there is a light facility at night time. when the resistance of LDR decreses then the flashlight will be turned on.due to the high intensity of light animals will not try to enter the farm at night time too.

the intension is only to divert the animals'path. similar when birds try to enter the farm then the motion sensor sense the presence of birds and gives a signal to the ardunio board

R.S.SRIDHARAN

The Ardunio uno also known as the ATmega 328p is an 8-bit RISC architeture microcontroller.Ardunio Uno is amain component of the system.other components which are important part of the system are PIR and ultrasonic sensors,a camera,GSM module,buzzers,and the speaker

the camera used in the system is the OV7670 model which will take the picture of the intruder

whereas if the motion detection is due to that of an unauthorized person without a valid RFID tag,the system further processes the image and video using Haar feature-based cascade classifiers for object detection,and decides if the entity is an animal or human intruder

A microcontroller (Ardunio UNO)is used for reading the inputs from PIR,a soil Moisture sensor .the GSM module is used for sending SMS to the farmer when movement is detected.

3

Group ideas

IOT SYSTEM

The PIR sensors go high on detecting motion within a range of 10 meters, the camera will be turned ON which first captures an image and then starts recording the video for about five to six minutes.

Once the sensor gets adapted to the surroundings,Then any variation in the level of infrared radiations shall trigger the PIR sensor. camera used in the system is the OV7670 model which take the picture pf the intruder

the proposed system uses a rasberry pi board and the different sensors and cameras are interface with the puppet.

DETECTION AND SOUND

The camera used in the system is the OV7670 model which will take the picture of the intruder. once the movement is detected ,the buzzers are activated.

The speaker will be given the prerecorded audio input,which will give different kinds of sounds continuously that can scare the birds and prevent the crops. it can also be accessed and turned off when not needed.

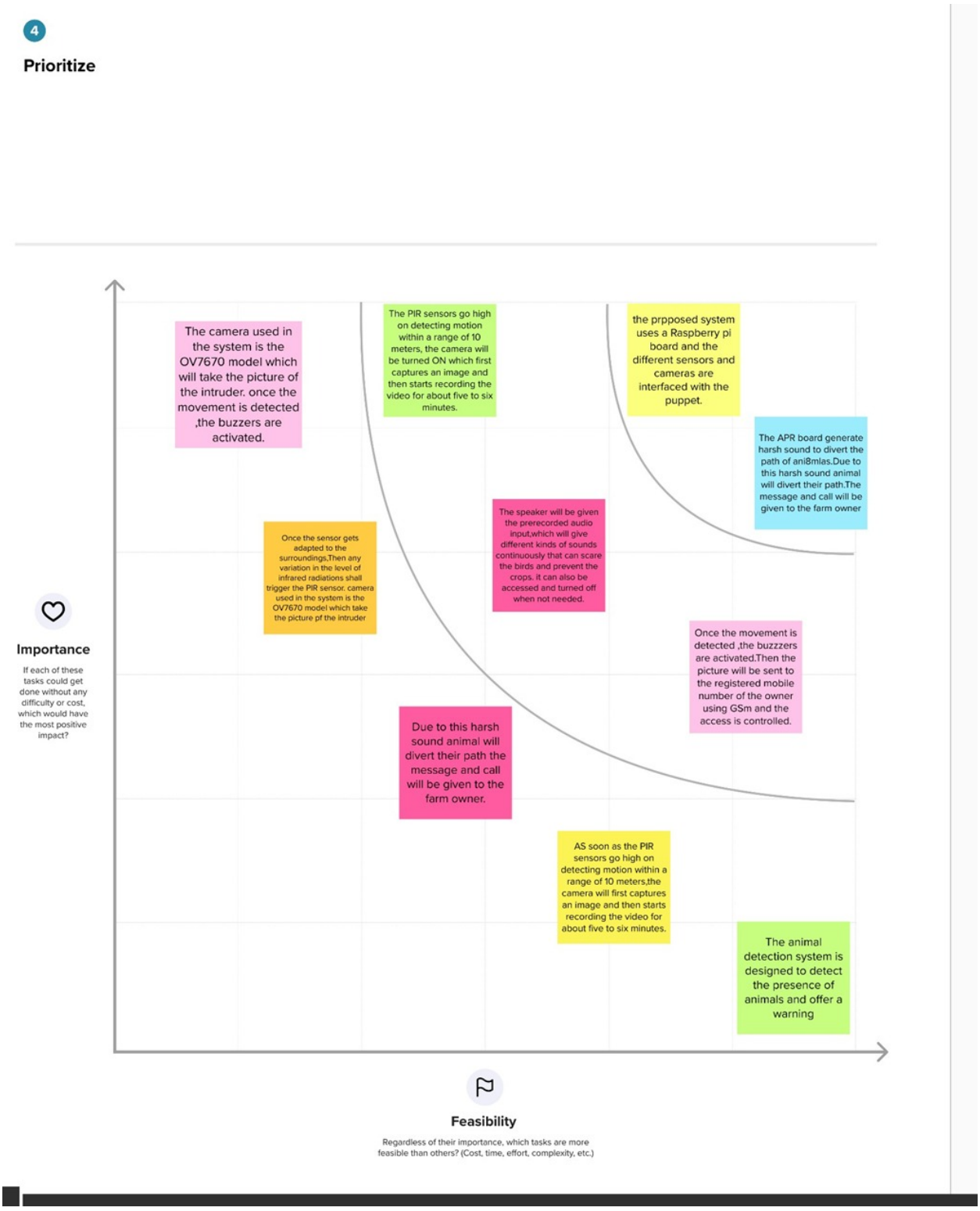
The APR board generate harsh sound to divert the path animals.Due to this harsh sound animal will sound animal will divert their path.

INTRUDER'S ALERTY

in the message ,there is a facility that,shows the direction of the intruded i.e.'animal is just entered from the left side.Also ,if any animal tries to enter the farm then an electric fence is provided which will give a small amount of shock that will not cause any severe harm to the animal

Due to this harsh sound animal will divert their path the message and call will be given to the farm owner.

Once the movement is detected ,the buzzzers are activated.Then the picture will be sent to the registered mobile number of the owner using GSm and the access is controlled.

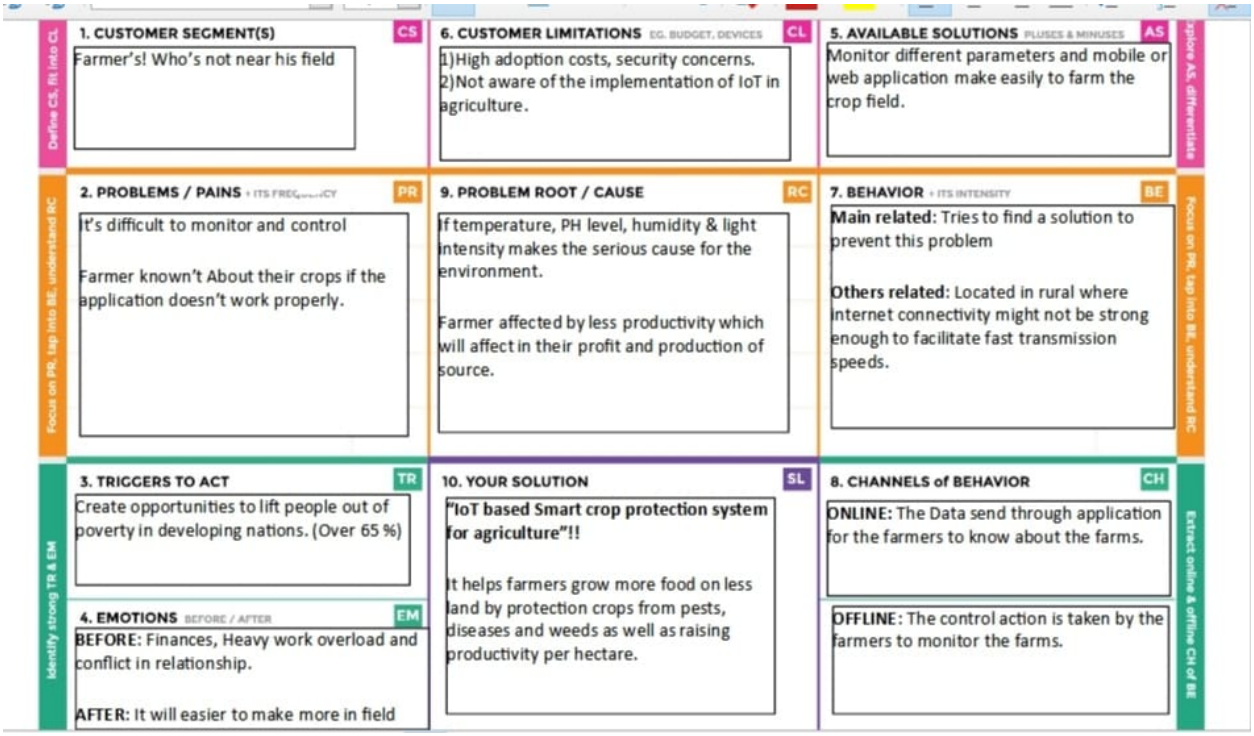

4
Prioritize
The camera used in the system is the OV7670 model which will take the picture of the intruder. once the movement is detected ,the buzzers are activated.
The PIR sensors go high on detecting motion within a range of 10 meters, the camera will be turned ON which first captures an image and then starts recording the video for about five to six minutes.
the prpposed system uses a Raspberry pi board and the different sensors and cameras are interfaced with the puppet.
The APR board generate harsh sound to divert the path of ani8mlas.Due to this harsh sound animal will divert their path.The message and call will be given to the farm owner
Once the sensor gets adapted to the surroundings,Then any variation in the level of infrared radiations shall trigger the PIR sensor. camera used in the system is the OV7670 model which take the picture pf the intruder
The speaker will be given the prerecorded audio input,which will give different kinds of sounds continuously that can scare the birds and prevent the crops. it can also be accessed and turned off when not needed.
Once the movement is detected ,the buzzzers are activated.Then the picture will be sent to the registered mobile number of the owner using GSm and the access is controlled.
Due to this harsh sound animal will divert their path the message and call will be given to the farm owner.
AS soon as the PIR sensors go high on detecting motion within a range of 10 meters,the camera will first captures an image and then starts recording the video for about five to six minutes.
The animal detection system is designed to detect the presence of animals and offer a warning
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.Proposed Solution:

s.no	Parameter	Description
1	Problem Statement (Problem to besolved)	● Crops are properly notirrigated due to in sufficient labourforces. ● Requires protecting crops from Wild animals attacks ,birds and pests. ● In environmental factors such as temperature climate,topography and soil quality which results in crop destruction because of crops against various.
2	Idea/Solution description	● Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF. ● The motor pump for managing the excess water level.It will be updated to authorities through IOT. ● IOT based fertilizing methods are followed, to minimize the negative effects on growth of crops while using fertilizers.
3	Novelty/Uniqueness	● Using the IoT Technology of Automated crop maintenance and protection of embedded system. ● The increasing demand fo rqulity foor.
4	Social Impact/Customer Satisfaction	● This proposed system provides many facilities which helps the farmers to maintain the crop field without much loss.
5	Business Model(RevenueModel)	● As the product usage can be understand by everyone,it is easy for them to use it properly for their saftest organisation
6	Scalability of the Solution	This prototype can be developed as product with minimum cost with high performance

		• **This can be developed to a scalableproduct by using sensors and transmitting the data through Wireless Sensor Network and Analysing the data in cloudand operation is performed using robots**

4. Problem Solution fit:



4. REQUIREMENT ANALYSIS

1. Functional requirement:

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

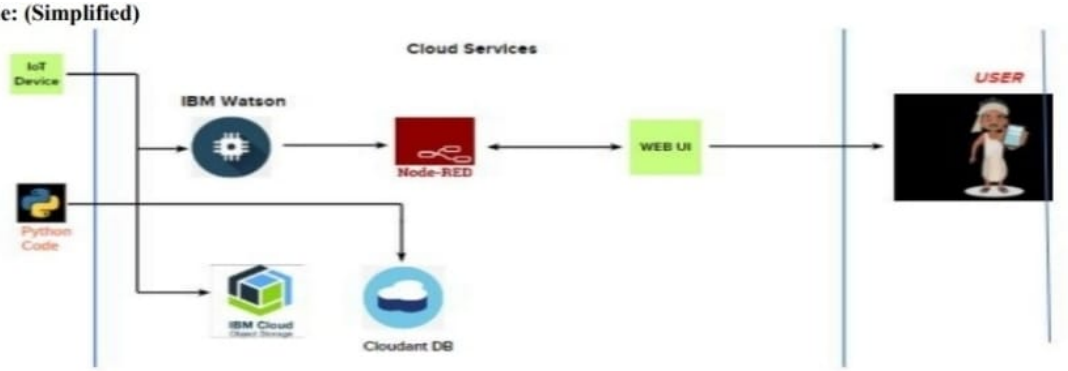
2.Non-Functional requirements:

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

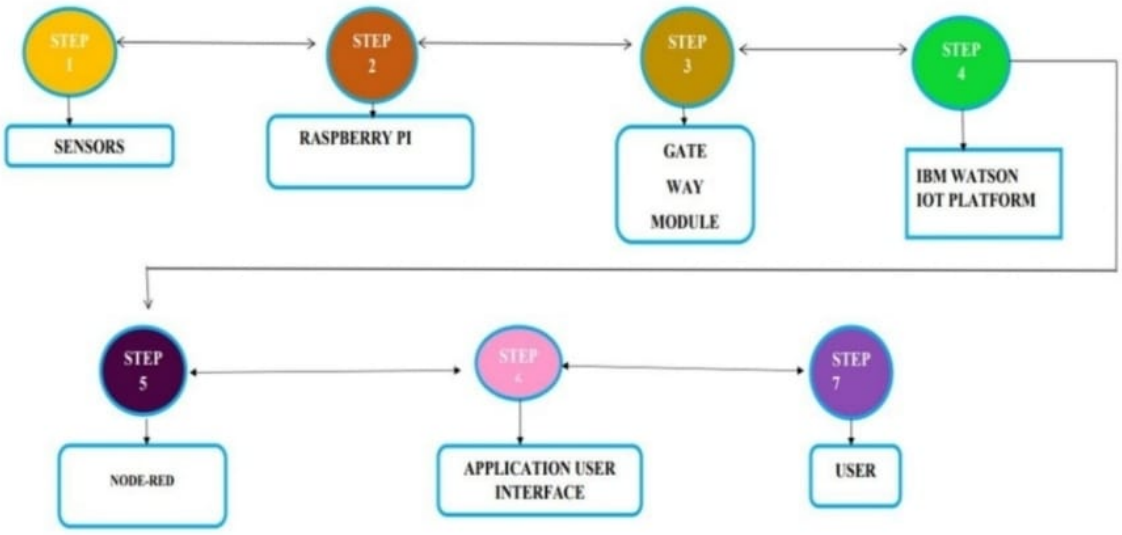
5. PROJECT DESIGN

1 Data Flow Diagrams:

Example: (Simplified)



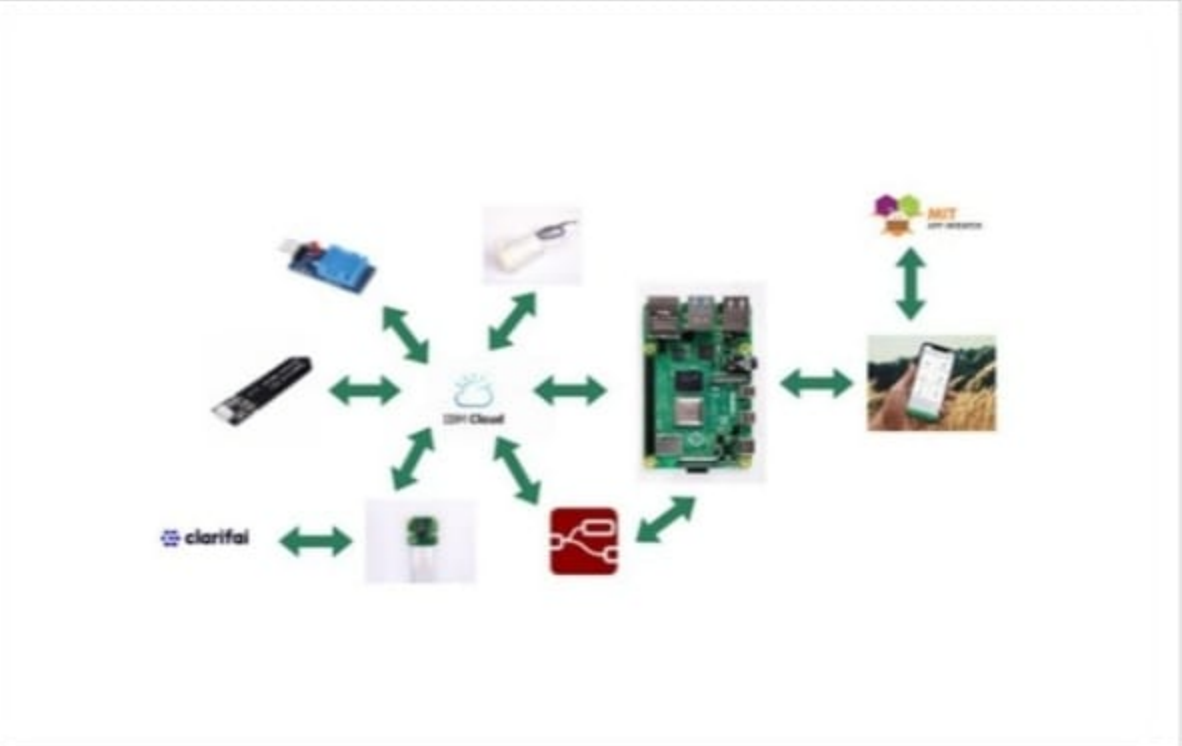
DATA FLOW GRAPH (DETAIL):



Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	User can register in application using their email id and password. Finally conform that their login	I can access my account / dashboard	High	Sprint-1
		USN-2	User can receive conformation mail after they can registered	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	User can also register through facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	User also can register using gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
Customer (Field Maintainer, Owner)	Problem solutions	USN-3	User can monitor the fields through remote mode	Checking Process.	Medium	Sprint-3
	Applications	USN-3	As a user, I can respond to the problems in the fields immediately.	Continuous monitoring and remedial actions.	Medium	Sprint-3
	Final process	USN-3	This proposed smart IOT based crop protection device is found to be cost effective and efficient.	I can take necessary action if required.	Medium	Sprint-4

2. Solution & Technical Architecture:



Sensors:

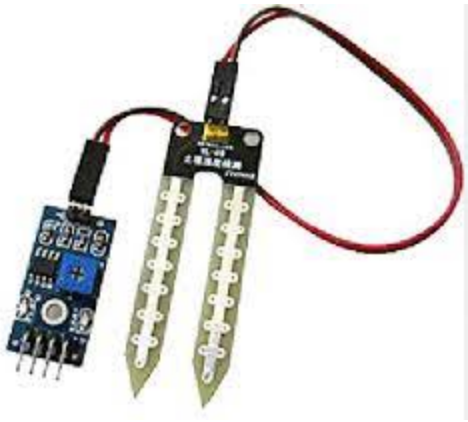
The soil moisture sensor senses the moisture level in the soil. The humidity and temperature sensor gives the humidity and temperature values of the atmosphere which determine whether the crop is suitable for growth. The soil moisture sensor, humidity and temperature sensor continuously monitors the soil and environmental conditions,sends the live data to mobile.

ARDUINO UNO:

Arduino Uno is the heart of the system. The facts gathered with the aid of the sensors are sent to the Arduino UNO. The gathered information may be displayed in a Arduino IDE

SOIL MOISTURE SENSOR:

A soil moisture sensor empowers agriculturalists to estimate the water levels without the need to be physically present in the field.



HUMIDITY SENSOR:

Humidity sensors are electronic devices that measure and report the moisture and air temperature of the surrounding environment.

1. The different soil parameters (temperature, humidity, Soil Moisture) are sensed using different sensors, and the obtained value is stored in the IBM cloud.

2. Arduino UNO is used as a processing unit that processes the data obtained from sensors and weather data from weather API.

3. Node-red is used as a programming tool to wire the hardware, software, and APIs. The MQTT protocol is followed for communication.

4. All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could make a decision through an app, whether to water the crop or not depending upon the sensor values. By using the app they can remotely operate the motor switch

NODE-RED:

Node-RED is a programming tool for wiring together hardware devices, APIs and online services. Primarily, it is a visual tool designed for the Internet of Things, but it can also be used for other applications to very quickly assemble flows of various services

3. User Stories:

SPRINT	FUNCTIONAL REQUIREMENT	USER STORY NUMBER	USER STORY/TASK	STORY POINTS	PRIORITY
Sprint-1		US-1	Create the IBM Cloud services which are being used in this project.	7	high
Sprint-1		US-2	Create the IBM Cloud services which are being used in this project.	7	high
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
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	6	high
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
Sprint-3		US-3	Create a Node-RED service	8	high
Sprint-3		US-2	Develop a python script to publish random	6	medium

6. PROJECT PLANNING & SCHEDULING

1.Sprint Schedule:

SPRINT	Functional Requirement (Epic)	User Story / Task	Story Points	Priority	Team members
Spirnt-1	registration	As a team leader, I can enrolled for the project by giving my email , password and within that I enter my team members name and their email .	2	high	gokulakannan
Spirnt-1		As a Team Leader, I will receive confirmation email once , I have enrolled for the project with team id and along with team members name.	2	high	gokulakannan
Spirnt-2		As a team member, I can login to the IBM portal by entering email & password .	1	medium	sridharan
Spirnt-2		As a team member, I can login to the IBM portal by entering email & password .	1	medium	parthasarathi
Spirnt-2		As a team member, I can login to the IBM portal by entering email & password .	1	medium	sunilraj

2.Sprint planning and estimation:

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	16 October 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	16 October2022
Brainstrom	List by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	16 October 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	18 October 2022
Problem Solution Fit	Prepare problem - solution fit document.	18 October 2022
Solution Architecture	Prepare solution architecture document.	19 October2022

Customer Journey	**Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).**	**27 OCTOBER 2022**
Functional Requirement	**Prepare the functional requirement document.**	**19 OCTOBER 2022**
Data Flow Diagrams	**Draw the data flow diagrams and submit for review.**	**27 OCTOBER 2022**
Technology Architecture	**Prepare the technology architecture diagram.**	**20 OCTOBER 2022**
Prepare Milestone & Activity List	**Prepare the milestones & activity list of the project.**	**30 OCTOBER 2022**
Project Development - Delivery of Sprint-1, 2, 3 & 4	**Develop & submit the developed code by testing it.**	**IN PROGRESS..**

7.CODING AND SOLUTIONING

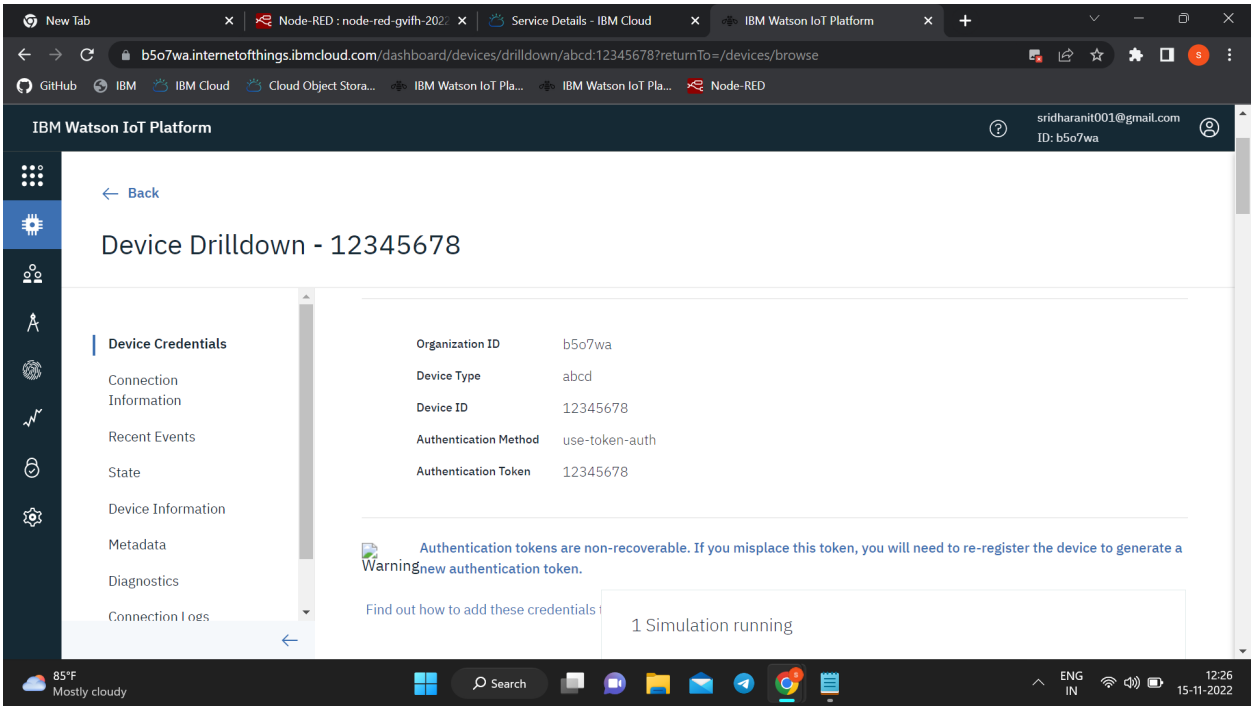
```
import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys
#IBM Watson Device Credentials.
organization = "b5o7wa"
deviceType = "1234"
deviceId = "12345678"
authMethod = "token"
authToken = ""14&t_V+snywrbm+vot"
def myCommandCallback(cmd):
 print("Command received: %s" % cmd.data['command'])
 status=cmd.data['command']
 if status=="sprinkler_on":
    print ("sprinkler is ON")
 else :
    print ("sprinkler is OFF")
 #print(cmd)


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

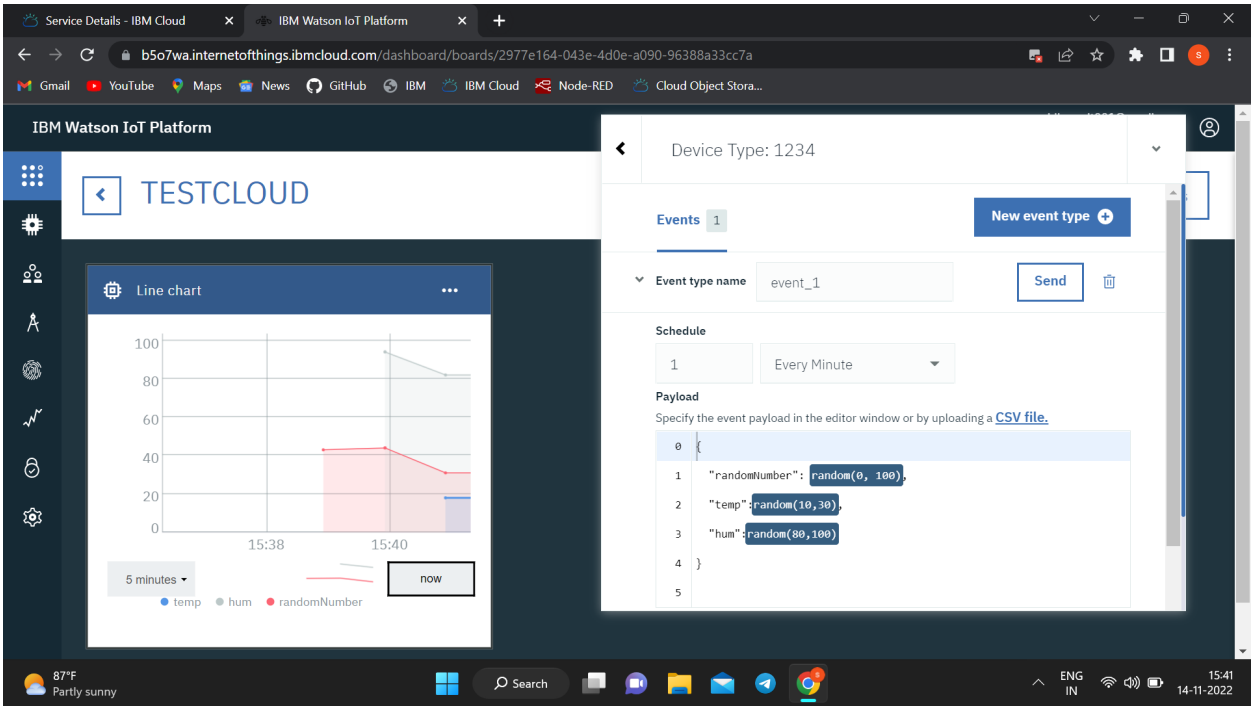
```
flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
flame_reading = random.choice(flame)
moist_level = round(random.uniform(0,100),2)
water_level = round(random.uniform(0,30),2)

#storing the sensor data to send in json format to cloud.

temp_data = { 'Temperature' : temp_sensor }
PH_data = { 'PH Level' : PH_sensor }
camera_data = { 'Animal attack' : camera_reading}
flame_data = { 'Flame' : flame_reading }
moist_data = { 'Moisture Level' : moist_level}
water_data = { 'Water Level' : water_level}

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

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

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

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

success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
```

```
 sleep(1)
 if success:
   print ("Published Water Level = %s cm" % water_level, "to IBM Watson")
 print ("")
 #Automation to control sprinklers by present temperature an to send alert message to IBM
Watson.

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

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

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

 #To send alert message to farmer that animal attack on crops.

 if (camera_reading == "Detected"):
   success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops detected"
}, qos=0)
 sleep(1)
 if success:
   print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM
```

```
Watson")
 print("")
 #To send alert message if flame detected on crop land and turn ON the splinkers to take
immediate action.

 if (flame_reading == "Detected"):
   print("sprinkler-2 is ON")
 success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in
danger,sprinklers turned ON" }, qos=0)
 sleep(1)
 if success:
   print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to
IBM Watson")

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

Project report





Features

Output: Digital pulse high (3V) when triggered (mo on detected) digital low when idle (no mo on detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor. Power supply: 5V-12V input voltage for most modules (they have a

3.3V regulator), but 5V is ideal in case the regulator has different specs.

BUZZER

Specifica ons

1. **RatedVoltage : 6V DC**
2. **Opera ng Voltage : 4 to 8V DC**
3. **Rated Current*: ≤30mA**
4. **SoundOutput at 10cm* : ≥85dB**
5. **Resonant Frequency : 2300 ±300Hz**
6. **Tone: Con nuous A buzzer is a loud noise maker.**

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

FEATURE-2:

a. **Goodsensi vity to Combus ble gas in wide range .**
b. **Highsensi vity to LPG, Propane and Hydrogen .**
c. **Longlife and low cost.**
d. **Simpledrive circuit.**

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%	

2 .USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the IoT Based smart crop protection system for agriculture project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	4	2	2	19
Duplicate	2	1	1	0	4
External	2	3	0	1	6
Fixed	11	2	2	20	35
Not	0	1	1	0	2
Reproduced					
Skipped	0	2	0	1	3
Won't Fix	0	5	2	1	8
Totals	26	18	8	25	77

1. 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	6	0	2	4
Client Application	45	0	3	42
Security	2	0	0	2
Outsource Shipping	3	0	0	3

Exception Reporting	10	0	1	9
Final Report Output	4	0	0	4
Version Control	3	0	1	2

RESULTS

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

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

ADVANTAGES AND DISADVANTAGES

Advantage:

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

Disadvantage:

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

CONCLUSION:

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

FUTURE SCOPE

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

will beactivated•

13.APPENDIX

1.SOURCE CODE:

```
# 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 Callback
 if 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 of type
"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 of type
"greeting" 10 times
deviceCli.connect()

while True:
temp=random.randint(0,100) pulse=random.randint(0,100)
soil=random.randint(0,100)

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

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)                         if not success:
print("Not connected to IoTF") time.sleep(1)

deviceCli.commandCallback = myCommandCallback

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



NODE RED FLOW

[
{
```

"id":"625574ead9839b34
",
"type":"ibmiotout",
"z":"630c8601c5ac3295",
"authentication":"apiKey",
"apiKey":"ef745d48e395ccc0",
"outputType":"cmd",
"deviceId":"b827ebd607b5",
"deviceType":"weather_monitor",
"eventCommandType":"data",
"format":"json",
"data":"data",
"qos":0,
"name":"IBM IoT",
"service":"registered",
"x":680,
"y":220,
"wires":[]
},
{
"id":"4cff18c3274cccc4","type":"ui_button",
"z":"630c8601c5ac3295",
"name":"",
"group":"716e956.00eed6c",
"order":2,
"width":"0",
"height":"0",
"passthru":false,
"label":"MotorON",
"tooltip":"",
"color":"",
"bgcolor":"",
"className":"",
"icon":"",
"payload":"{\"command\":\"motoron\"}",
"payloadType":"str",
"topic":"motoron",
"topicType":"s
tr","x":360,
"y":160
, "wires":[["625574ead9839b34"]]},

{
"id":"659589baceb4e0b0",
"type":"ui_button",
"z":"630c8601c5ac3295",
"name":"",
"group":"716e956.00eed6c",
"order":3,
"width":"0",
"height":"0",
"passthru":true,
"label":"MotorOFF",
"tooltip":"",
"color":"",
"bgcolor":"",
"className":"",
"icon":"",
"payload":"{\"command\":\"motoroff\"}",
"payloadType":"str",
"topic":"motoroff",
"topicType":"str",
"x":350,
"y":220,
"wires":[["625574ead9839b34"]]
},
{"id":"ef745d48e395ccc0",
"type":"ibmiot",
"name":"weather_monitor",
"keepalive":"60",
"serverName":"",
"cleansession":true,
"appId":"",
"shared":false},
{"id":"716e956.00eed6c",
"type":"ui_group",
"name":"Form",
"tab":"7e62365e.b7e6b8
","order":1,
"disp":true,
"width":"6",
"collapse":false},
{"id":"7e62365e.b7e6b8",

"type":"ui_tab",
"name":"contorl",
"icon":"dashboard
","order":1,
"disabled":false,
"hidden":false}
]
[
{
"id":"b42b5519fee73ee2",
"type":"ibmiotin",
"z":"03acb6ae05a0c712",
"authentication":"apiKey",
"apiKey":"ef745d48e395ccc0",
"inputType":"evt",
"logicalInterface":"",
"ruleId":"",
"deviceId":"b827ebd607b5",
"applicationId":"",
"deviceType":"weather_monitor",
"eventType":"+",
"commandType":"",
"format":"json",
"name":"IBMIoT",
"service":"registered",
"allDevices":"",
"allApplications":"",
"allDeviceTypes":"",
"allLogicalInterfaces":"",
"allEvents":true,
"allCommands":"",
"allFormats":"",
"qos":0,
"x":270,
"y":180,
"wires":[["50b13e02170d73fc","d7da6c2f5302ffaf","a949797028158f3f","a71f164bc3 78bcf1"]]
},
{
"id":"50b13e02170d73fc",
"type":"function",
"z":"03acb6ae05a0c712",

"name":"SoilMoisture",
"func":"msg.payload = msg.payload.soil;\nglobal.set('s',msg.payload);\nreturn msg;",
"outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":490,
"y":120,
"wires":[["a949797028158f3f","ba98e701f55f04fe"]]
},
{
"id":"d7da6c2f5302ffaf",
"type":"function",
"z":"03acb6ae05a0c712",
"name":"Humidity",
"func":"msg.payload = msg.payload.pulse;\nglobal.set('p',msg.payload)\nreturn msg;",
"outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":480,
"y":260,
"wires":[["a949797028158f3f","70a5b076eeb80b70"]]
},
{
"id":"a949797028158f3f",
"type":"debug",
"z":"03acb6ae05a0c712",
"name":"IBMo/p",
"active":true,
"tosidebar":true,
"console":false,
"tostatus":false,
"complete":"payload",
"targetType":"msg",
"statusVal":"",
"statusType":"auto",
"x":780,
"y":180,

"wires":[]
},
{
"id":"70a5b076eeb80b70",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",
"name":"",
"group":"f4cb8513b95c98a4",
"order":6,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Humidity",
"label":"Percentage(%)",
"format":"{{value}}",
"min":0,
"max":"100",
"colors":["#00b500","#e6e600","#ca3838"],
"seg1":"",
"seg2":"",
"className":"",
"x":860,
"y":260,
"wires":[]
},
{
"id":"a71f164bc378bcf1",
"type":"function",
"z":"03acb6ae05a0c712",
"name":"Temperature",
"func":"msg.payload=msg.payload.temp;\nglobal.set('t',msg.payload);\nreturnmsg;",
"outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":490,
"y":360,
"wires":[["8e8b63b110c5ec2d","a949797028158f3f"]]
},
{

"id":"8e8b63b110c5ec2d",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",
"name":"",
"group":"f4cb8513b95c98a4",
"order":11,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Temperature",
"label":"DegreeCelcius",
"format":"{{value}}",
"min":0,
"max":"100",
"colors":["#00b500","#e6e600","#ca3838"],
"seg1":"",
"seg2":"",
"className":"",
"x":790,
"y":360,
"wires":[]
},
{
"id":"ba98e701f55f04fe",
"type":"ui_gauge",
"z":"03acb6ae05a0c712",
"name":"",
"group":"f4cb8513b95c98a4",
"order":1,
"width":"0",
"height":"0",
"gtype":"gage",
"title":"Soil Moisture",
"label":"Percentage(%)",
"format":"{{value}}",
"min":0,
"max":"100",
"colors":["#00b500","#e6e600","#ca3838"],
"seg1":"",
"seg2":"",
"className":"",

"x":790,
"y":120,
"wires":[]
},
{
"id":"a259673baf5f0f98",
"type":"httpin",
"z":"03acb6ae05a0c712",
"name":"",
"url":"/sensor",
"method":"get",
"upload":false,
"swaggerDoc":"",
"x":370,
"y":500,
"wires":[["18a8cdbf7943d27a"]]
},
{
"id":"18a8cdbf7943d27a",
"type":"function",
"z":"03acb6ae05a0c712",
"name":"httpfunction",
"func":"msg.payload{\"pulse\":global.get('p'),\"temp\":global.get('t'),\"soil\":global.get(
's')};\nreturn
msg;",
"outputs":1,
"noerr":0,
"initialize":"",
"finalize":"",
"libs":[],
"x":630,
"y":500,
"wires":[["5c7996d53a445412"]]
},
{
"id":"5c7996d53a445412",
"type":"httpresponse",
"z":"03acb6ae05a0c712
","name":"",
"statusCode":"",
"headers":{},

"x":870,
"y":500,
"wires":[]
},
{
"id":"ef745d48e395ccc0",
"type":"ibmiot",
"name":"weather_monitor",
"keepalive":"60",
"serverName":"",
"cleansession":true,
"appId":"",
"shared":false},
{
"id":"f4cb8513b95c98a4",
"type":"ui_group",
"name":"monitor",
"tab":"1f4cb829.2fdee8",
"order":2,
"disp":true,
"width":"6",
"collapse":false,
"className":""
},
{
"id":"1f4cb829.2fdee8",
"type":"ui_tab",
"name":"Home",
"icon":"dashboard",
"order":3,
"disabled":false,
"hidden":false }

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-37341-1660304616

DEMO LINK:

https://youtu.be/b-NkRkZuNuo

Project report