PROJECT REPORT

On

“SMART CROP PROTECTION FOR AGRICULTURE”

Submitted

To

NALAIYA THIRAN

TEAM ID : PNT2022TMID26635

Submitted By

TEAM LEADER

JAGADESH.S

TEAM MEMBERS

KAMESH.J

BALAMURUGAN.U

CHARNDU.T

AJAY.C

CONTENTS

1. INTRODUCTION

1.1 PROJECT OVERVIEW

This paper aims at designing and executing the advanced development in embedded system for 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 **ESP32** 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.

1.2 PURPOSE

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. Animal detection system is designed to detect the presence of animal and offer a warning. In this project we used PIR and to detect the movement of the animal and send signal to the controller .It diverts the animal by producing sound by using Buzzer and the signal transmitted to GSM. Which can be accessed through the MIT Application by the Farmer.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

One of the major economic issues faced by the country is agriculture as this is the sector which is source of livelihood for about 54% of Indians till date. Still today this sector is not well developed and faces lots of problems resulting into low productivity of crops. As 43% of land in India, is used for farming but contributes only 18% of the nation's GDP. The poor condition of agriculture in the country is the point of concern for Indians. The rural farmers in India suffer from poverty and most of them are illiterate so there is lack of good extension services. The problem of wild life attack on crops i.e., crop Vandalization is becoming very common in the states of Tamil Nadu, Himachal Pradesh, Punjab, Haryana, Kerala and many other states. Wild animals like monkeys, elephants, wild pigs, deer, wild dogs, bison, nilgais, estray animals like cows and buffaloes and even birds like parakeets cause a lot of damage to crops by running over them, eating and completely vandalizing them. This lead to poor yield of crops and significant financial loss to the owners of the farmland. This problem is so pronounced that sometimes the farmers decide to leave the areas barren due to such frequent animal attacks Another major problem faced by Indian farmer is their dependency on nature and poorly maintained irrigation system. Current agricultural practice are neither economically nor environmentally sustainable and India's yields for many agricultural commodities are low.

2.2 REFERENCE

[1] ArturFrankiewicz; RafałCupek." Smart Passive Infrared Sensor - Hardware Plat- form "Year: 2013 IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society Pages: 7543 – 7547 .

[2] Discant, A. Rogozan, C. Rusu and A. Bensrhair, “Sensors for Obstacle Detection” 2007 30th International Spring Seminar 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.

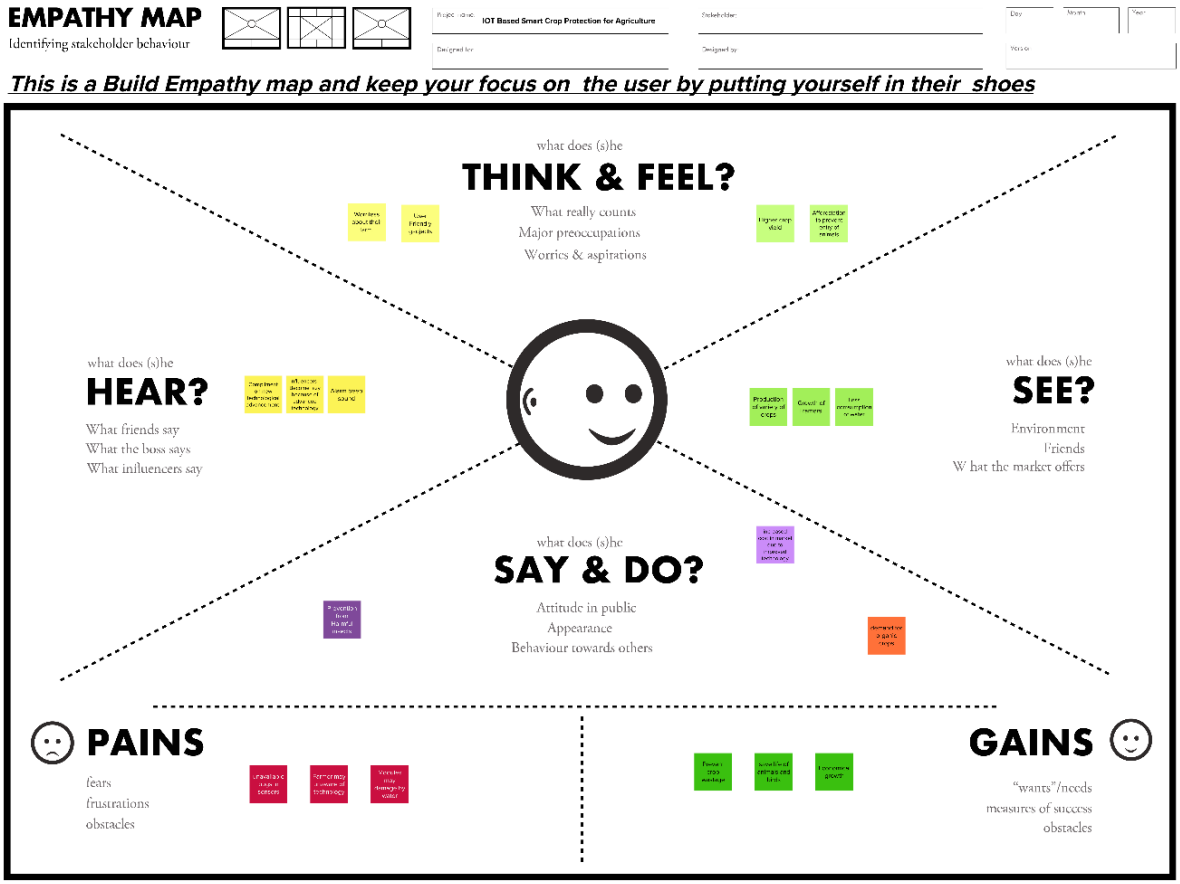
[3] Mustapha, Baharuddin, AladinZayegh, and Rezaul K. Begg. “Ultrasonic And Infrared Sensors Performance in A Wireless Obstacle Detection System” Artificial Intelligence, Modelling and Simulation (AIMS), 2013 1st International Conference on. IEEE, 2013.

2.3 PROBLEM STATEMENT DEFINITION

Crop monitoring is done where sensors are used to collect information from the agricultural field by our proposed work, PIR, Smoke sensor and GSM is used along with soil moisture sensor providing farmers more information about the water content present in the soil. When animals approach close to the PIR sensor, it detects the movement. After getting the initial input signal from the warm body, it is passed for further processing, then it will be passed on to the microcontroller. Then the system will be activated, immediately the buzzer goes on, and simultaneously it sends an SMS to the owner. Microcontroller (ESP32) is used for reading the inputs from PIR, Soil Moisture Sensor and Smoke sensor. The GSM module is used for sending SMS to farmer when movement or smoke is detected.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION & BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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.

Open article →

1 Define your problem statement

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

5 minutes

PROBLEM

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

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

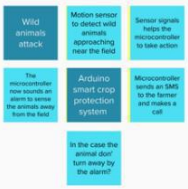
2 Brainstorm

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

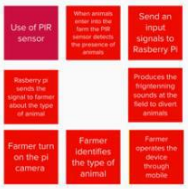
10 minutes

JAGADESH S

AJAY C



KAMESH J



BALAMURUGAN U

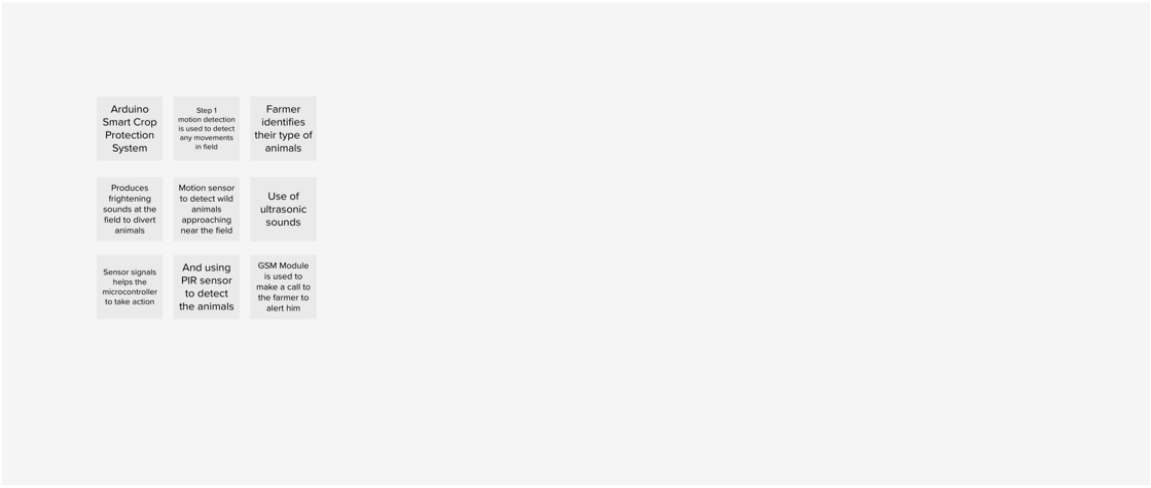
CHANDRU T

3

Group ideas

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

20 minutes

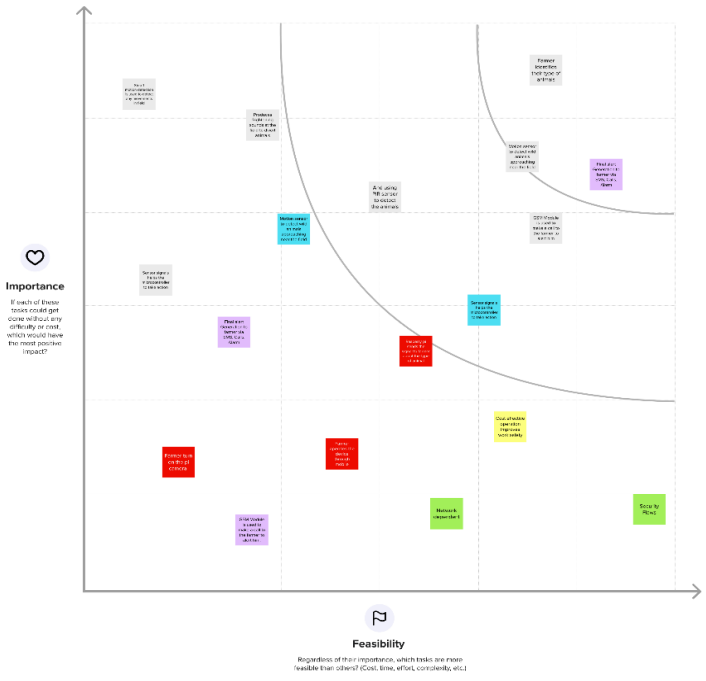


4

Prioritize

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

20 minutes



After you collaborate

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

Quick add-ons

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

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

Keep moving forward

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

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

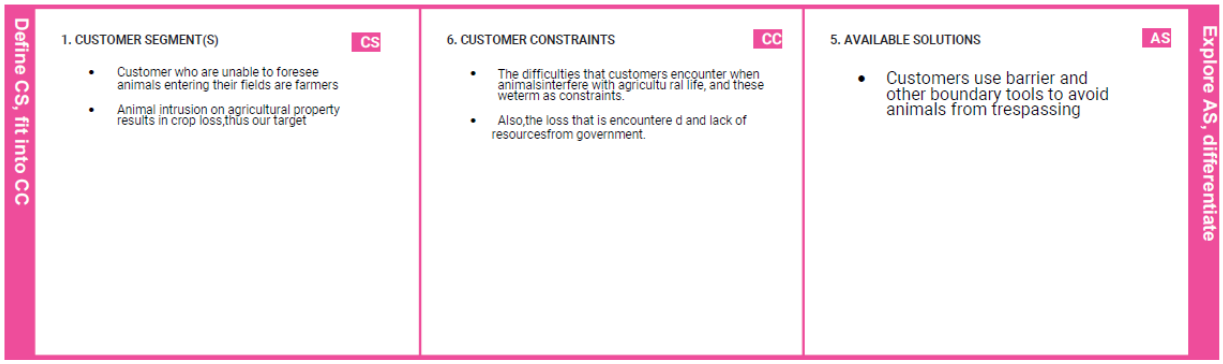
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

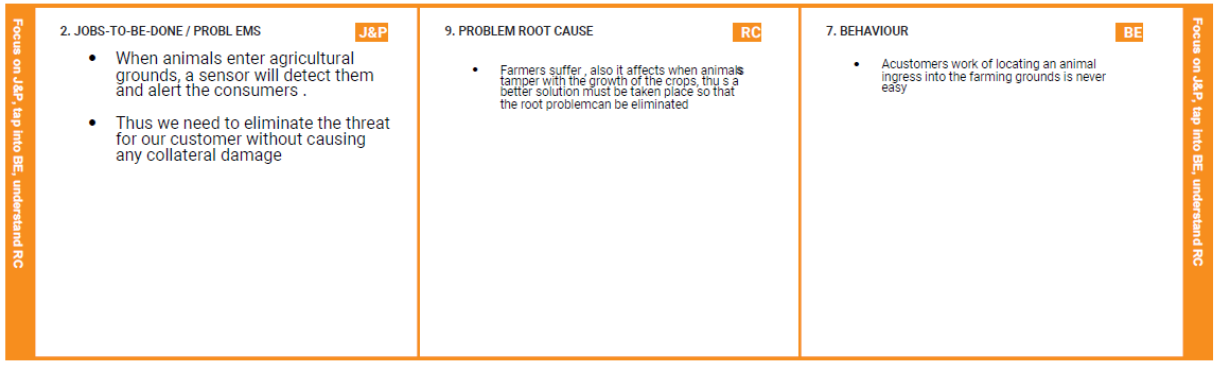
Share template feedback

3.3 PROPOSED SOLUTION

Our project is smart crop protection system Using ESP32. This project is helpful for the farmer to protect his farm from animals and unknown person near to his farm. We are use PIR sensors for sensing the movement at the boarder of farm and that data will be given to ESP32 after processing it can be display on lcd display. But we it is not sufficient to protect the farm hence we can add dog sounds via speaker so that the animals are not come inside the farm. We are interface nodemcu for message of alert. When any movement detect then we have a message on our register Android phone. This project is fully works on free energy i.e. solar energy is store at battery. The battery is connected to our system hence we don't require to give another power supply. We have added new feature to protect our farm by another issue. When the fire on our farm then we have received a fire message. So this is very protective and costly project. Hence because of our project the farmer can check the security and get immediate action.

3.4 PROBLEM SOLUTION FIT





4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

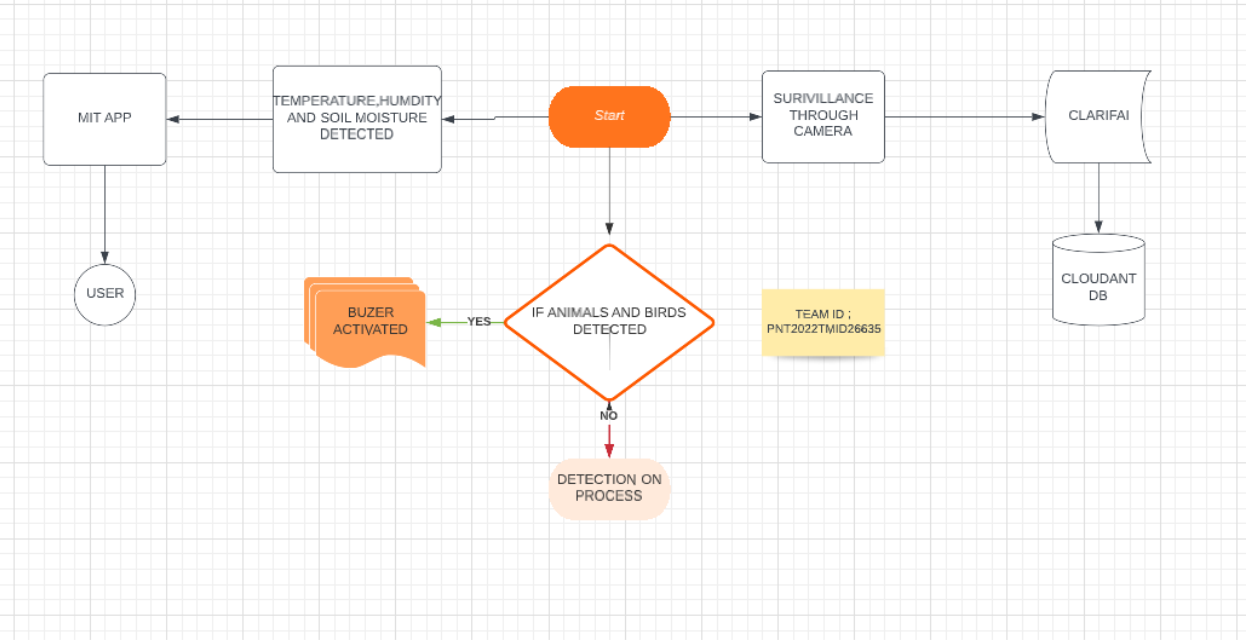
S.NO.	Functional Requirement.	Sub Requirement.
1.	User Visibility	Sense animals nearing the crop field & sounds alarm to woo them away as well as sends SMS to farmer using cloud service.
2.	User Reception	The Data like values of Temperature, Humidity, Soil moisture Sensors are received via SMS.
3.	User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

4.2 NON-FUNCTIONAL 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 production don'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.

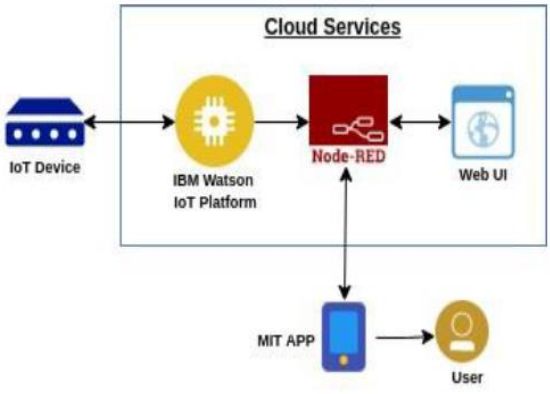
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



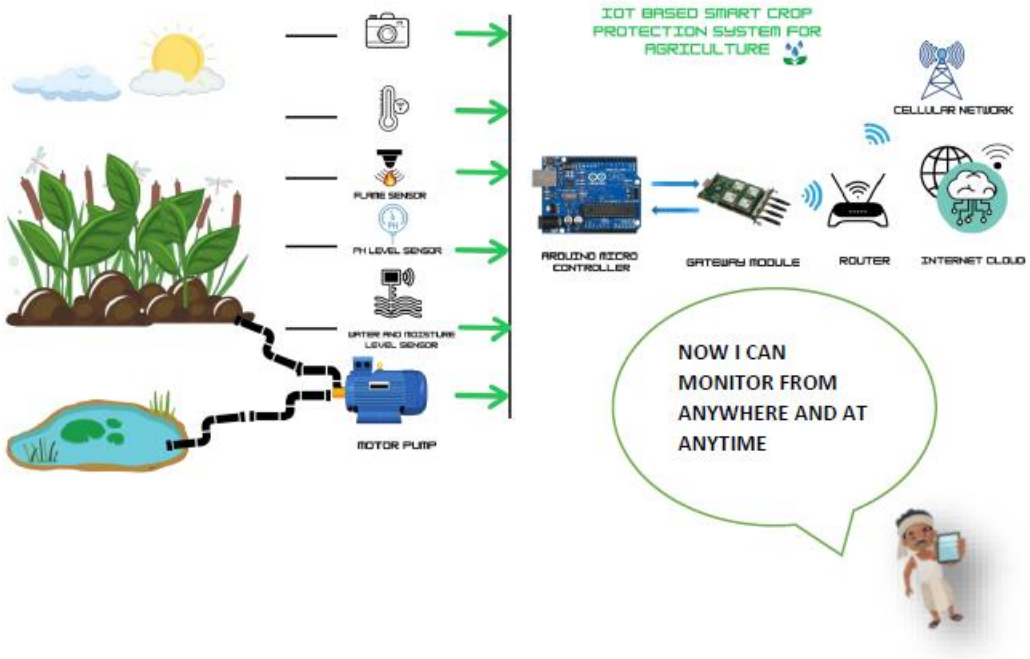
Guidelines:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)

Table : Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	*The internet of things system(IoT) refers to the set of devices and systems that systems that stay interconnected with real word sensors and actuators to the internet	Internet of things
2.	Security Implementations	*We can use sensors for detecting surroundings	Sensing technology
3.	Scalable Architecture	*it is clearly explained the IoT concept ,crop Damage issues and the need of using smart crop protection system	Internet of things
4.	Availability	*This system is developed using board programmed in embedded C and interfaced with sensing the surroundings	Microchip technology
5.	Performance	*The novelty of the work is that the system automatically alert the farmer by sending sms ,when animals enter into the fields	PIR sensor

5.3 SOLUTION ARCHITECTURE:
PROTECTION SYSTEM FOR AGRICULTURE



User Stories
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	As the user, he/she can register for the application by installing the application, signing up with the Google account or Gmail, then enter the password, conform the password, create profile, read Carefully, understand the guidelines clearly.	can access my account / dashboard	High	Sprint-1
Customer	Confirmation	USN-2	As a user, he/she will receive confirmation message via Email once they have registered for the application.	can receive confirmation email & click confirm	High	Sprint-1
Customer	Login	USN-3	As a user, if possible he/she can register for the application through Facebook.	can register & access the dashboard with Facebook Login	Low	Sprint-1
Administrator	Login page	USN-4	As a user name and password which is already existing.	Redirecting to user account	High	Sprint-1
Controlling the water pump	Controlling	USN-5	It used to control and field sprinklers	Switching on and off the motor pump manually via mobile application	High	Sprint-2
Customer (Web User)	Usage	USN-1	As user, he/she can access the website and obtain the information.	Can see the information that is provided by the device	High	Sprint-2
	Working	USN-2	As user, he/she will receive an alert notification given by the device and can access the application for motor control, field sprinklers.	Can access the application whenever needed	High	Sprint-2
Customer Care Executive	Action	USN-1	As user when any issues arisen by the customer can resolve the problem.	Can solve the issues arisen by the provided application and device	High	Sprint-3
Administrator	Administration	USN-1	User stores every information needed to develop the administration.	Can store the gained information	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

Use the below template to create product backlog and sprint 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	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T
Sprint-1		US-2	Configure the IBM Cloud services which are being used in completing this project.	4	Medium	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T
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	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
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	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T
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	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T
Sprint-3		US-2	Create a Node-RED service.	10	High	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T
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	JAGADESH S KAMESH J AJAY C BALAMURGAN U CHANDRU T
Sprint-3		US-2	After developing python code, commands are received just print the statements which represent the control of the devices.	5	Medium	JAGADESH S KAMESH J AJAY C

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
						BALAMURGANU CHANDRU T
Sprint-4		US-3	Publish Data to The IBM Cloud	8	High	JAGADESH S KAMESH J AJAY C BALAMURGANU CHANDRU T
Sprint-4		US-1	Create Web UI in Node-Red	10	High	JAGADESH S KAMESH J AJAY C BALAMURGANU CHANDRU T
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	JAGADESH S KAMESH J AJAY C BALAMURGANU CHANDRU T

Project Tracker, Velocity & Burndown Chart: (4 marks)

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

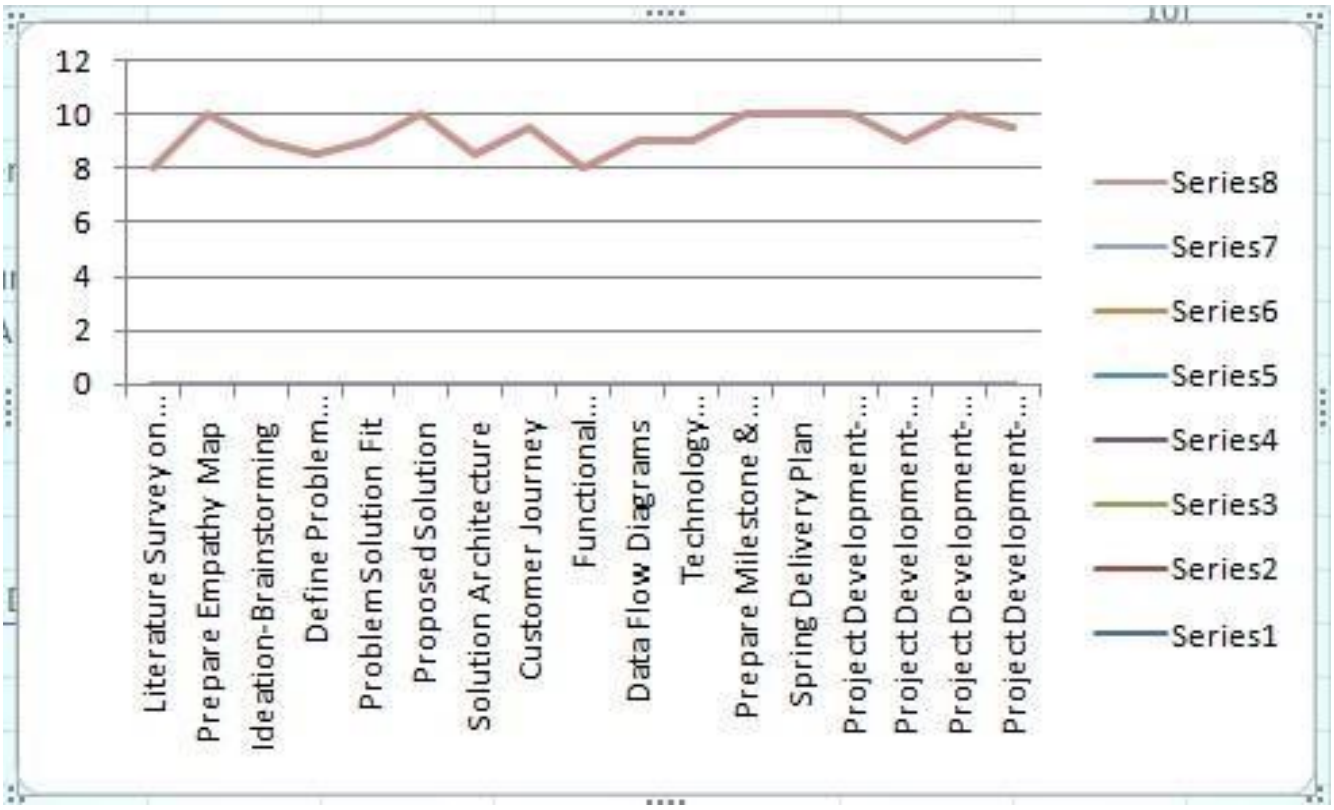
VELOCITY:

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)

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

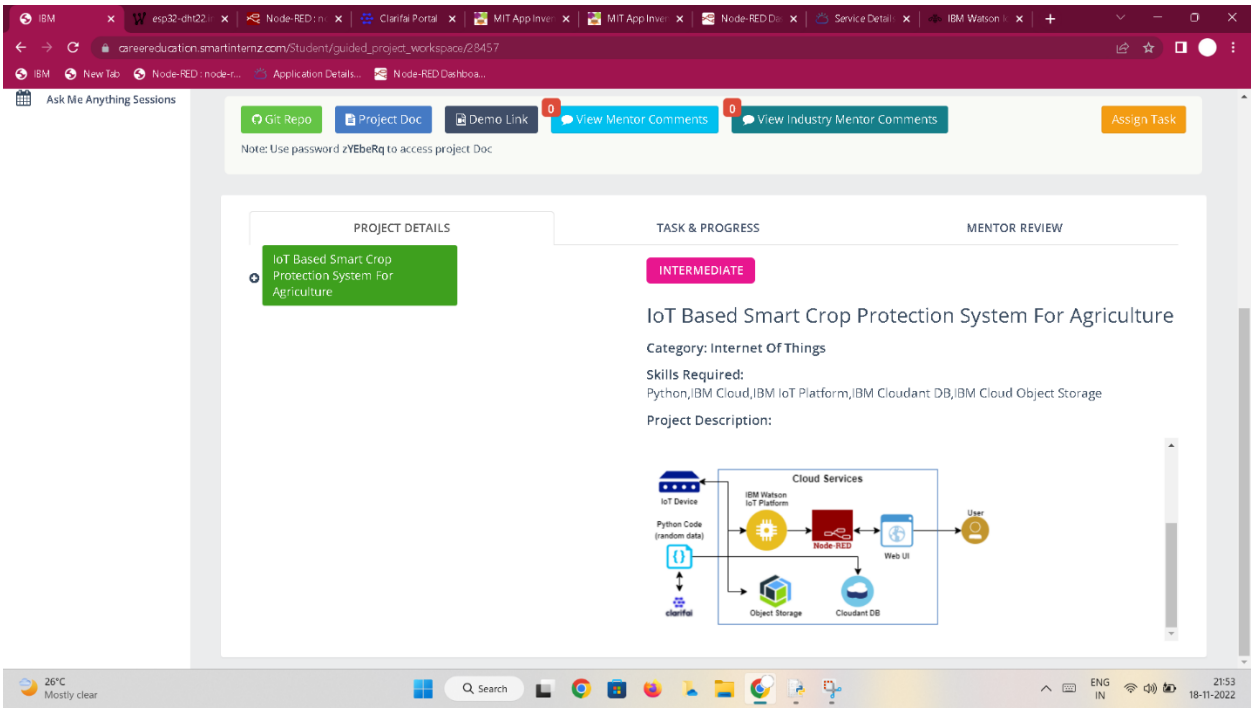
BURNDOWN CHART:

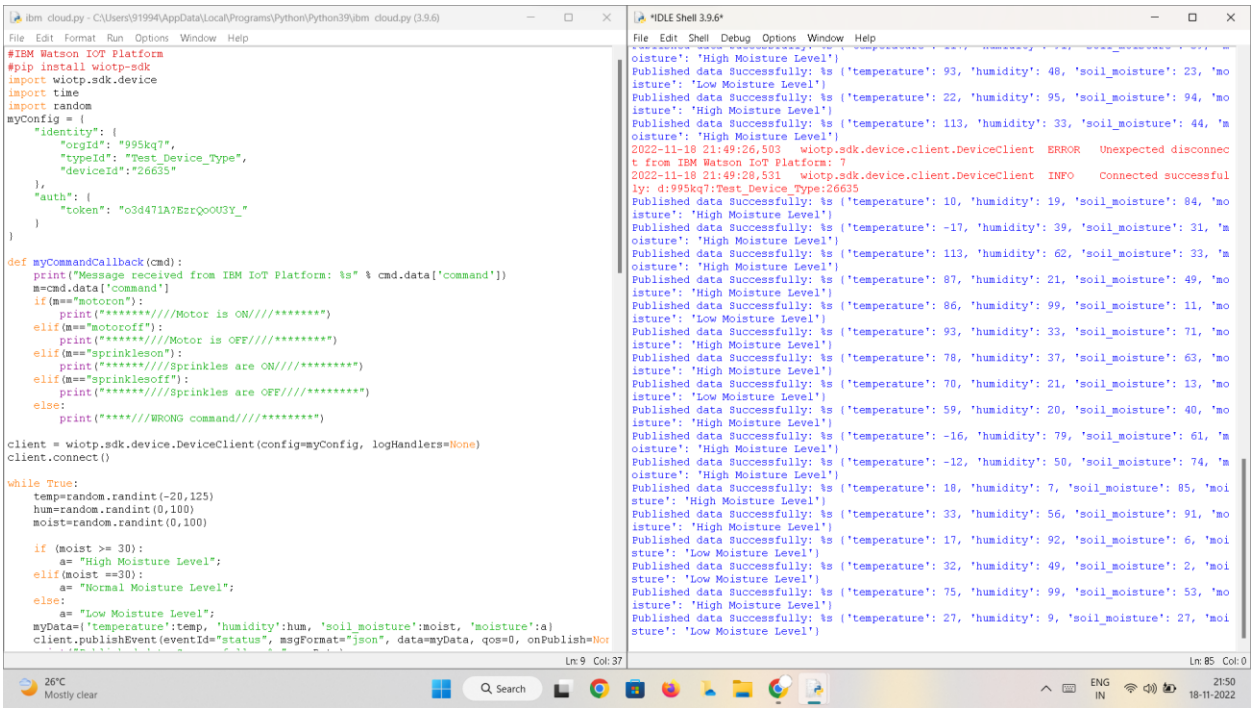
A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burndown charts can be applied to any project containing measurable progress overtime.

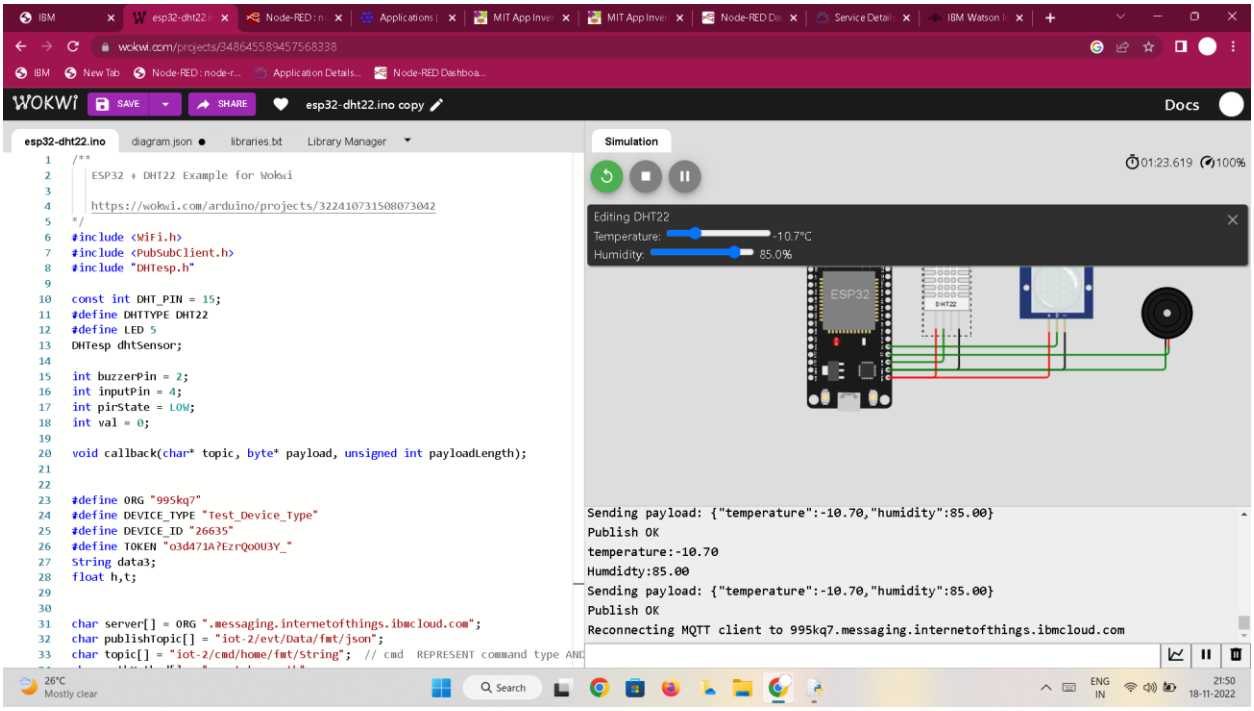


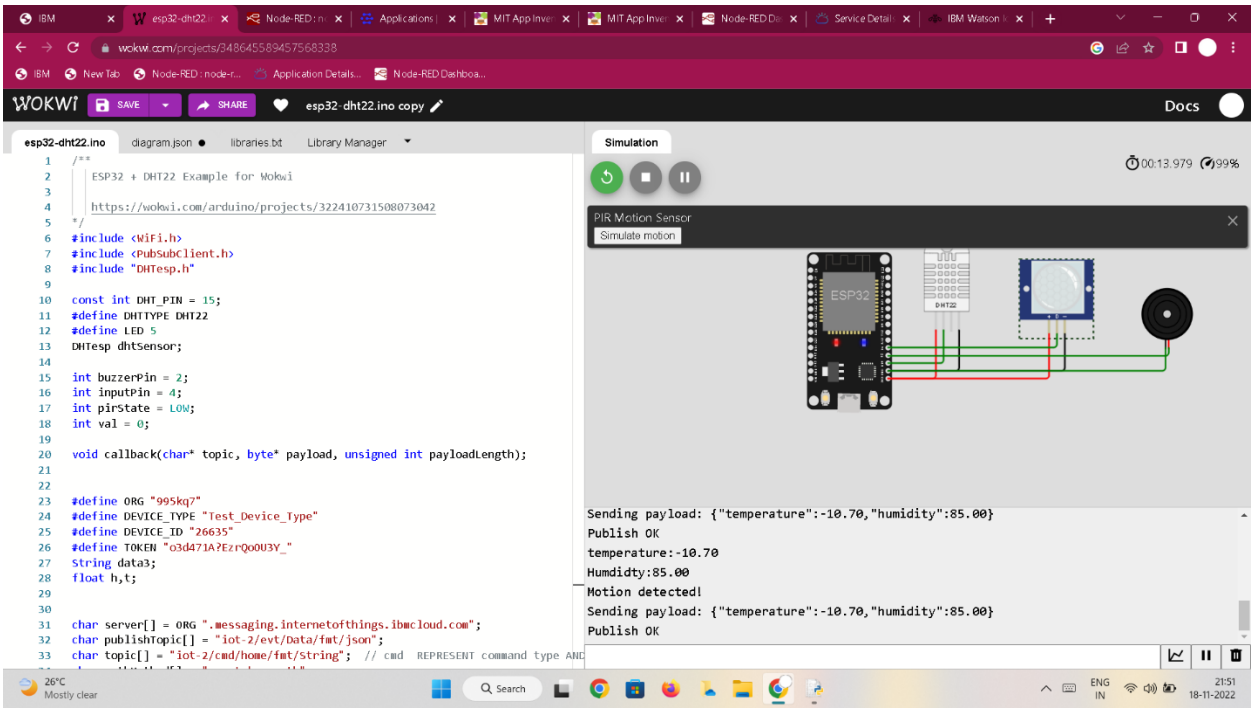
7. CODING & SOLUTIONING

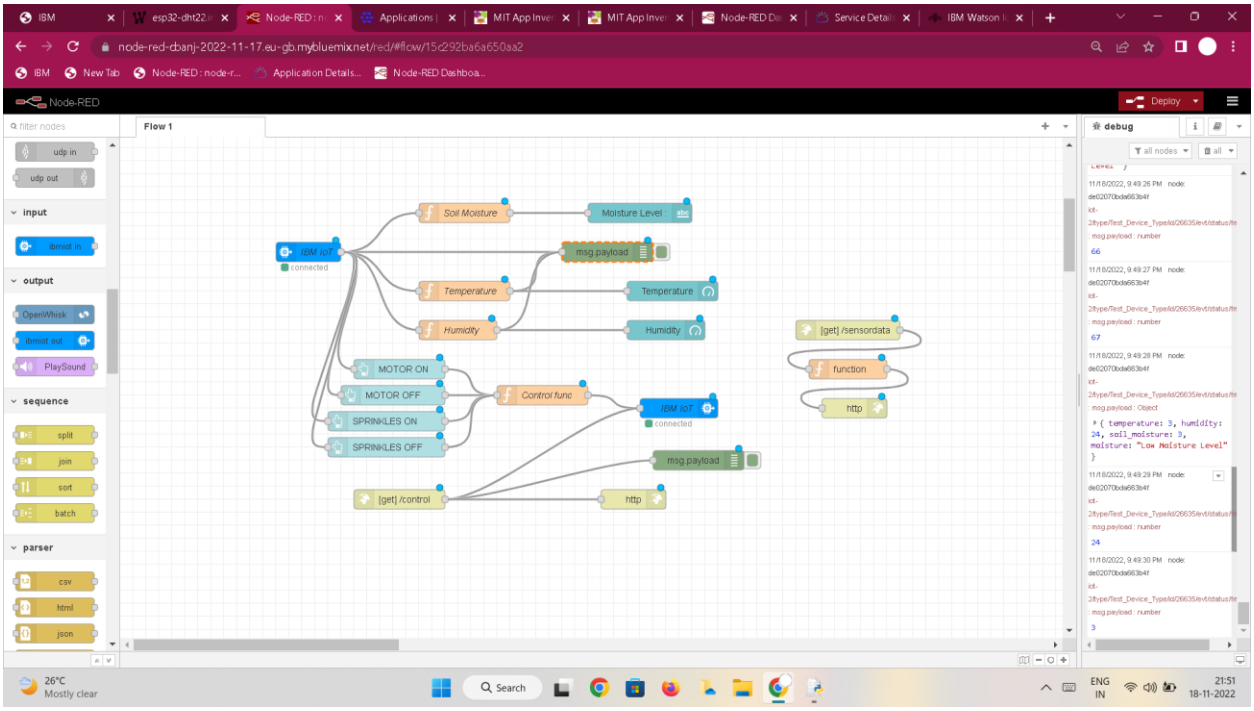
7.1 FEATURE 1

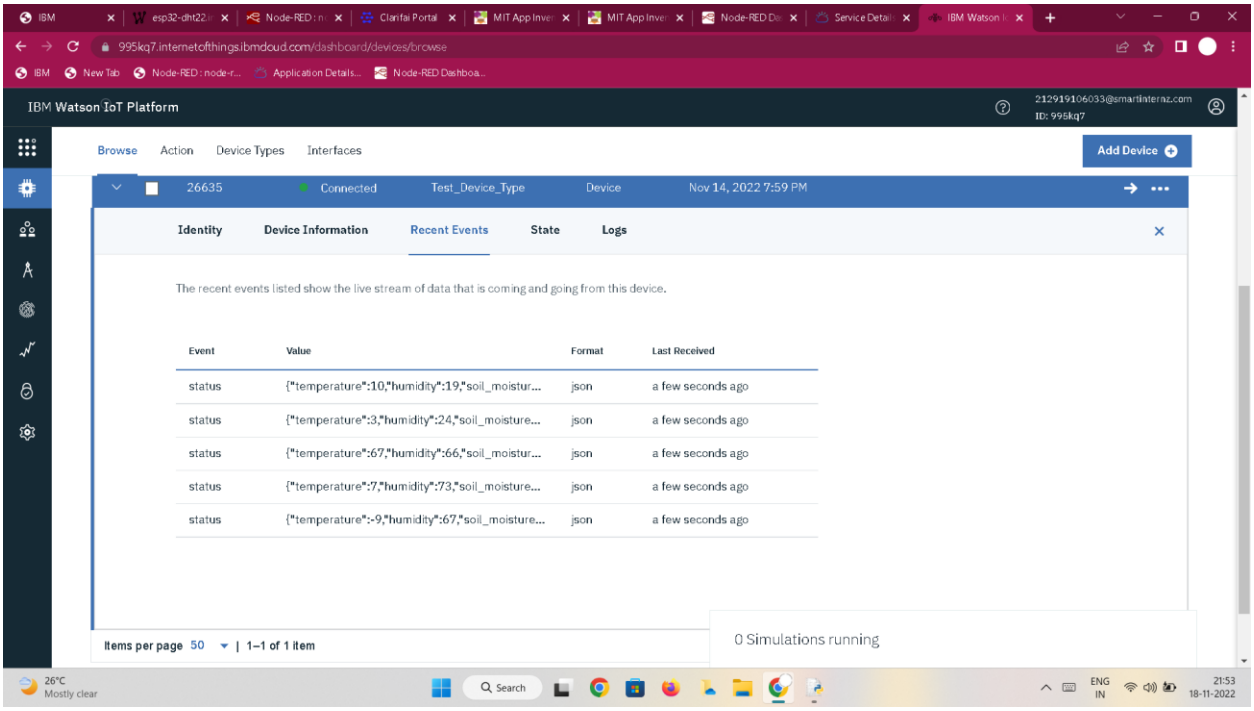


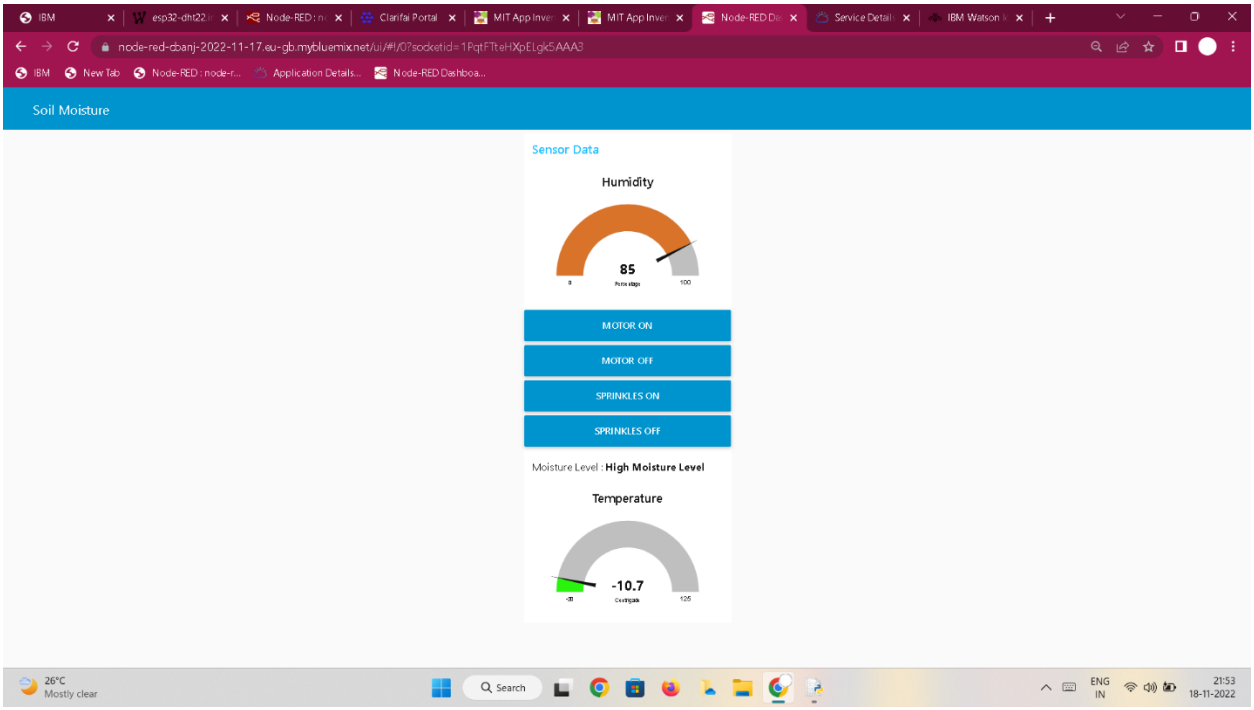


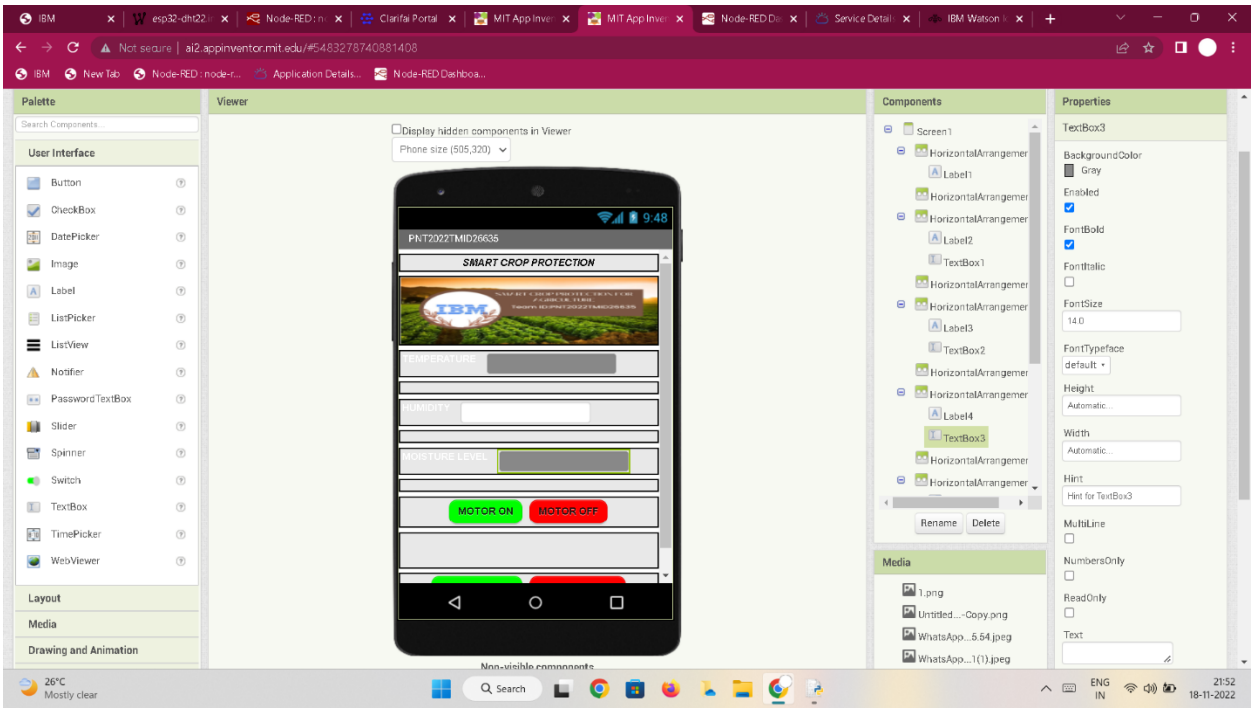

wokwi.com/projects/348645589457568338
WOKWI
SAVE
SHARE
esp32-dht22.ino copy
Docs
esp32-dht22.ino
diagram.json
libraries.txt
Library Manager
Simulation
01:23.619
100%
Editing DHT22
Temperature: -10.7°C
Humidity: 85.0%
ESP32
DHT22
Sending payload: {"temperature":-10.70,"humidity":85.00}
Publish OK
temperature:-10.70
Humdidty:85.00
Sending payload: {"temperature":-10.70,"humidity":85.00}
Publish OK
Reconnecting MQTT client to 995kq7.messaging.internetofthings.ibmcloud.com
26°C
Mostly clear
Search
ENG
IN
21:50
18-11-2022

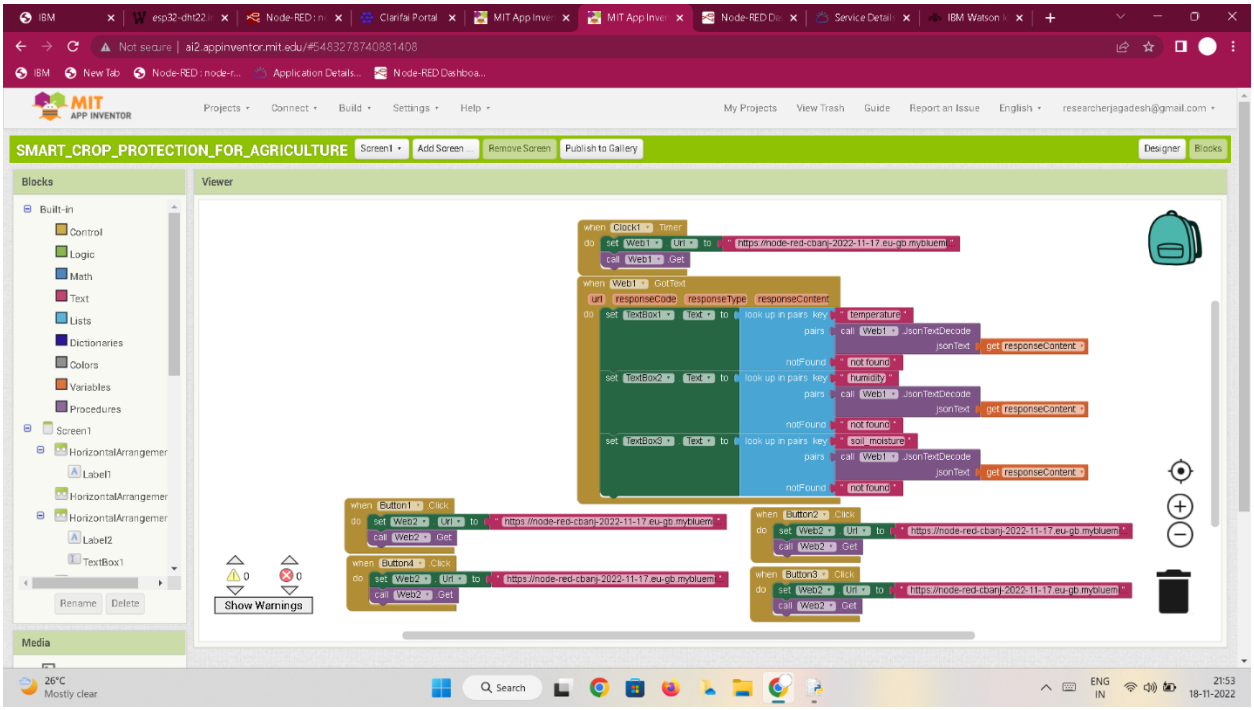

wokwi.com/projects/348645589457568338
WOKWI
SAVE
SHARE
esp32-dht22.ino copy
Docs
esp32-dht22.ino
diagram.json
libraries.txt
Library Manager
Simulation
00:13.979
99%
PIR Motion Sensor
Simulate mobon
ESP32
DHT22
Sending payload: {"temperature":-10.70,"humidity":85.00}
Publish OK
temperature:-10.70
Humdidty:85.00
Motion detected!
Sending payload: {"temperature":-10.70,"humidity":85.00}
Publish OK
26°C
Mostly clear
Search
ENG
IN
21:51
18-11-2022


Node-RED
Deploy
Flow 1
Soil Moisture
Moisture Level :
IBM IoT
connected
msg.payload
Temperature
Temperature
Humidity
Humidity
[get] /sensordata
function
http
MOTOR ON
MOTOR OFF
SPRINKLES ON
SPRINKLES OFF
Control func
IBM IoT
connected
msg.payload
[get] /control
http
debug
udp in
udp out
input
ibmiot in
output
OpenWhisk
ibmiot out
PlaySound
sequence
split
join
sort
batch
parser
csv
html
json

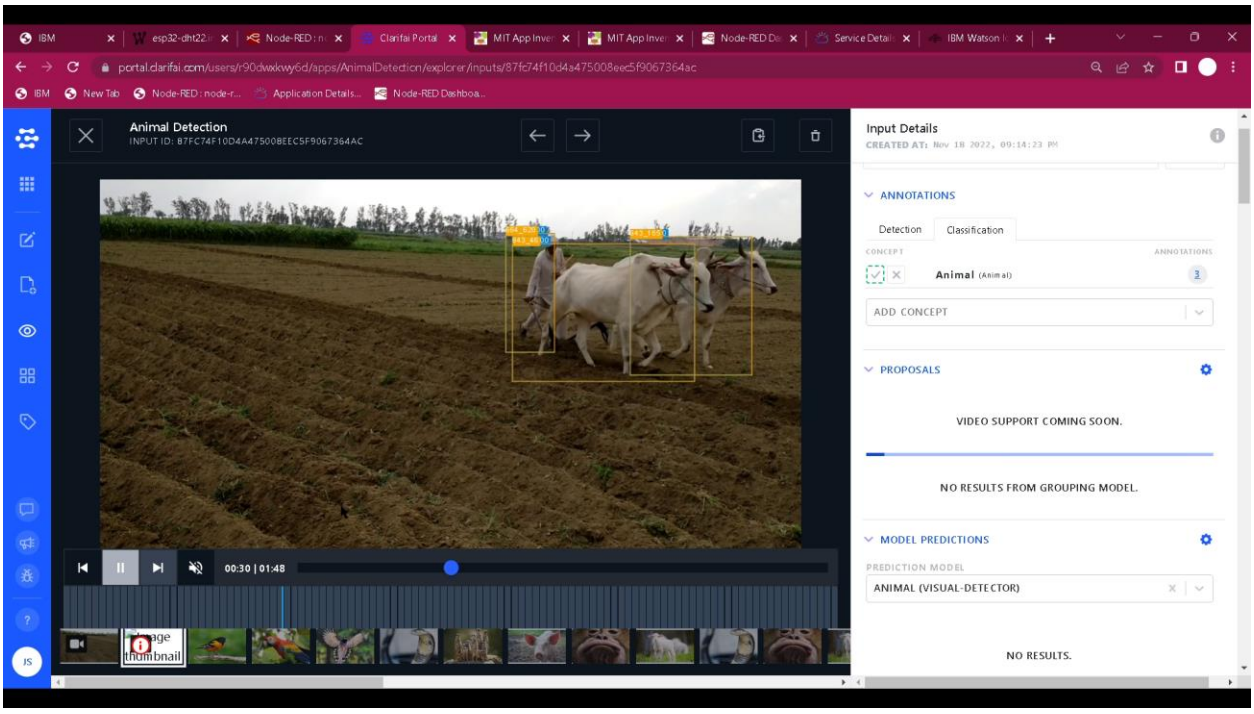

IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
26635
Connected
Test_Device_Type
Device
Nov 14, 2022 7:59 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
status
{"temperature":10,"humidity":19,"soil_moistur...
json
a few seconds ago
status
{"temperature":3,"humidity":24,"soil_moisture...
json
a few seconds ago
status
{"temperature":67,"humidity":66,"soil_moistur...
json
a few seconds ago
status
{"temperature":7,"humidity":73,"soil_moisture...
json
a few seconds ago
status
{"temperature":-9,"humidity":67,"soil_moisture...
json
a few seconds ago
Items per page 50
1–1 of 1 item
0 Simulations running


Soil Moisture
Sensor Data
Humidity
85
MOTOR ON
MOTOR OFF
SPRINKLES ON
SPRINKLES OFF
Moisture Level : High Moisture Level
Temperature
-10.7


Palette
Viewer
Components
Properties
PNT2022TMID26635
SMART CROP PROTECTION
TEMPERATURE
HUMIDITY
MOISTURE LEVEL
MOTOR ON
MOTOR OFF


SMART_CROP_PROTECTION_FOR_AGRICULTURE
Screen1
Add Screen
Remove Screen
Publish to Gallery
Designer
Blocks
Viewer
Show Warnings

7.2 FEATURE 2


Animal Detection
INPUT ID: 87FC74F10D4A47500BEEC5F9067364AC
Input Details
ANNOTATIONS
Detection
Classification
Animal (Animal)
ADD CONCEPT
PROPOSALS
VIDEO SUPPORT COMING SOON.
NO RESULTS FROM GROUPING MODEL.
MODEL PREDICTIONS
ANIMAL (VISUAL-DETECTOR)
NO RESULTS.

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	TC for Automation(Y/N)	BUG ID	Executed By
TC_001	Functional	Home Page	Verify user is able to see the values of humidity, temerature and moisture level	IBM cloud, Python IDLE,Node-Red,MIT App	1.Navigate to the temperature,humidity and soil moisture level in UI 2.User should see the measurement changes whenever they sense moisture,humidity and temperature.	ESP8266, DHT22	Desired output	Working as expected	Pass	Executed successfully	Yes	Nil	User
TC-002	Functional	Home Page	Verify user is buzzer make noise when motionis detected.	IBM cloud, Python IDLE,Node-Red,MIIT App	1.Navigate to the animals and birds. 2.Buzzer to thearten them.	ESP8266,PIR Sensor , Buzzer	Desired output	Working as expected	Pass	Executed successfully	Yes	Nil	User
TC-003	UI	Home Page	Verify whether the expected measurement sections are present	IBM cloud, Python IDLE,Node-Red,MIT App	1.Navigate to the temperature,humidity and soil moisture level in UI 2.User should see the measurement changes whenever they sense moisture,humidity and temperature. 3.Buzzer alert when the motion detected by PIR Sensor.	Arduino board, ESP8266, Smoke sensor	Desired output	Working as expected	Pass	Executed successfully	Yes	Nil	User
TC-004	UI	Home Page	Verify whether the expected measurement sections are present and with default values	IBM cloud, Python IDLE,Node-Red,MIT App	1.Navigate to the temperature,humidity and soil moisture level in UI 2.User should see the measurement changes whenever they sense moisture,humidity and temperature. 3.Buzzer alert when the motion detected by PIR Sensor.	ESP8266,DHT22,PIR Sensor, Buzzer	Desired output.	Working as expected	Pass	Executed successfully	Yes	Nil	User
TC-005	Functional	Home Page	Verify user is able to see the values when the motor and sprinkles status are showing up	IBM cloud, Python IDLE,Node-Red,MIT App	1.Navigate to MIT App 2.Press the buttons motor on, motor off 3.The Status of the Motor aon and off is shown up in output	MIT app, Node red	Desired output.	Not Working as expected	Pass	Executed sucessfully	Yes	Nil	User
TC-006	Functional	Home Page	Verify the DHT22 sensor detect the temperature and humidity	IBM cloud, Python IDLE,Node-Red,MIT App	1.Navigate to the temperature,humidity and soil moisture level in UI 2.User should see the measurement changes whenever they sense moisture,humidity and temperature. 3.Buzzer alert when the motion detected by PIR Sensor.	ESP8266,DHT22,PIR Sensor, Buzzer	Desired output	Not Working as expected	Pass	Executed Sucessfully	yes	Nil	User
TC-007	Functional	Home Page	Verify the clarifai detect the type of animals in the video	IBM cloud, Python IDLE,Clarifai	1.Load the Code in python 2.Uploading the video in clarifai 3. Specifying the type of animals and birds 4.Store in Cloudant DB	Clarifai, Python Code	Desired output	Working as expected	Fail	Not successful	No	BUG ID-26635	User

8.2 USER ACCEPTING TESTING

PURPOSE OF DOCUMENT

The purpose of this document is to briefly explain the test coverage and open issues of the [Moisture,Temperature, Soil Moisture Level,Motor On and Off] project at the time of the release to User Acceptance Testing (UAT).

DEFECT ANALAYSIS

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
Blocks due to Environment changes in Temp,Hum and Soil Moist	16	8	5	3	32
Control of Motor And Sprinkles ON & OFF	18	10	7	4	39
Continuous Battery Consumption	20	9	3	2	34
Detection Coverage Area	13	6	2	2	23
Altering the Calibration Curve	20	9	7	6	42
Maintenance	11	3	2	2	18
Accuracy detection of Leakage Location	19	10	7	4	40
Totals	117	55	33	23	228

TEST CASE ANALYSIS

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

Section	Total Cases	Not Tested	Fail	Pass
Blocks due to Environment changes in Temp,Hum and Soil Moist	10	0	1	9
Control of Motor And Sprinkles ON & OFF	15	0	0	15
Continuous Battery Consumption	13	0	0	13
Detection Coverage Area	5	0	1	4
Altering the Calibration Curve	4	0	0	4
Maintenance	5	0	0	5
Accuracy detection of Leakage Location	1	0	0	1

9.RESULTS

9.1. PERFORMANCE MERITS

Once the animal or bird is detected, the speaker will be on and the recorded sound is played for 20 seconds and GSM module makes a call to the farmer. Along with the call, repellent system of irritating loud noise like cracker sound is used simultaneously with interval of 4 seconds is used upon the animal. This system works continuously for better effectiveness for protecting the crops from animals.

10.ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

1. It allows farmers to maximize yields using minimum resources such as water, fertilizers, seeds etc.
2. Solar powered and mobile operated pumps save cost of electricity.
3. Smart agriculture use drones and robots which helps in many ways. These improves data collection process and helps in wireless monitoring and control.
4. lt is cost effective method.
5. It delivers high quality crop production.

10.2 DISADVANTAGES

1. The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.
2. The smart farming based equipments require farmers to understand and learn the use of technology. This is major challange in adopting smart agriculture farming at large scale across the countries.

11. CONCLUSION

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 achieving better crop yields thus leading to their economic wellbeing.

12. FUTURE SCOPE

Future scope of our project providing protection system which distinguishes between intruder and an authorized person using RFID's 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.. 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 invaders

14. APPENDIX

14.1 SOURCE CODE FOR RUN IN PYTHON

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "995kq7",
        "typeId": "Test_Device_Type",
        "deviceId":"26635"
    },
    "auth": {
        "token": "o3d471A?EzrQoOU3Y_"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if(m=="motoron"):
        print("*******////Motor is ON////*******")
    elif(m=="motoroff"):
        print("******////Motor is OFF////********")
    elif(m=="sprinkleson"):
        print("******////Sprinkles are ON////********")
    elif(m=="sprinklesoff"):
        print("******////Sprinkles are OFF////********")
    else:
        print("****///WRONG command////********")

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    moist=random.randint(0,100)

    if (moist >= 30):
        a= "High Moisture Level";
    elif(moist ==30):
        a= "Normal Moisture Level";
    else:
        a= "Low Moisture Level";
    myData={'temperature':temp, 'humidity':hum, 'soil_moisture':moist, 'moisture':a}
```

```
   client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
   print("Published data Successfully: %s", myData)
   client.commandCallback = myCommandCallback
   time.sleep(2)

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

except Exception as e:
   print("Exception detected in connecting device: %s" % str(e))
   sys.exit()

#Connecting to IBM watson...
deviceCli.connect()

while True:
   #Getting values from sensors...
   temp= round( random.uniform(0,80),2)
   camera = ["Detected","Not Detected","Not Detected","Not Detected","Not
Detected","Not Detected",]
   camera_reading = random.choice(camera)
   moist= round(random.uniform(0,100),2)
   hum= round(random.uniform(0,100),2)


   #storing the sensor data to send in json format to cloud.
   temp= { 'Temp' : temp_sensor }
   camera_data = { 'Animal attack' : camera_reading}
   moist= { 'Moisture level' : moist_level}
   hum= { 'HUmidity level' : hum_level}

   # publishing Sensor datas 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 Temp = %s C" % temp_sensor,  "to IBM Watson")
      success = deviceCli.publishEvent("PH sensor", "json", PH_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 Moisture level = %s " % moist_level,  "to IBM Watson")
        success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
        sleep(1)



    #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 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("")

#command recived by farmer
deviceCli.commandCallback = myCommandCallback

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

12.2 SOURCE CODE FOR ESP32 (WOWKI ONLINE STIMULATOR)

```
#include <WiFi.h>
#include <PubSubClient.h>
#include "DHTesp.h"

const int DHT_PIN = 15;
#define DHTTYPE DHT22
#define LED 5
DHTesp dhtSensor;

int buzzerPin = 2;
int inputPin = 4;
int pirState = LOW;
int val = 0;

void callback(char* topic, byte* payload, unsigned int payloadLength);


#define ORG "995kq7"
#define DEVICE_TYPE "Test_Device_Type"
#define DEVICE_ID "26635"
#define TOKEN "o3d471A?EzrQoOU3Y_"
String data3;
float h,t;


char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] = "iot-2/cmd/home/fmt/String";  // cmd  REPRESENT command type AND
COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;


WiFiClient WiFiClient;
PubSubClient client(server, 1883, callback, WiFiClient);

void setup(){
  Serial.begin(115200);
  dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
  pinMode(LED, OUTPUT);
  delay(10);
  Serial.println();
  pinMode(buzzerPin, OUTPUT);
  pinMode(inputPin, INPUT);
  WiFiConnect();
```

```
  mqttConnect();

}

void loop() {
  TempAndHumidity  data = dhtSensor.getTempAndHumidity();
  t= data.temperature;
  h= data.humidity;
  Serial.print("temperature:");
  Serial.println(t);
  Serial.print("Humdidty:");
  Serial.println(h);

  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    digitalWrite(buzzerPin, HIGH);  // turn LED ON
    if (pirState == LOW) {
      // we have just turned on
      Serial.println("Motion detected!");
      // We only want to print on the output change, not state
      pirState = HIGH;
    }
  } else {
    digitalWrite(buzzerPin, LOW); // turn LED OFF
    if (pirState == HIGH) {
      // we have just turned of
      Serial.println("Motion ended!");
      // We only want to print on the output change, not state
      pirState = LOW;
    }
  }

  publishData(t,h);
  delay(3000);
  if (!client.loop()) {
    mqttConnect();
  }
}



void publishData(float temp, float humid) {
  mqttConnect();



  String payload = "{\"temperature\":";
  payload += temp;
```

```
  payload += "," "\"humidity\":";
  payload += humid;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
  } else {
    Serial.println("Publish FAILED");
  }

}
void mqttConnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting MQTT client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }
    initManagedDevice();
    Serial.println();
  }
}
void WiFiConnect()
{
  Serial.begin(9600);
  Serial.print("Connecting to WiFi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected: ");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(topic)) {
    // Serial.println(client.subscribe(topic));
    Serial.println("subscribe to cmd OK");
  } else {
```

```
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* topic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(topic);

  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("Data: " + data3 );
  if (data3 == "motoron") {
    Serial.println("motor is on");
    digitalWrite(LED,HIGH);
  }
  else if (data3 == "motoroff") {
    Serial.println(data3);
    digitalWrite(LED,LOW);
  }
  data3 = "";
}
```

13. BIBLIOLOGRAPHY

IBM Cloud : https://cloud.ibm.com/resources

Watson IOT : https://995kq7.internetofthings.ibmcloud.com/dashboard/devices/browse

Node-Red : https://node-red-cbanj-2022-11-17.eu-gb.mybluemix.net/red/#flow/15c292ba6a650aa2

Node-Red UI : https://node-red-cbanj-2022-11-17.eu-gb.mybluemix.net/ui/#!/0?socketid=dK7-g30YXdCloeRXAABF

Github Link : https://github.com/IBM-EPBL/IBM-Project-28457-1660112388

Final DEMO Video Link : https://youtube.com/watch?v=Q4srmsntFHM&feature=share