FINAL DELIVERABLES

PROJECT REPORT

Date	19 November 2022
Team ID	PNT2022TMID32413
Project Name	IoT Based Smart Crop Protection System for Agriculture.

Team Leader

Hariprasath V V(810019106029)

Team Members

Boomika S (810019106019)

Anuja R(810019106008)

Eniyavan P(810019106023)

Bachelor of Engineering In
Electronics and Communication Engineering

University college of engineering BIT campus
Tiruchirappalli-620 024

Project Report Index

7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    1. Feature 1
    2. Feature 2
    3. Database Schema (if Applicable)
8. **TESTING**
9. **ADVANTAGES & DISADVANTAGES**
10. **CONCLUSION**
11. **FUTURE SCOPE**
12. **APPENDIX**
    1. Source Code
    2. GitHub Link

IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

I. Introduction:

1. Project Objectives:

- The device will detect the animals and birds using the Clarifai service.
- If any animal or bird is detected the image will be captured and stored in the IBM Cloud object storage.
- It also generates an alarm and avoid animals from destroying the crop.
- The image URL will be stored in the IBM Cloudant DB service.
- The device will also monitor the soil moisture levels, temperature, and humidity values and send them to the IBM IoT Platform.
- The image will be retrieved from Object storage and displayed in the web application.
- A web application is developed to visualize the soil moisture, temperature, and humidity values.
- Users can also control the motors through web applications.

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.

II. Literature Survey:

1. Existing problem:

Crops in the farms are many times devastated by the wild as well as domestic animals and low productivity of crops is one of the reasons for this. It is not possible to stay 24 hours in the farm to guard the crops.

2. References:

- **Title:** IOT IN AGRICULTURE CROP PROTECTION AND POWER GENERATION (2020)
  Author: Anjana M, Charan Kumar A, Monisha R, Sahana R H

- **Title:** IOT BASED CROP PROTECTION SYSTEM AGAINST AND WILD ANIMAL ATTACKS (2020)
  Author: Navaneetha P, RamiyaDevi R, Vennila S, Manikandan P, Dr. Saravanan S

- **Title:** SMART CROP PROTECTION SYSTEM (2021)
  Author: Krunal Mahajan, Riya Parate, Ekta Zade, Shubham Khante, Shishir Bagal

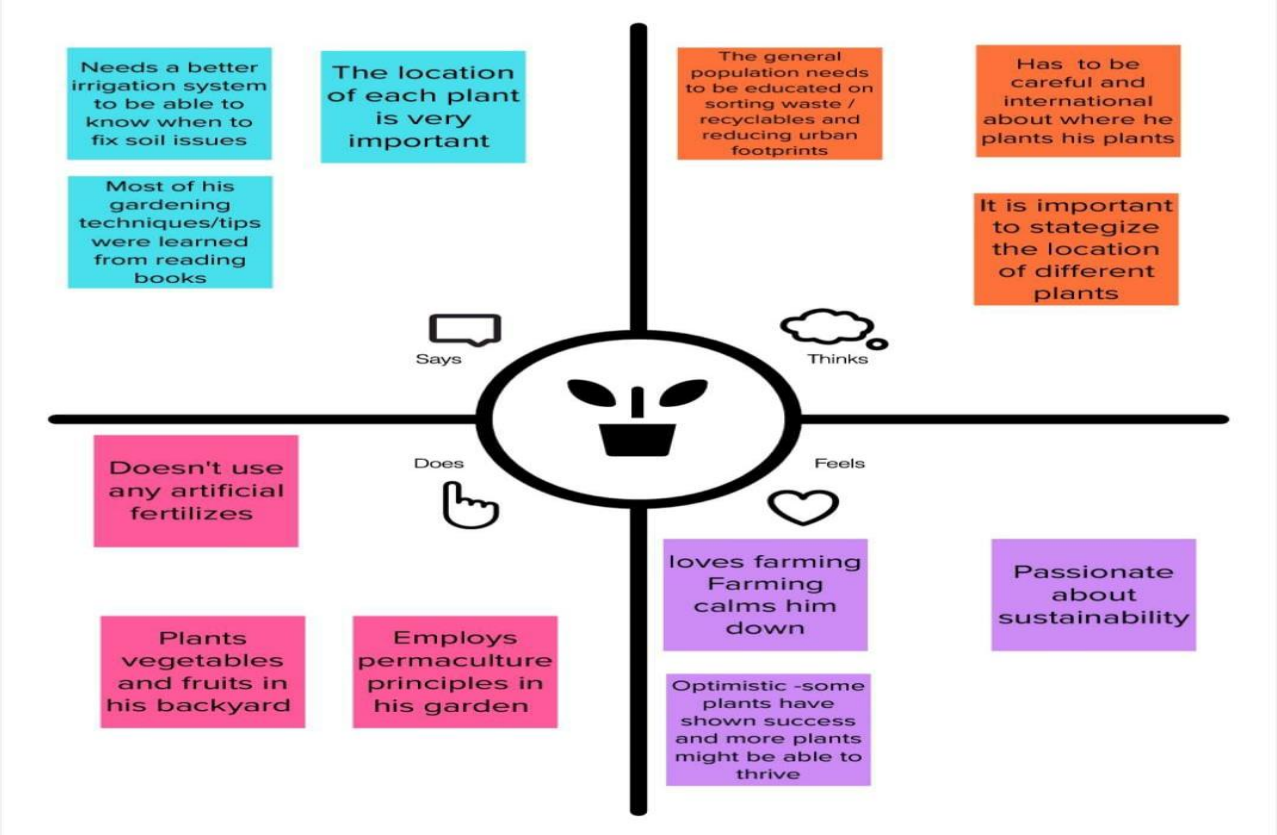
3. Problem statement:

The Farmers (User characteristics) who needs to water their plants on time and to prevent their plants from animals (User need)

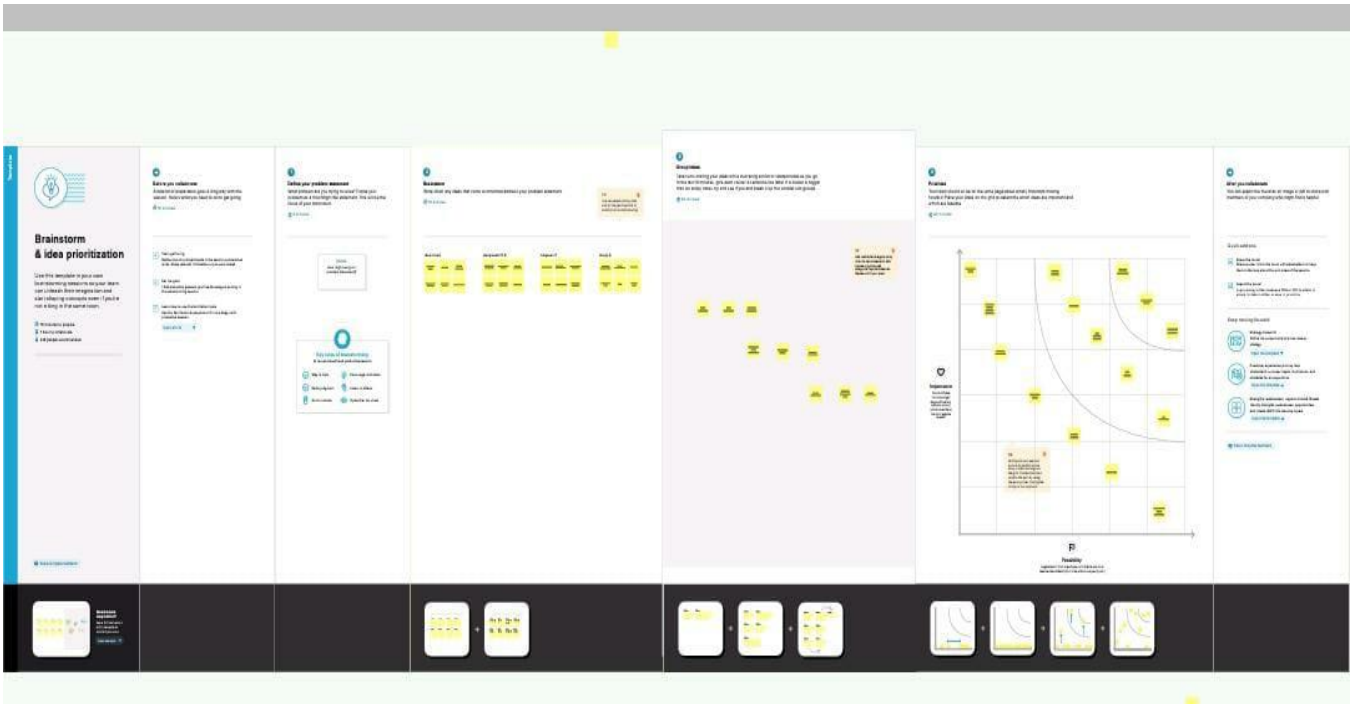
because the plants became dry when watering is improper and animals and birds often affects the field. (Insight)

III. IDEATION & PROPOSED SOLUTION

1. Empathy Map:



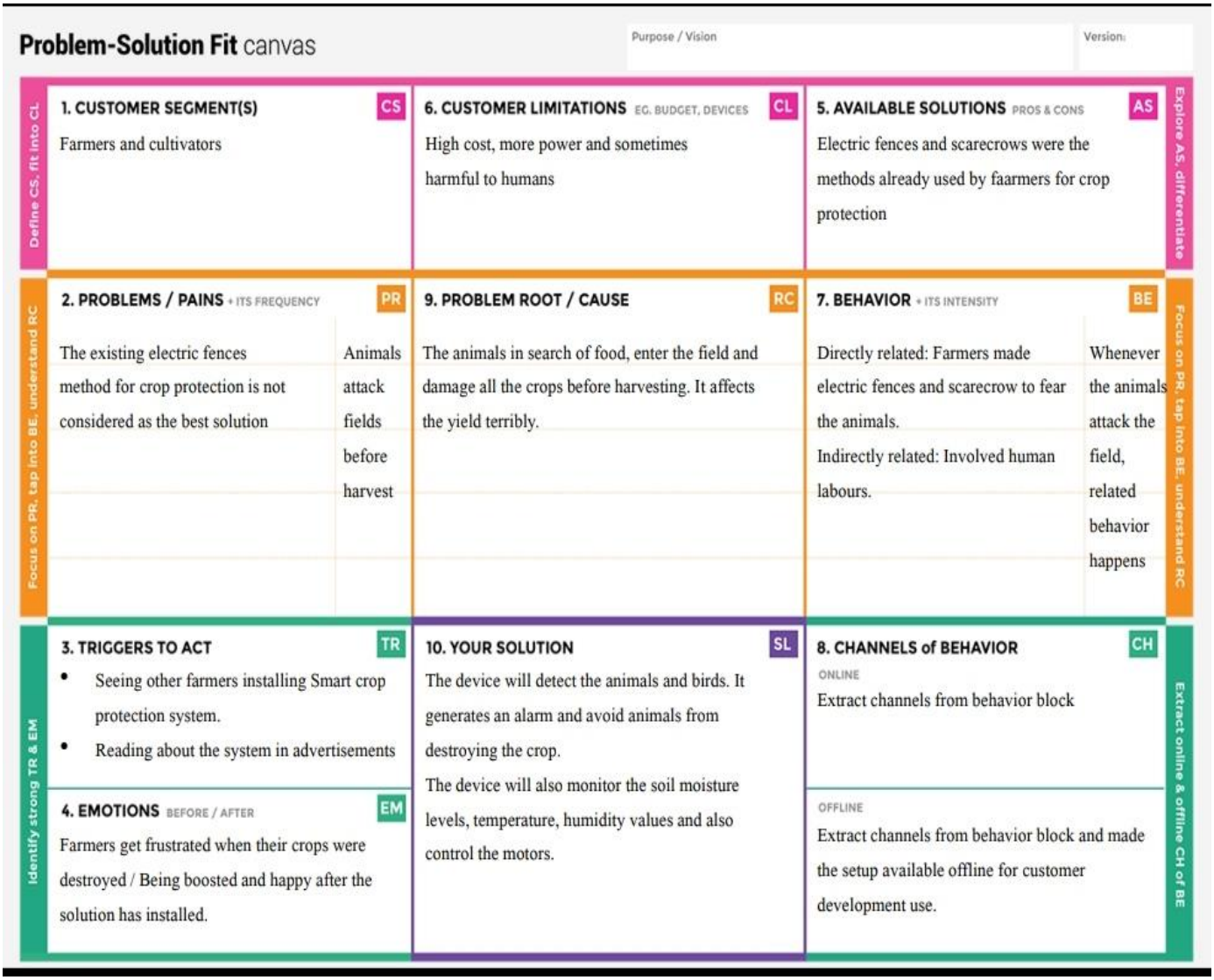
2. Ideation & Brainstorming:



3. Proposed Solution:

Sl No	Parameter	Description
1.	Problem Statement	The farmer who needs to water their plants on time and to prevent their plants from animal then proper watering for field
2.	Solution description	The device will detect the animals and birds.it generate an alarm and avoid animals from destroying the crop
3.	Novelty	The unique of project is to monitor the soil moisture levels,temperature,humidity.
4.	Customer Satisfaction	They can easily protect the field and yielding more profits.
5.	Business Model	Farmers and cooperatives(minimize costs). Farming as a service(Faas) Commerce and government Pay per use. Performing based model. Additional sharing model
6.	Scalability of the Solution	In a field of IOT we proposed to deal with brilliant sensors and electrical equipments to achieve “SMART CROP PROTECTION SYSTEM”

4. Problem Solution fit:



Problem-Solution Fit canvas

Purpose / Vision

Version:

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Farmers and cultivators

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

High cost, more power and sometimes harmful to humans

5. AVAILABLE SOLUTIONS PROS & CONS AS

Electric fences and scarecrows were the methods already used by faarmers for crop protection

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

The existing electric fences method for crop protection is not considered as the best solution

Animals attack fields before harvest

9. PROBLEM ROOT / CAUSE RC

The animals in search of food, enter the field and damage all the crops before harvesting. It affects the yield terribly.

7. BEHAVIOR + ITS INTENSITY BE

Directly related: Farmers made electric fences and scarecrow to fear the animals.

Indirectly related: Involved human labours.

Whenever the animals attack the field, related behavior happens

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

- Seeing other farmers installing Smart crop protection system.
- Reading about the system in advertisements

4. EMOTIONS BEFORE / AFTER EM

Farmers get frustrated when their crops were destroyed / Being boosted and happy after the solution has installed.

10. YOUR SOLUTION SL

The device will detect the animals and birds. It generates an alarm and avoid animals from destroying the crop.

The device will also monitor the soil moisture levels, temperature, humidity values and also control the motors.

8. CHANNELS of BEHAVIOR CH

ONLINE

Extract channels from behavior block

OFFLINE

Extract channels from behavior block and made the setup available offline for customer development use.

Extract online & offline CH of BE

IV. REQUIREMENT ANALYSIS:

1. Functional requirement:

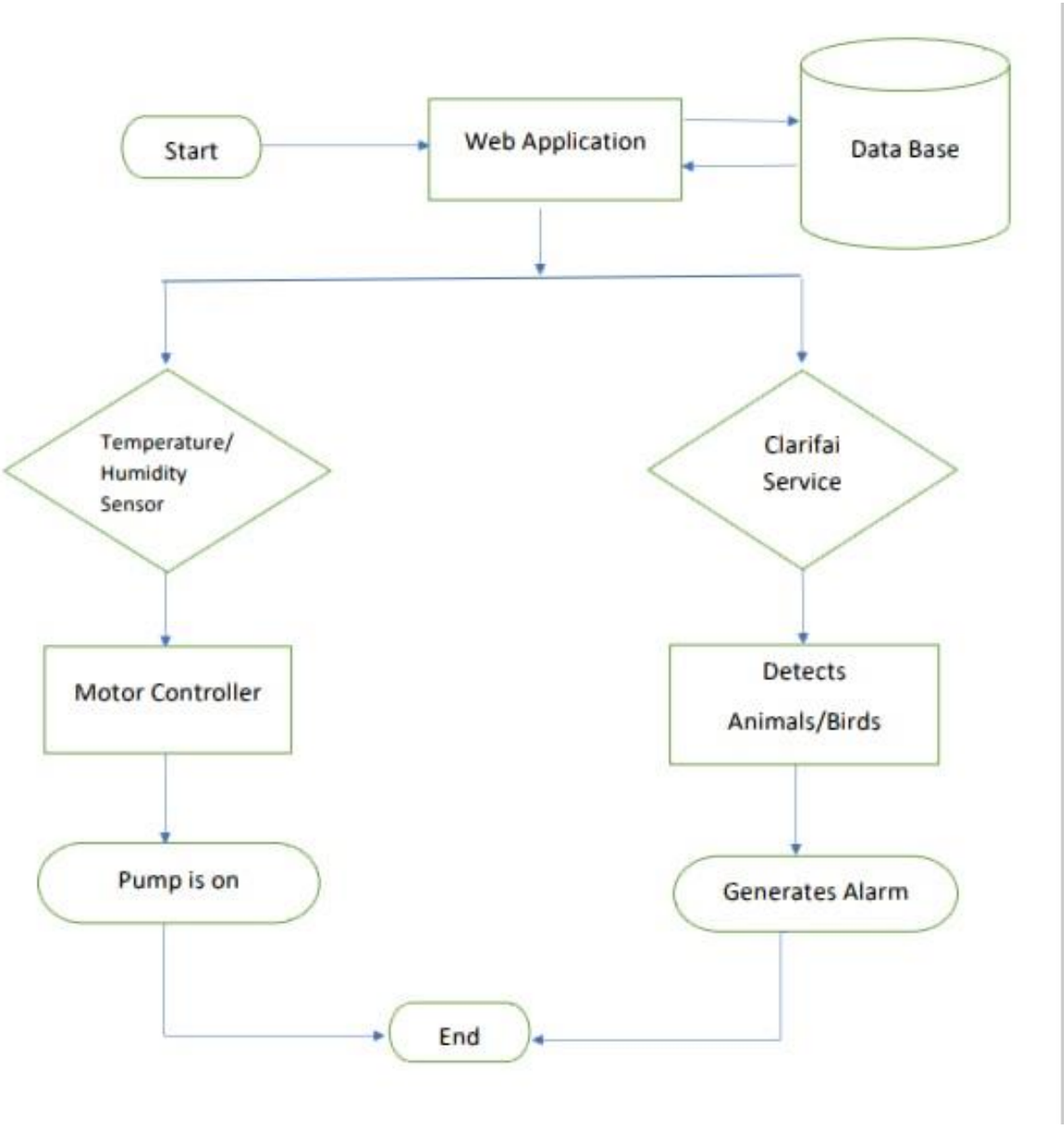
FR No.	Functional requirement (Epic)	Description
FR-1	User registration	Download the app Registration through Gmail Create an account Follow the instructions
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Interface Clarifai service	Interface Clarifai service and so if animals enter the field it gives alarm.
FR-4	Interface sensors	Interface sensors like temperature and humidity sensor to measure the values and to irrigate the field
FR-5	Accessing datasets	Datasets are retrieved from Cloudant DB
FR-6	Mobile application	Motos and sprinklers in the field can be controlled by mobile application.

2.Non –functional requirements

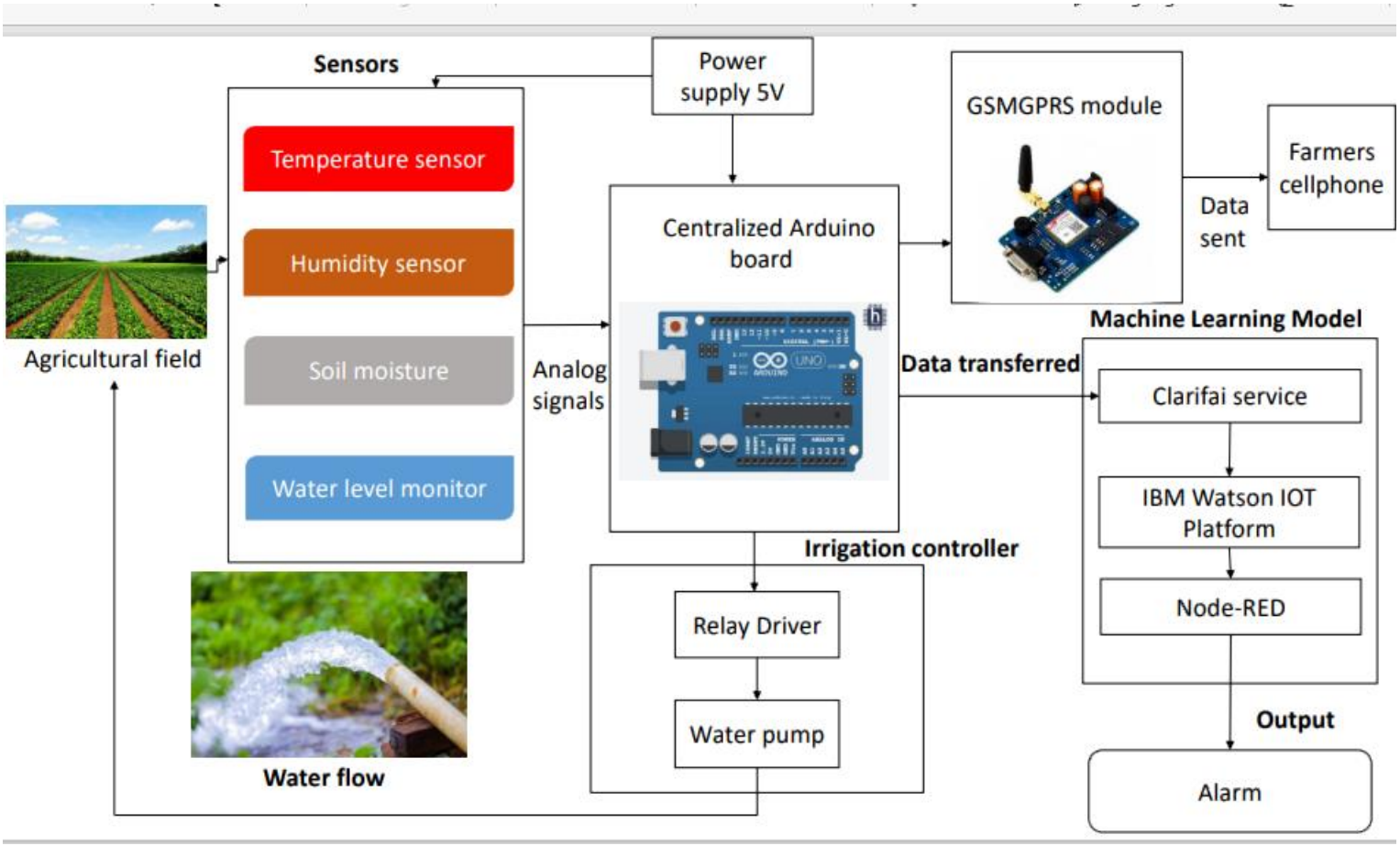
NFR No	Non Functional Requirement	description
NFR-1	Usability	The smart protection system defines that this project helps farmers to protect the farm
NFR-2	Security	We have designed this project to secure the crops from animals.
NFR-3	Reliability	This project will help farmers in protecting their fields and save them from significant financial losses. This will also help them in achieving better crop yields thus leading to their economic well being.
NFR-4	Performance	IOT devices and sensors are used to indicate the farmer by a message when animals try to enter into the field and also we use an SD card module that helps to store a specified sound to scare the animals
NFR-5	Availability	By developing and deploying resilient hardware and software we can protect the crops from wild animals
NFR-6	Scalability	Since this system uses computer vision techniques integrated with IBM cloudant services helps efficiently to retrieve images in large scale thus improving scalability

V. PROJECT DESIGN:

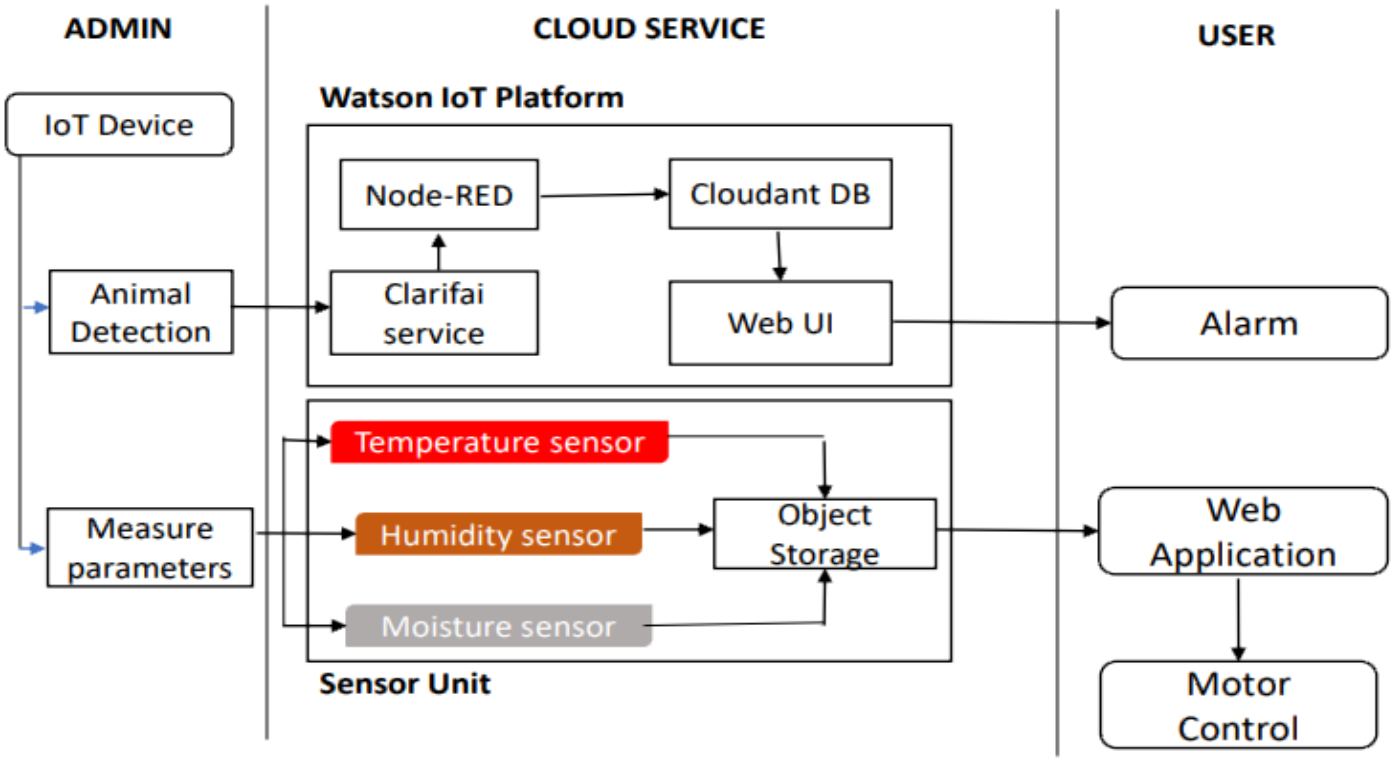
1. Data Flow Diagrams:



2. Solution Architecture:



3.Technology Architecture:



4.Components & technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g., Mobile Application	HTML, CSS, JavaScript / Angular JS / Node Red.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	IoT Model	Purpose of IoT Model is for integrating the sensors with a user interface.	IBM IoT Platform
10	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

3. User Stories:

User Type	Functional requirement (Epic)	User Story number	User Story/Task	Acceptance criteria	Priority	Release
Mobile users	Registration	USN-1	User can enter into the web application	I can access my account /dashboard	High	Sprint 1
		USN-2	User can register their credentials like email id and password	I can receive confirmation email & click confirm	High	Sprint 1
	Login	USN-3	User can log into the application by entering email & password	I can login to my account	High	Sprint 1
	Dashboard	USN-4	User can view the temperature	I can view the data given by the device	High	Sprint 2
		USN-5	User can view the level of sensor monitoring value	I can view the data given by the device	High	Sprint 2
Web users	Usage	USN-1	User can view the web page and get the information	I can view the data given by the device	High	Sprint 3
Customer	Working	USN-1	User act according to the alert given by the device	I can get the data work according to it.	High	Sprint 3
		USN-2	User turns ON the water motors/Buzzer/Sound Alarm when occur the disturbance on field.	I can get the data work according to it.		Sprint 4

Customer care Executive	Action	USN-1	User solve the problem when some faces any usage issues	I can solve the issues when someone fails to understanding the procedure	High	Sprint 4
Administration	Administration	USN-1	User store every information	I can store the gained information	High	Sprint 4

VI. PROJECT PLANNING & SCHEDULING:

1. Sprint Planning & Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	BOOMIKA .S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	ANUJA.R
Sprint-1		USN-3	As a user, I can register for the application through Facebook	2	Low	ENIYAVAN .P
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	HARIPRASATH V.V
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	BOOMIKA. S
Sprint-1	Dashboard	USN-6	As a user, I can log into the application by entering email & password and access all the resources and services available	2	High	HARIPRASATH V.V
Sprint-1	Dashboard	USN-6	As a user, I can log into the application by entering email & password and access all the resources and services available	2	High	HARIPRASATH V.V

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Login	USN-1	As a weather data controller, I log into my profile and start monitoring the weatherupdates	3	High	ENIYAVAN.P
Sprint-2	Dashboard	USN-2	I receive all the information about weather from web from weather API. Whenever there is change in weather, corresponding updates are made on sign boards.	2	Medium	ANUJA. R
Sprint-3	Login	USN-1	As a image controller, I keep note of all the images received from various areas and detect traffic in that particular area.	3	High	HARIPRASATH V.V
Sprint-3	Dashboard	USN-2	With the traffic, updates I change the status of sign board as “take diversion”.	2	Medium	BOOMIKA.S
Sprint-4	Login	USN-1	With the traffic, updates I change the status of sign board as “take diversion With the traffic, updates I change the status of sign board as “take diversion	3	High	ENIYAVAN.P
Sprint-4	Login	USN-1	As an administrator, I ensure that all departments work co-ordinated and ensure the accuracy and efficiency.	2	Medium	HARIPRASATH V.V

2. Sprint Delivery Schedule:

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	20Oct 2022	24 Oct 2022	20	21 Oct 2022
Sprint-2	20	6 Days	25 Oct 2022	29 Oct 2022	20	27 Oct 2022
Sprint-3	20	6 Days	31 Oct 2022	4 Nov 2022	20	2 Nov 2022
Sprint-4	20	6 Days	5 Nov 2022	11 Nov 2022	20	8 Nov 2022

VII. CODING & SOLUTIONING:

1. Feature 1:

IoT based smart crop protection system was implemented using traditional farming concepts and it has a user interfacing system to monitor the temperature humidity and moisture level of the soil. It enables smart farming through that the farmer can access the environmental parameters. The Random module used to generate the values for moisture, temperature and humidity. These values are further sent to the Watson platform.

2. Feature 2:

Further the smart crop protection system was enhanced by creating the user interface. Node red web user interface and MIT app inventor were used to create the user interface. The data from the python script were stored in Watson and the animal detected information were uploaded in the object storage. The opencv2 module is used to capture the animal picture in the field and alter message will be sent to the farmer through the web user interface and mobile application.

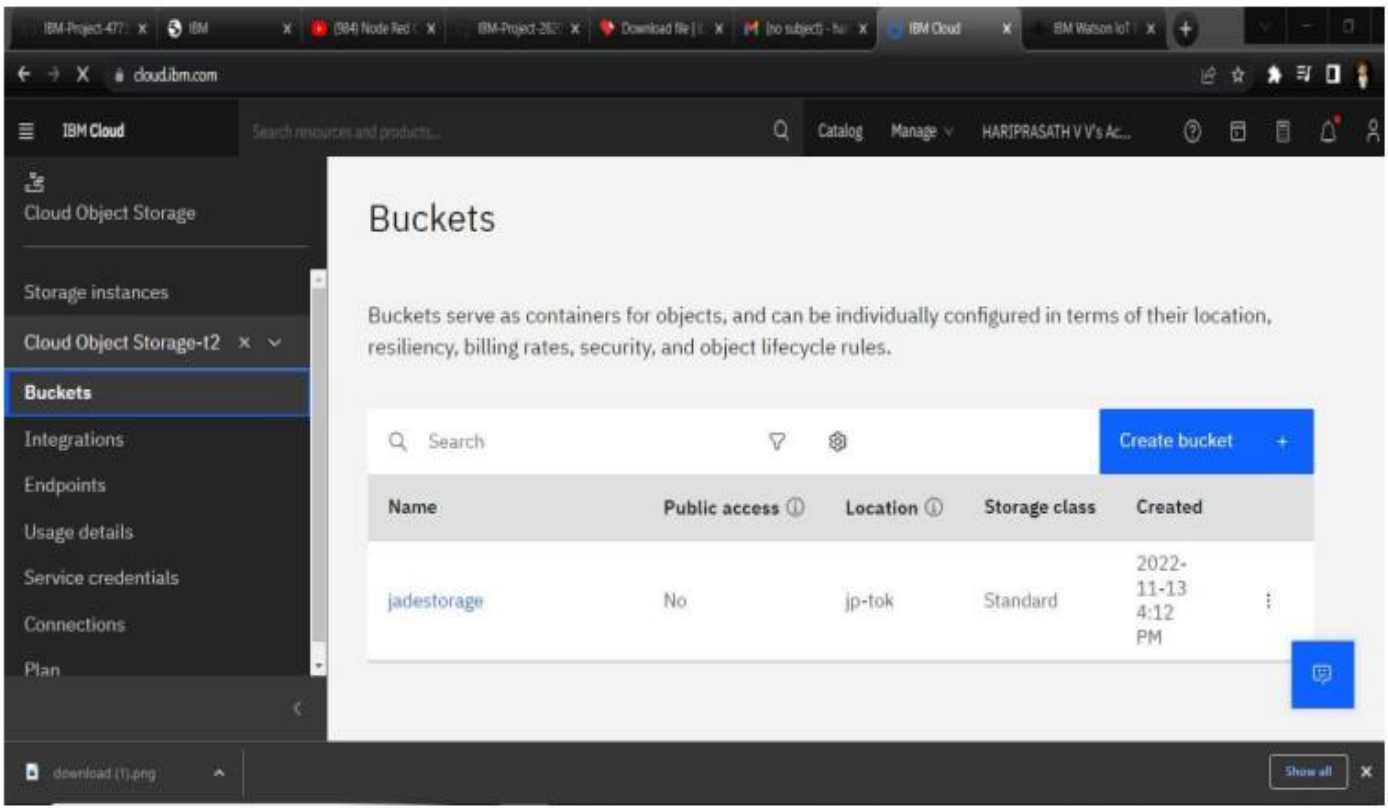
3. Database Schema:

- **IBM Watson IoT platform:**

Random temperature, humidity, and moisture values are generated using the python code and the values are sent to the IBM cloud. IBM cloud sends those values to the node red and shown in the node red dashboard

➤ **Cloud object storage :**

This is the cloud storage area where we can store the images of the detected animal.



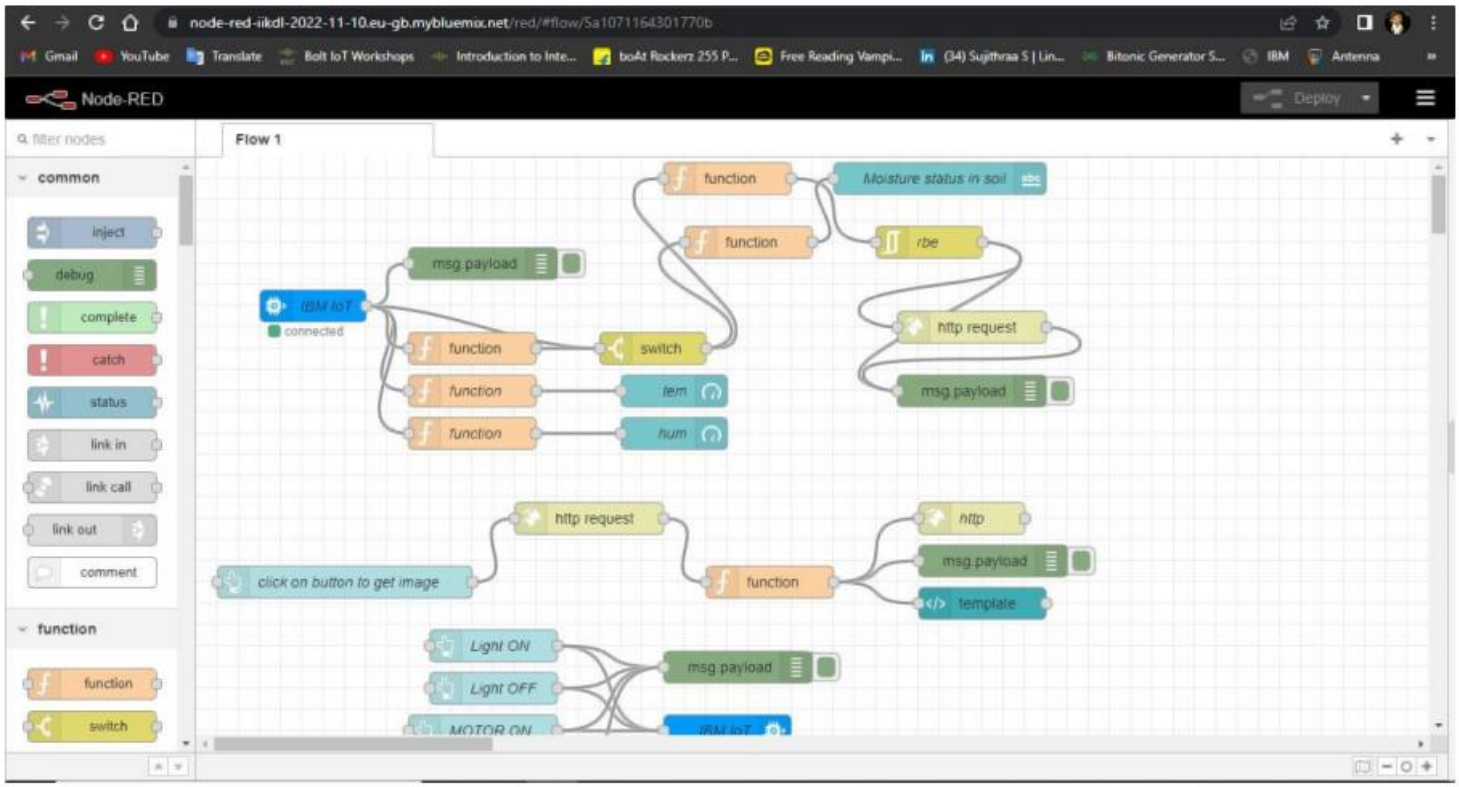
VIII. TESTING

➢ Python code testing:

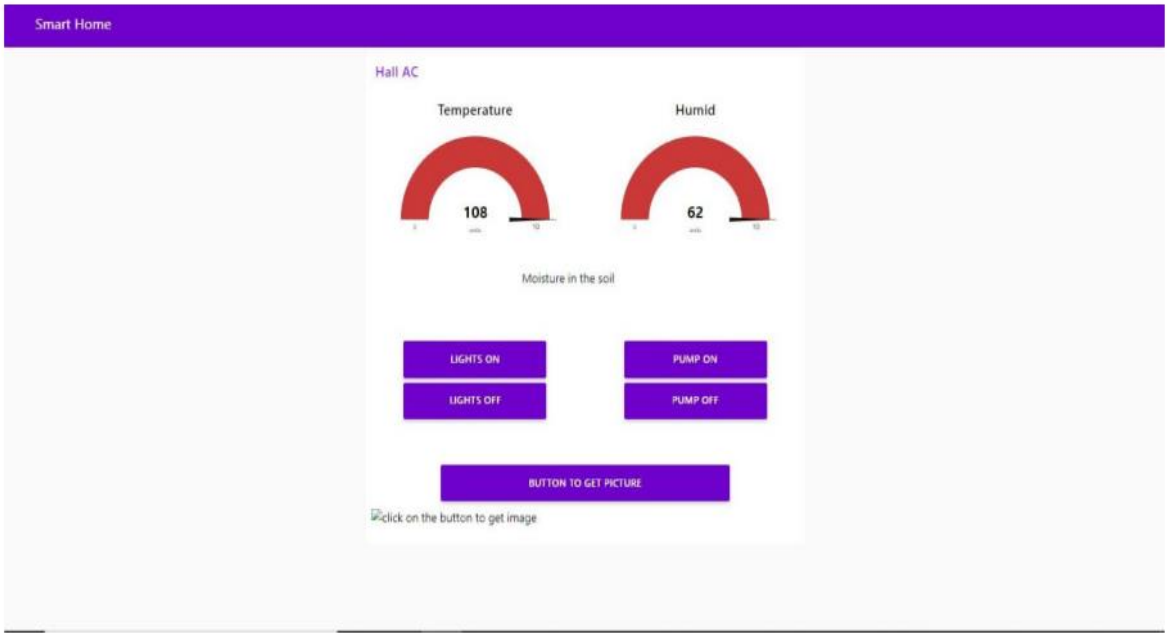
```
sprinkler-1 is ON
Published Alert1 :  Temperature(41.42) is high, sprinkerlers are turned ON to IBM Watson

Published Alert2 :  Fertilizer PH level(7.063) is not safe,use other fertilizer to IBM Watson

Published Alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published Alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson

Published Alert5 :  Moisture level(97.48) is low, Irrigation started to IBM Watson

Published Alert6 :  water level(14.02) is high, so motor is ON to take water out  to IBM Watson

... ...publish ok... ... ...
Published Temp = 37.64 C to IBM Watson
Published PH value = 7.008 to IBM Watson
Published Animal attack Not Detected  to IBM Watson
Published Flame Not Detected  to IBM Watson
Published Moisture level = 59.12  to IBM Watson
Published Water level = 22.61 cm to IBM Watson

sprinkler-1 is ON
Published Alert1 :  Temperature(37.64) is high, sprinkerlers are turned ON to IBM Watson

Published Alert2 :  Fertilizer PH level(7.008) is not safe,use other fertilizer to IBM Watson

Published Alert3 :  Animal attack on crops detected to IBM Watson to IBM Watson

Published Alert4 :  Flame is detected crops are in danger,sprinklers turned ON to IBM Watson

Published Alert5 :  Moisture level(59.12) is low, Irrigation started to IBM Watson

Motor-2 is turning ON
Published Alert6 :  water level(22.61) is high, so motor is ON to take water out  to IBM Watson

... ...publish ok... ... ...
Published Temp = 19.07 C to IBM Watson
```

➢ Node-red

Connection and output:

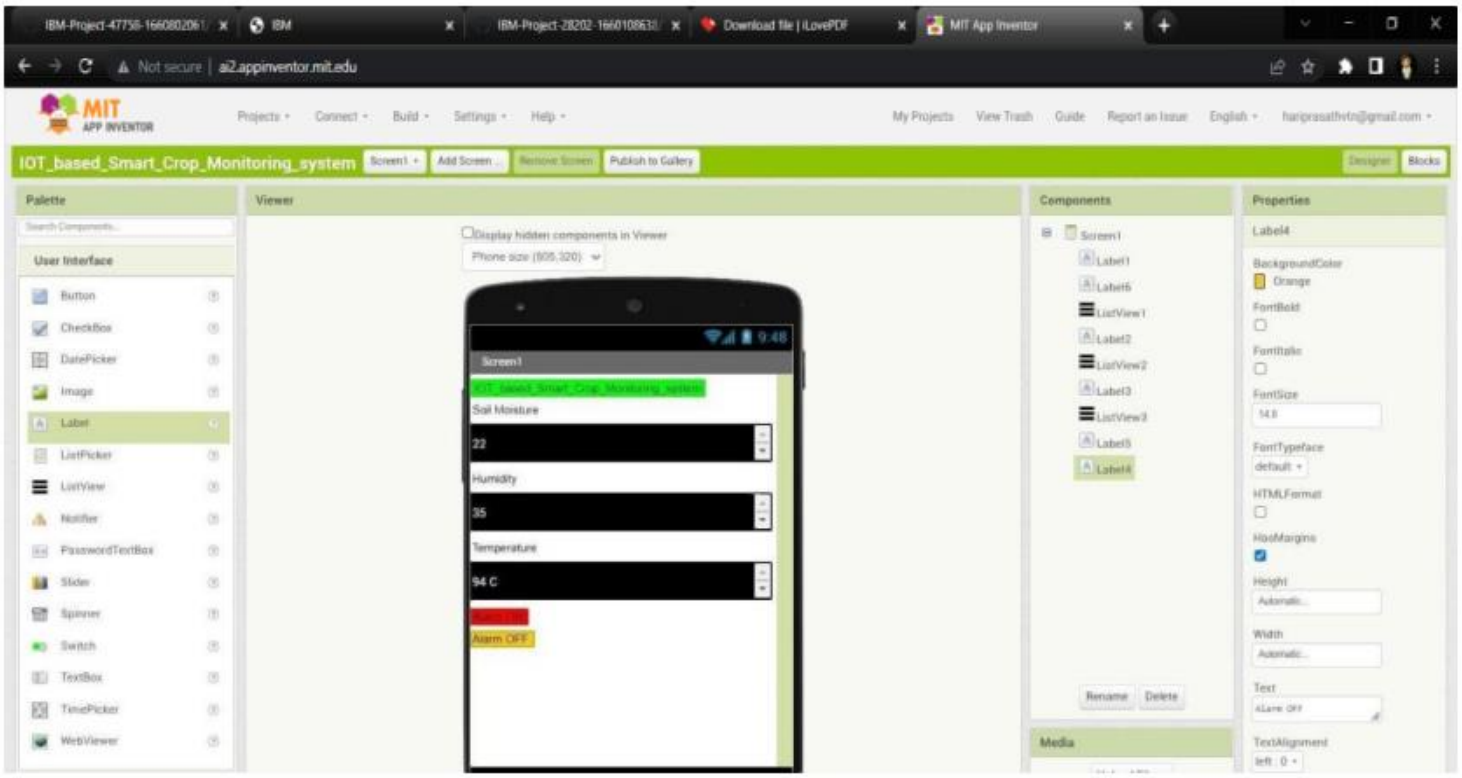


Web application testing:



- **Mobile Application testing**

Mobile application creation :



IX. ADVANTAGES & DISADVANTAGES:

1. ADVANTAGES:

•Farmers can monitor the health of farm animals closely, even if they are physically distant.

•Smart farming systems reduce waste, improve productivity and enable management of a greater number of resources through remote sensing.

• High reliance.

• Enhanced Security.

2. DISADVANTAGES:

• Farms are located in remote areas and are far from access to the internet.

•A farmer needs to have access to crop data reliably at any time from any location, so connection issues would cause an advanced monitoring system to be useless.

• High Cost

• Equipment needed to implement IoT in agriculture is expensive.

X. CONCLUSION

As a result of this system, we can detect the changes in the field easily and intimate the farmers about it and also, we can take precautions and do remedies accordingly. Here we use very low power consuming highly efficient components that give us accurate results and also, they perform at low data rate conditions without any lag and help in finding the remedies. This crop protection system helps in detection of all kinds of external dangers and it saves time and money to the farmers before any loss that may occur. With the help of this system the farmers can be in a peaceful environment at ease without any pressure.

XI. FUTURE SCOPE:

Study and analysis of the developed Crop protection systems for its cost effectiveness with the development of Arduino based variable frequency Ultrasonic bird deterrent circuit. outline of the crop damage caused by a particular Wild animal if the behavioural features of the with the reduced cost in the smart phones.

XII. APPENDIX:

1. Source code:

```
import cv2
import numpy as np
import wiotp.sdk.device
import playsound
import random
import time
import datetime
import ibm_boto3
from ibm_botocore.client import Config, ClientError

#CloudantDB
from cloudant.client import Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, ResultByKey
from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
from clarifai_grpc.grpc.api import service_pb2_grpc
stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
from clarifai_grpc.grpc.api import service_pb2, resources_pb2
from clarifai_grpc.grpc.api.status import status_code_pb2

#This is how you authenticate
metadata = (('authorization', 'key 83ddcfb774c54cfd81d7a67ba69a0678'),)
```

```
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"

COS_API_KEY_ID = "kn05el2QeCyawCFMRytUXLFirKVxw8v5HAIRvDKsIHmu"

COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

COS_RESOURCE_CRN="crn:v1:bluemix:public:cloudantnosqldb:eu-gb:a/98d92dfd0ccf4f32a116d3d0fe24e15c:02d1fcad-1310-4403-93a6-a0eabc4c768b::"

Clientdb=Cloudant("apikey-v2-d8mn8ful7bxv3pw2cq0o1p1d8z3icznh8qu8y2xsv5", "400eef0a90d31fd7fa41c9dd0a2baa4b", url="https://cbf0b64e-c2d3-4404-be21-36565dc150b9-bluemix.cloudantnosqldb.appdomain.cloud")

clientdb.connect()

#Create resource

cos = ibm_boto3.resource("s3",

   ibm_api_key_id=COS_API_KEY_ID,

   ibm_service_instance_id=COS_RESOURCE_CRN,

   ibm_auth_endpoint=COS_AUTH_ENDPOINT,

   config=Config(signature_version="oauth"),

   endpoint_url=COS_ENDPOINT

)

def multi_part_upload(bucket_name, item_name, file_path):

   try:

  print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))

      #set 5 MB chunks

      part_size = 1024 * 1024 * 5

      #set threadhold to 15 MB
```

```
    file_threshold = 1024 * 1024 * 15
    #set the transfer threshold and chunk size
    transfer_config = ibm_boto3.s3.transfer.TransferConfig(
          multipart_threshold=file_threshold,
          multipart_chunksize=part_size
    )

    #the upload_fileobj method will automatically execute a multi-part upload
    #in 5 MB chunks size
    with open(file_path, "rb") as file_data:
       cos.Object(bucket_name, item_name).upload_fileobj(
       Fileobj=file_data,
        Config=transfer_config
       )
    print("Transfer for {0} Complete!\n".format(item_name))
 except ClientError as be:
    print("CLIENT ERROR: {0}\n".format(be))
 except Exception as e:
    print("Unable to complete multi-part upload: {0}".format(e))
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    command=cmd.data['command']
    #print(command)
```

```
        if(command=="lighton"):
            print('lighton')
        elif(command=="lightoff"):
            print('lightoff')
        elif(command=="motoron"):
            print('motoron')
        elif(command=="motoroff"):
            print('motoroff')
myConfig = {
   "identity": {
      "orgId": "tw9ckq",
      "typeId": "node",
      "deviceId": "6020"
   },
   "auth": {
      "token": "27102001"
   }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

database_name = "sample1"
my_database = clientdb.create_database(database_name)
if my_database.exists():
```

```
    print(f"'{database_name}' successfully created.")
cap=cv2.VideoCapture("garden.mp4")

if(cap.isOpened()==True):
    print('File opened')
else:
    print('File not found')
while(cap.isOpened()):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    imS= cv2.resize(frame, (960,540))
    cv2.imwrite('ex.jpg',imS)
    with open("ex.jpg", "rb") as f:
    file_bytes = f.read()
    detect=False
    t=random.randint(-1,1)
    if(t==0):
          detect=True
          print("Alert! Alert! animal detected")
         #playsound.playsound('alert.mp3')
         picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
           cv2.imwrite(picname+'.jpg',frame)
       multi_part_upload('jadestorage', picname+'.jpg', picname+'.jpg')
json_document=
{"link":COS_ENDPOINT+'/'+'jadestorage'+'/'+picname+'.jpg'}
```

```
 new_document = my_database.create_document(json_document)
if new_document.exists():
        print(f"Document successfully created.")
        time.sleep(5)

   moist=random.randint(0,100)
   humidity=random.randint(0,200)
   temperature=random.randint(0,100)
   myData={'Animal':detect,'moisture':moist,'hum':humidity,'temp':temperature}
   print(myData)
   if(humidity!=None):
      client.publishEvent(eventId="status",msgFormat="json",          data=myData,
qos=0, onPublish=None)
      print("Publish Ok..")

   client.commandCallback = myCommandCallback
   cv2.imshow('frame',imS)
   if cv2.waitKey(1) & 0xFF == ord('q'):
      break
client.disconnect()
cap.release()
cv2.destroyAllWindows()
```

› **GITHUB LINK:**

https://github.com/IBM-EPBL/IBM-Project-10308-1659163331