IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

Team ID : PNT2022TMID40471

Team Members:

UKKESH.S

RAMYA.H

LAVANYA .A.P

ABIRAMI.P

INTRODUCTION

PROJECT OVREVIEW:

Crops in farms are many times ravaged by local animals like buffaloes, cows, goats, birds etc. this leads to huge losses for the farmers. It is not possible for farmers to barricade entire fields or stay on field 24 hours and guard it.so here we propose automatic crop protection system from animals. This is a microcontroller based system using PIC family microcontroller. The microcontroller now sound an alarm to woo the animal away from the field as well as sends SMS to the farmer so that he may about the issue and come to the spot in case the animal don't turn away by the alarm. This ensures complete safety of crop from animals thus protecting farmers loss.

PURPOSE:

Our main purpose of the project is to develop intruder alert to the farm, to avoid losses due to animal and fire. These intruder alert protect the crop that damaging that indirectly increase yield of the crop. The develop system will not harmful and injurious to animal as well as human beings. Theme of project is to design a intelligent security system for farm protecting by using embedded system.

LITERATURE SURVEY

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 and manual surveillance and various such exhaustive and dangerous method.

REFERENCES:

i. Mr. Pranav shitap, Mr. Jayesh redj , Mr .Shikhar Singh, Mr. Durvesh Zagade, Dr. Sharada Chougule. Department of ELECTRONICS AND TELECOMMUNICATION ENGINEERING, Finolex Academy of Management and technology, ratangir i, India.

ii. N .Penchalaiah, D. Pavithra, B. Bhargavi, D.P .Madhurai,
K. EliyasShaik, S.Md. sohaib.Assitant Professor, Department of CSE,AITS, Rajampe t,India UG Student, Department of CSE ,AITS , Rajampet, India.

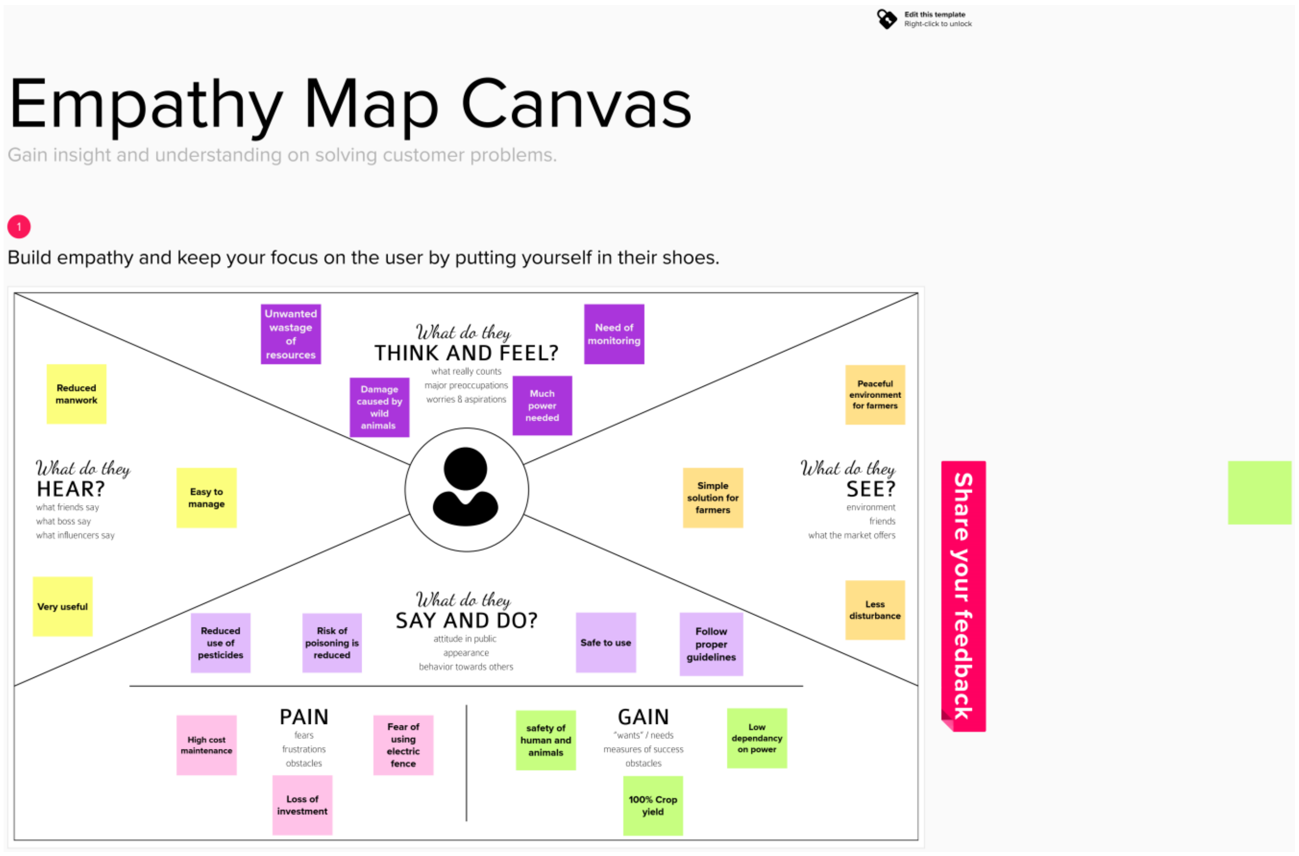
PROBLEM STATEMENT DEFINITION STATEMENT:

In the world economy of many Country dependent 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 materials for industries. But because of animal interference and fire in agricultural lands, there will be huge loss of crops. Crops will be totally getting destroyed.

<u>IDEATION AND PROPOSED SOLUTION</u>

EMPATHY MAP CANVAS:



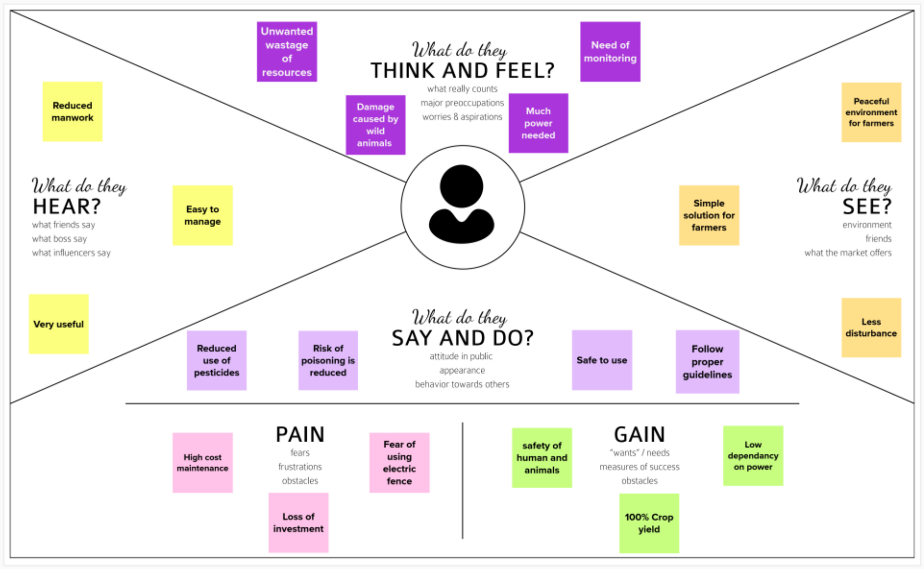
Edit this template
Right-click to unlock

Empathy Map Canvas

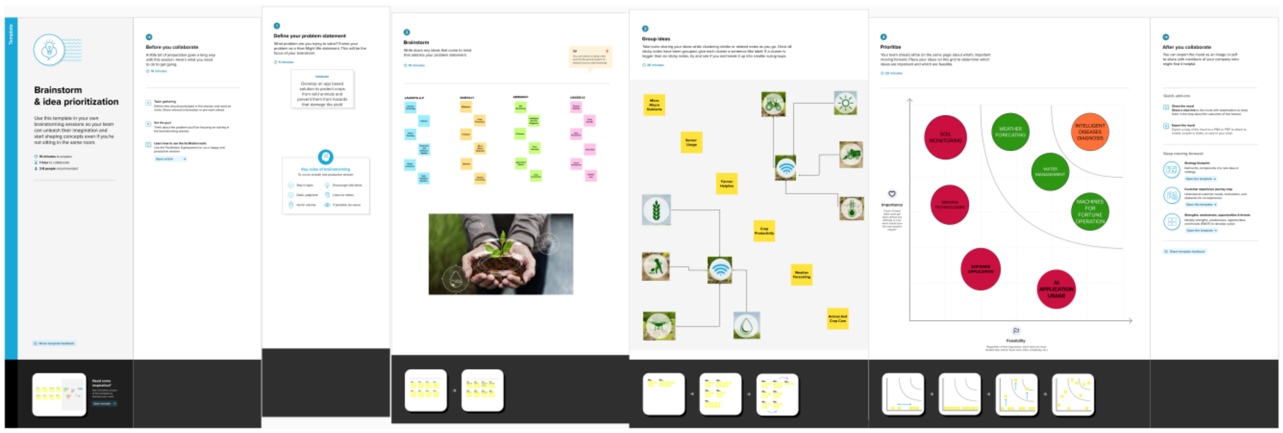
Gain insight and understanding on solving customer problems.

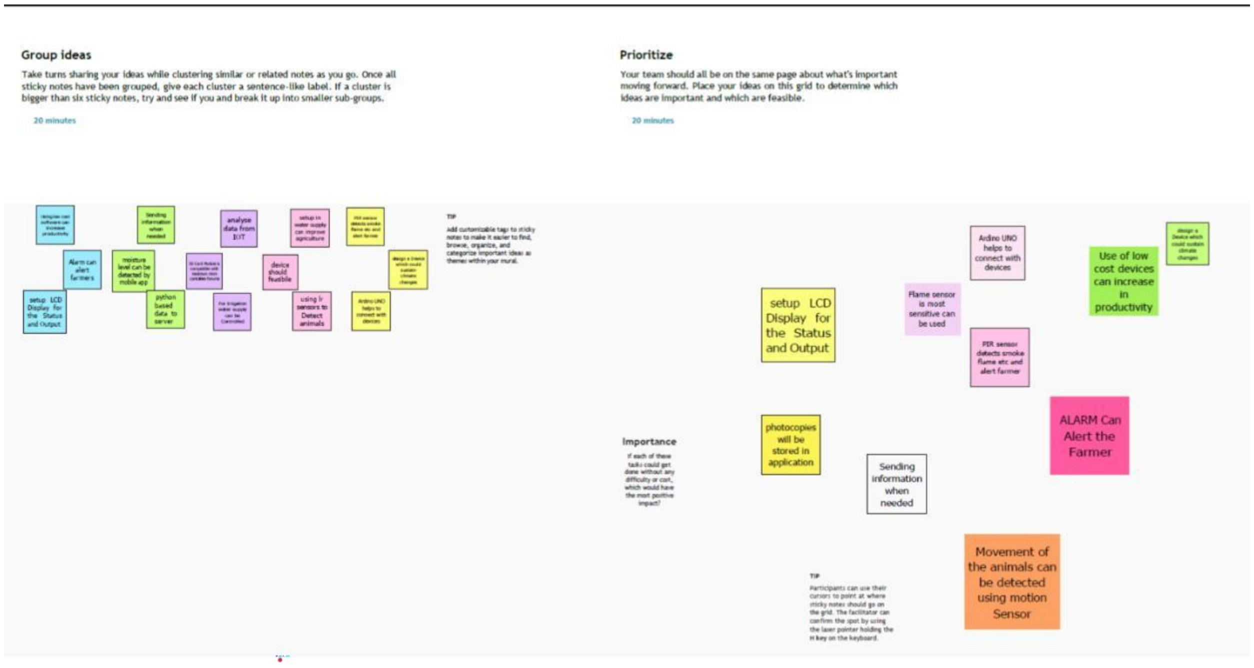
1

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



Share your feedback





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

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

[illegible]
Sending information when needed
analyse data from IOT
setup in water supply can improve agriculture
[illegible]
Alarm can alert farmers
moisture level can be detected by mobile app
[illegible]
device should feasible
design a Device which could sustain climate changes
setup LCD Display for the Status and Output
python based data to server
For Irrigation water supply can be Controlled
using Ir sensors to Detect animals
Ardino UNO helps to connect with devices

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

Importance

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

TIP

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

Ardino UNO helps to connect with devices
Use of low cost devices can increase in productivity
design a Device which could sustain climate changes
setup LCD Display for the Status and Output
Flame sensor is most sensitive can be used
PIR sensor detects smoke flame etc and alert farmer
ALARM Can Alert the Farmer
photocopies will be stored in application
Sending information when needed
Movement of the animals can be detected using motion Sensor

PROPOSED SOLUTION:

Project Design Phase-I
Proposed Solution Template

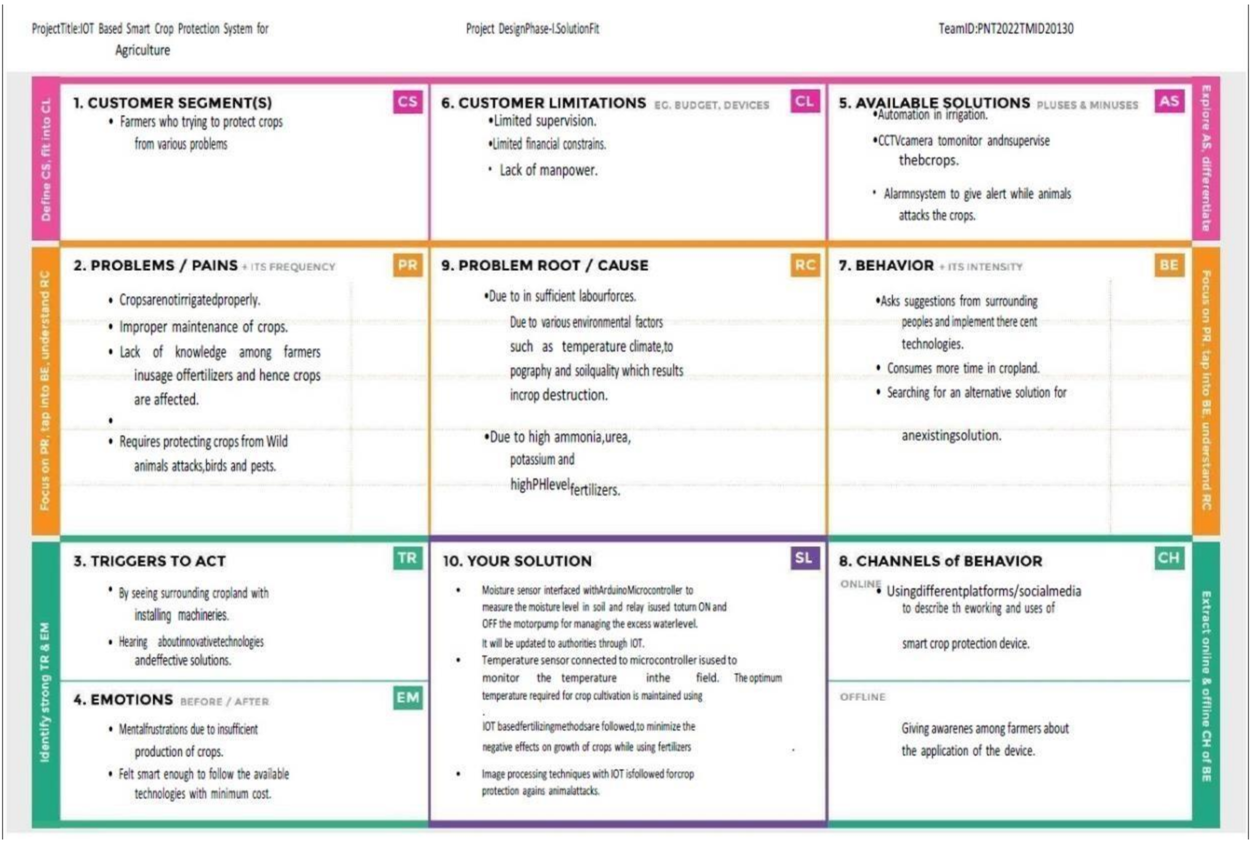
Date	19 September 2022
Team ID	PNT2022TMID40471
Project Name	Project – IOT-Based smart crop protection for agriculture
Maximum Marks	2 Marks

Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop.
2.	Idea / Solution description	This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field. .
3.	Novelty / Uniqueness	The motors and sprinklers in the field can be controlled using the mobile application.
4.	Social Impact / Customer Satisfaction	The damage of crop is reduced and yeild of crop is increased.There is no need of monitoring the Field 24×7
5.	Business Model (Revenue Model)	User friendly and Portable.
6.	Scalability of the Solution	There is no chance to failure.

PROBLEM SOLUTIONFIT:



ProjectTitle:IOT Based Smart Crop Protection System for Agriculture

Project DesignPhase-I.SolutionFit

TeamID:PNT2022TMID20130

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

- Farmers who trying to protect crops from various problems

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- Limited supervision.
- Limited financial constrains.
- Lack of manpower.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

- Automation in irrigation.
- CCTVcamera tomonitor andnsupervise thebcrops.
- Alarmnsystem to give alert while animals attacks the crops.

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- Cropsarenotirrigatedproperly.
- Improper maintenance of crops.
- Lack of knowledge among farmers inusage offertilizers and hence crops are affected.
-
- Requires protecting crops from Wild animals attacks,birds and pests.

9. PROBLEM ROOT / CAUSE RC

- Due to in sufficient labourforces.
  Due to various environmental factors such as temperature climate,to pography and soilquality which results incrop destruction.
- Due to high ammonia,urea, potassium and highPHlevelfertilizers.

7. BEHAVIOR + ITS INTENSITY BE

- Asks suggestions from surrounding peoples and implement there cent technologies.
- Consumes more time in cropland.
- Searching for an alternative solution for

  anexistingsolution.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

- By seeing surrounding cropland with installing machineries.
- Hearing aboutinnovativetechnologies andeffective solutions.

4. EMOTIONS BEFORE / AFTER EM

- Mentalfrustrations due to insufficient production of crops.
- Felt smart enough to follow the available technologies with minimum cost.

10. YOUR SOLUTION SL

- Moisture sensor interfaced withArduinoMicrocontroller to measure the moisture level in soil and relay isused toturn ON and OFF the motorpump for managing the excess waterlevel.
  It will be updated to authorities through IOT.
- Temperature sensor connected to microcontroller isused to monitor the temperature inthe field. The optimum temperature required for crop cultivation is maintained using
  .
  IOT basedfertilizingmethodsare followed,to minimize the negative effects on growth of crops while using fertilizers .
- Image processing techniques with IOT isfollowed forcrop protection agains animalattacks.

8. CHANNELS of BEHAVIOR CH

ONLINE

- Usingdifferentplatforms/socialmedia to describe th eworking and uses of

  smart crop protection device.

OFFLINE

Giving awarenes among farmers about the application of the device.

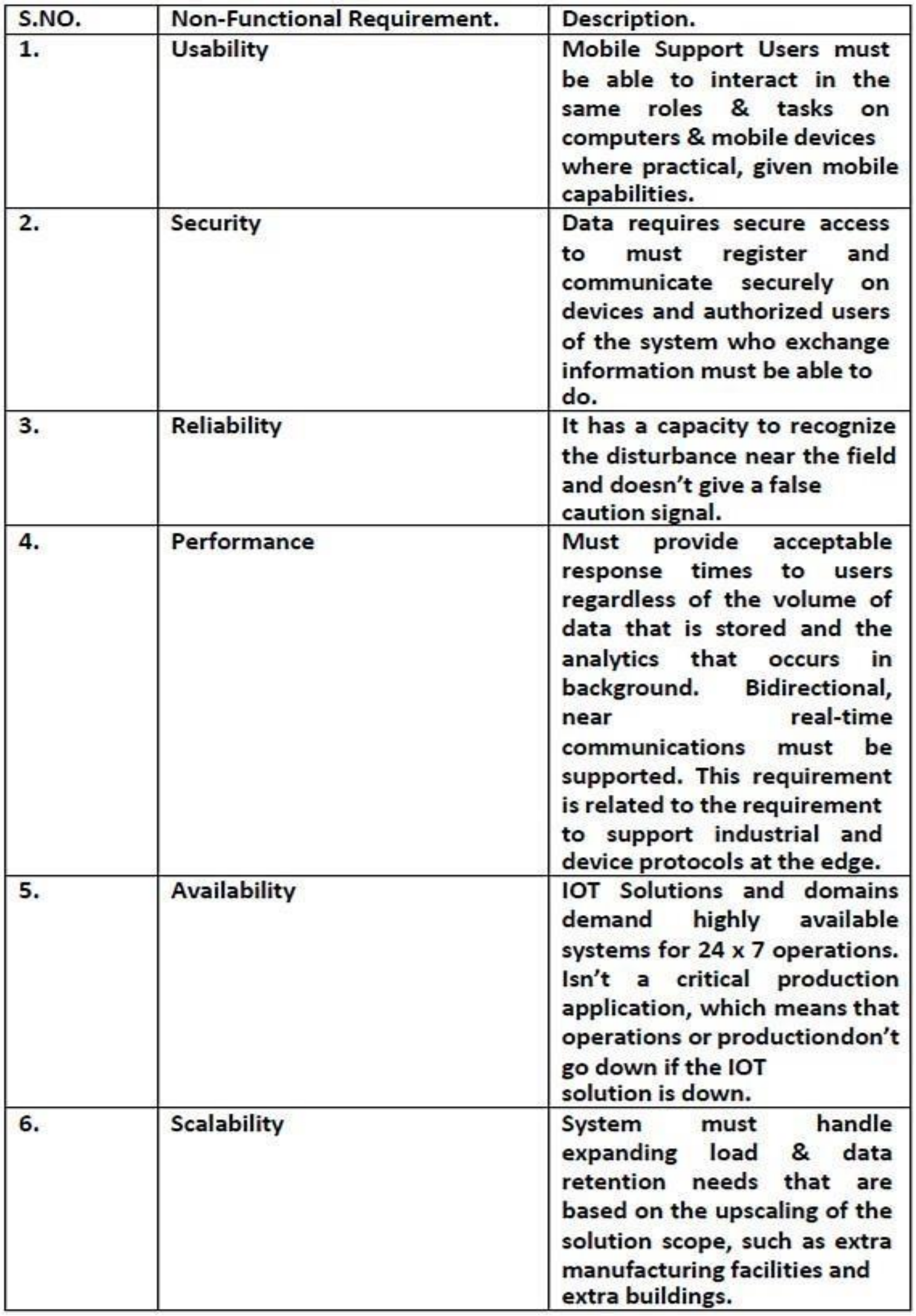
Extract online & offline CH of BE

<u>REQUIREMENT ANALYSIS</u>

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.

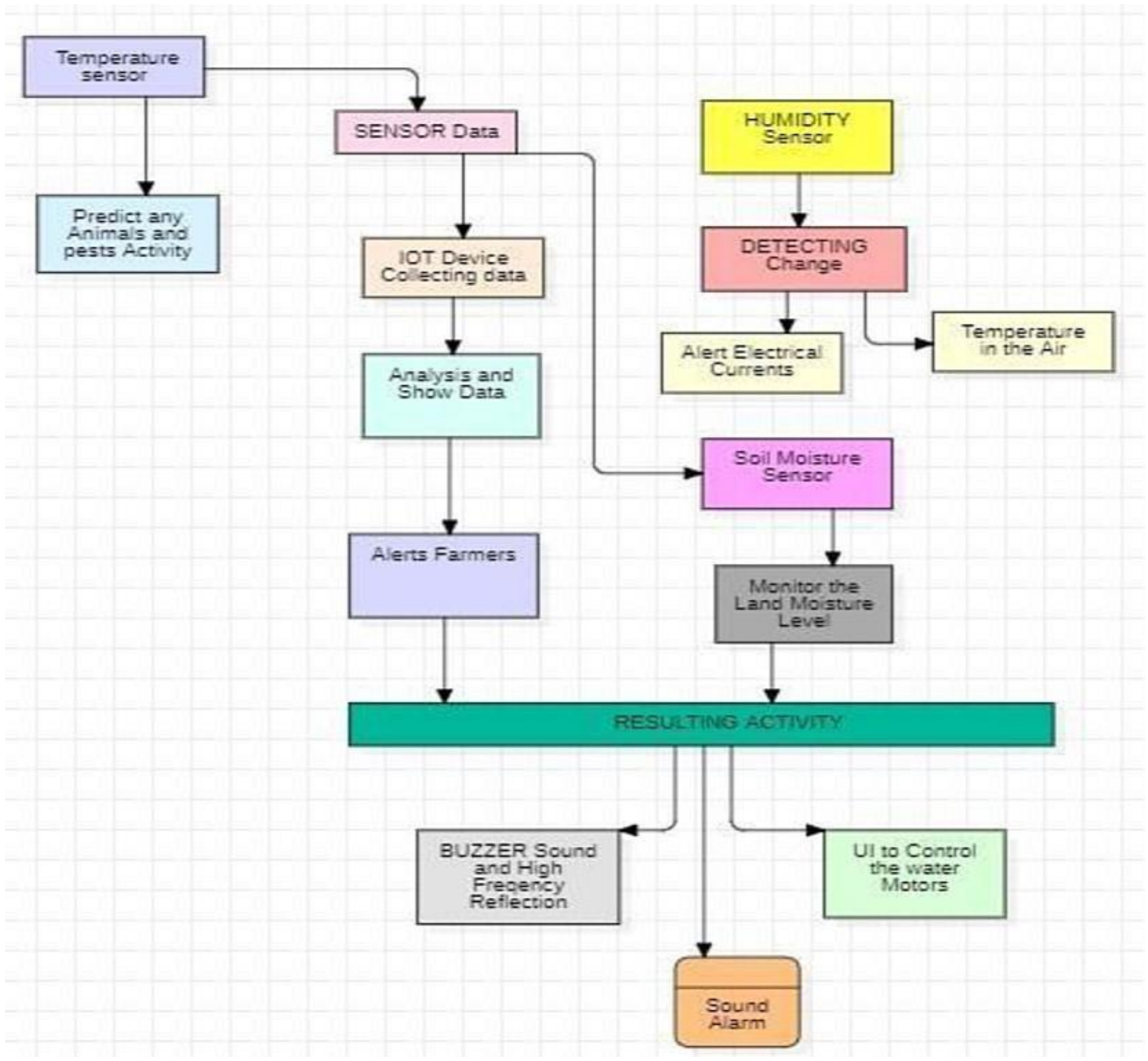
NON FUNCTINAL REQUIREMENT:



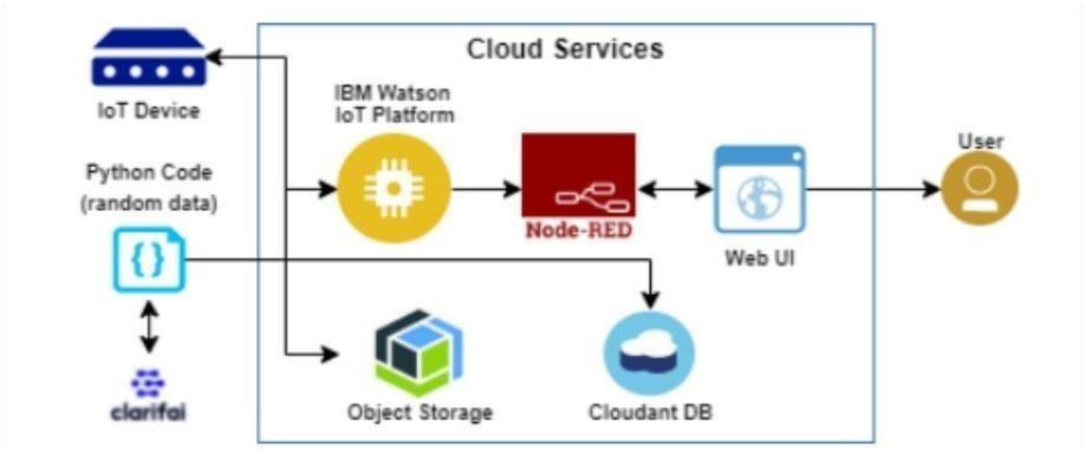
S.NO.	Non-Functional Requirement.	Description.
1.	Usability	Mobile Support Users must be able to interact in the same roles & tasks on computers & mobile devices where practical, given mobile capabilities.
2.	Security	Data requires secure access to must register and communicate securely on devices and 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.

PROJECT DESIGN

DATA FLOW DIAGRAM:


Temperature sensor
SENSOR Data
HUMIDITY Sensor
Predict any Animals and pests Activity
IOT Device Collecting data
DETECTING Change
Alert Electrical Currents
Temperature in the Air
Analysis and Show Data
Soil Moisture Sensor
Alerts Farmers
Monitor the Land Moisture Level
RESULTING ACTIVITY
BUZZER Sound and High Freqency Reflection
UI to Control the water Motors
Sound Alarm

SOLUTION AND TECHNICAL ARCHITECTURE:



a.

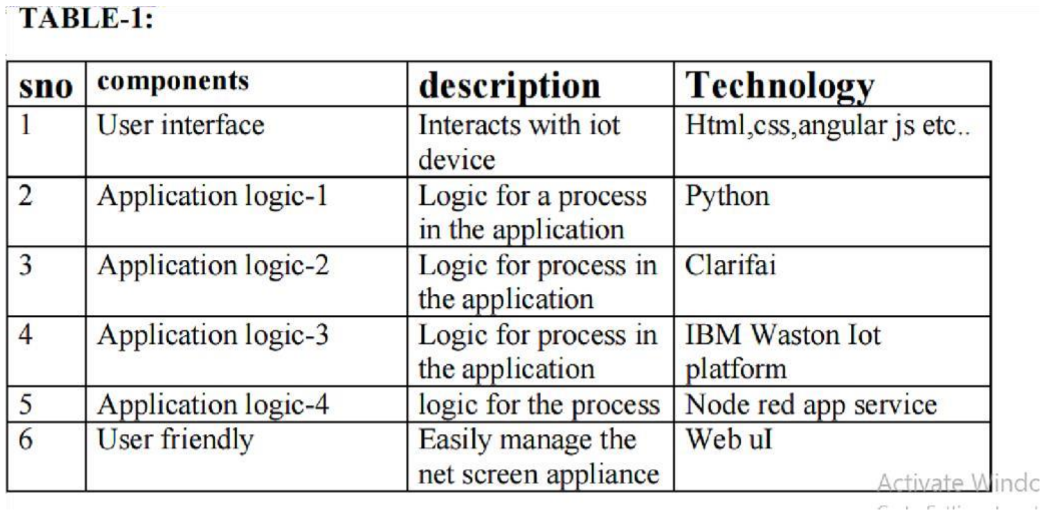


TABLE-1:

sno	components	description	Technology
1	User interface	Interacts with iot device	Html,css,angular js etc..
2	Application logic-1	Logic for a process in the application	Python
3	Application logic-2	Logic for process in the application	Clarifai
4	Application logic-3	Logic for process in the application	IBM Waston Iot platform
5	Application logic-4	logic for the process	Node red app service
6	User friendly	Easily manage the net screen appliance	Web uI

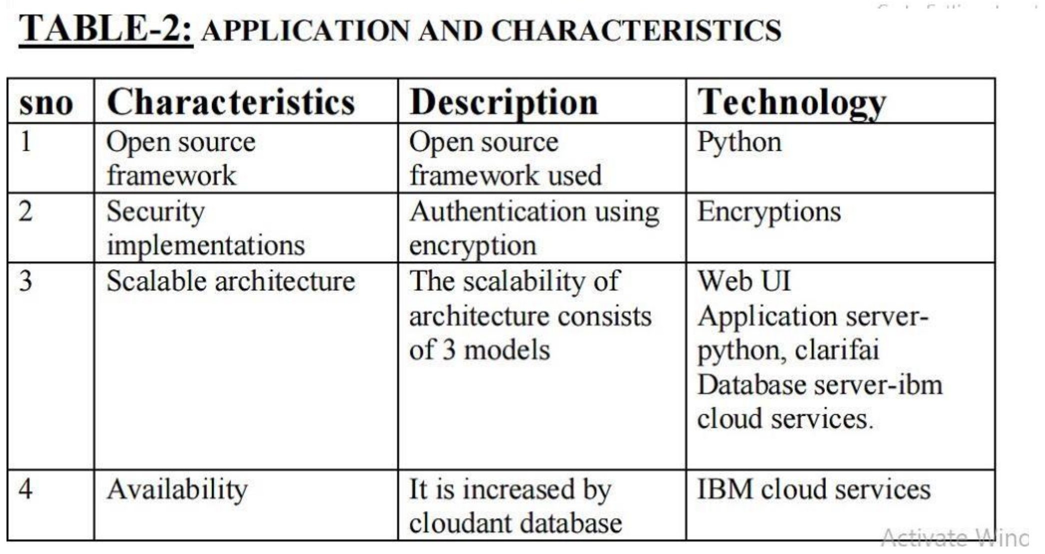


TABLE-2: APPLICATION AND CHARACTERISTICS

sno	Characteristics	Description	Technology
1	Open source framework	Open source framework used	Python
2	Security implementations	Authentication using encryption	Encryptions
3	Scalable architecture	The scalability of architecture consists of 3 models	Web UI Application server-python, clarifai Database server-ibm cloud services.
4	Availability	It is increased by cloudant database	IBM cloud services

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 sensor data such as temperature, moisture, soil and humidity to the IBM IoT platform	6	medium
Sprint-3		US-1	After developing python code, commands are received just print the statements which represent the control of the devices.	8	high
Sprint-4		US-3	Publish Data to The IBM Cloud	5	high
Sprint-4		US-2	Create Web UI in Node- Red	8	high
Sprint-4		US-1	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	6	high

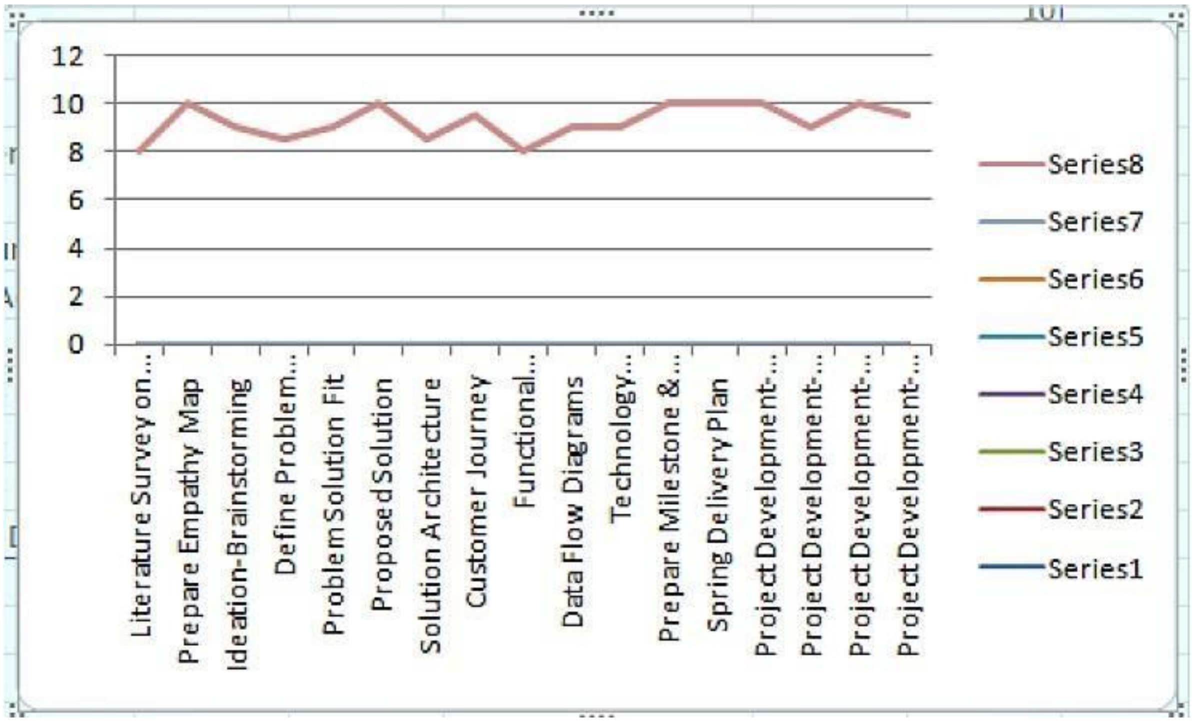
PROJECT PLANNINGAND SCHEDULING

SPRINT PLANNINGAND ESTIMATION:

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
iteration unit (story points per day)

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



CODING AND SOLUTIONING

FEATURE 1:

```
#include

 #include WiFiClient

  wifiClient; String data3;

  #define ORG "lkagkz"

  #define DEVICE_TYPE

  "lav15" #define DEVICE_ID

  "121212" #define TOKEN

  "U*dGW+dWzn04OS01xo

  " #define speed 0.034

  #define led 14 char

  server[] = ORG

  ".messaging.internetofthi

  ngs.ibmcloud.com"; char

  publishTopic[] = "iot-

  2/evt/Data/fmt/json";

  char topic[] = "iot-
```

```
2/cmd/home/fmt/String";
char authMethod[] = "use-
token-auth"; char token[]
= TOKEN; char clientId[] =
"d:" ORG ":" DEVICE_TYPE
":" DEVICE_ID;
PubSubClient
client(server, 1883,
wifiClient); void
publishData(); const int
trigpin=2; const int
echopin=4; String
command; String data="";
long duration; float dist;
void setup() {
Serial.begin(115200);
pinMode(led, OUTPUT);
pinMode(trigpin,
OUTPUT);

pinMode(echopin, INPUT);

wifiConnect();

mqttConnect(); } void

loop() { bool isNearby =

dist < 100;

digitalWrite(led,

isNearby); publishData();

delay(500); if

(!client.loop()) {

mqttConnect(); } } void

wifiConnect() {

Serial.print("Connecting to

"); Serial.print("Wifi");

WiFi.begin("Wokwi-

GUEST", "", 6); while

(WiFi.status() !=

WL_CONNECTED) {
```

```
delay(500);

Serial.print("."); }

Serial.print("WiFi

connected, IP address: ");

Serial.println(WiFi.localIP(

)); } 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

initManagedDevice() { if
```

```
(client.subscribe(topic)) {

//

Serial.println(client.subscr

ibe(topic));

Serial.println("IBM

subscribe to cmd OK"); }

else {

Serial.println("subscribe

to cmd FAILED"); } } void

publishData() {

digitalWrite(trigpin,LOW);

digitalWrite(trigpin,HIGH);

delayMicroseconds(10);

digitalWrite(trigpin,LOW);

duration=pulseIn(echopin,

HIGH);

dist=duration*speed/2;

if(dist<100){ String
```

```
payload = "{\"Normal

Distance\":"; payload +=

dist; payload += "}";

Serial.print("\n");

Serial.print("Sending

payload: ");

Serial.println(payload); if

(client.publish(publishTopi

c, (char*) payload.c_str()))

{ Serial.println("Publish

OK"); } } if(dist>101 &&

dist<111){ String payload =

"{\"Alert distance\":";

payload += dist; payload

+= "}"; Serial.print("\n");

Serial.print("Sending

payload: ");

Serial.println(payload);
```

```
if(client.publish(publishTo
pic, (char*)
payload.c_str())) {
Serial.println("Warning
crosses 110cm -- it
automaticaly of the
loop");
digitalWrite(led,HIGH);
}else {
Serial.println("Publish
FAILED"); } } } void
callback(char*
subscribeTopic, byte*
payload, unsigned int
payloadLength){
Serial.print("callback
invoked for topic:");
Serial.println(subscribeTo
```

pic); for(int i=0; i

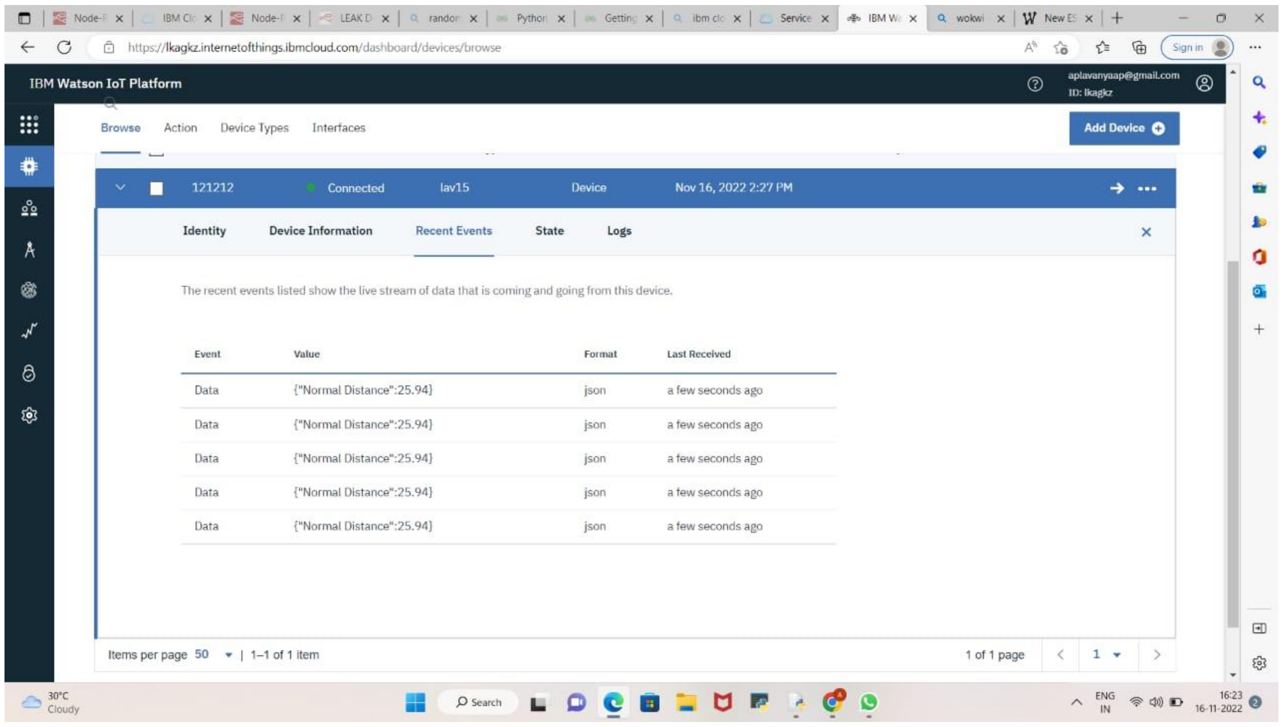
Features

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

BUZZER

Specifications

- Rated Voltage : 6V DC



121212 Connected lav15 Device Nov 16, 2022 2:27 PM

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Data	{"Normal Distance":25.94}	json	a few seconds ago
Data	{"Normal Distance":25.94}	json	a few seconds ago
Data	{"Normal Distance":25.94}	json	a few seconds ago
Data	{"Normal Distance":25.94}	json	a few seconds ago
Data	{"Normal Distance":25.94}	json	a few seconds ago

Items per page 50 | 1–1 of 1 item 1 of 1 page

- Opera ng Voltage : 4 to 8V DC
- Rated Current*: ≤30mA
- Sound Output at 10cm* : ≥85dB
- Resonant Frequency : 2300 ±300Hz
- Tone: Continuous 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 vehicles such as ambulances, police cars and fire trucks. There are two general types, pneuma c and electronic.

FEATURE-2:

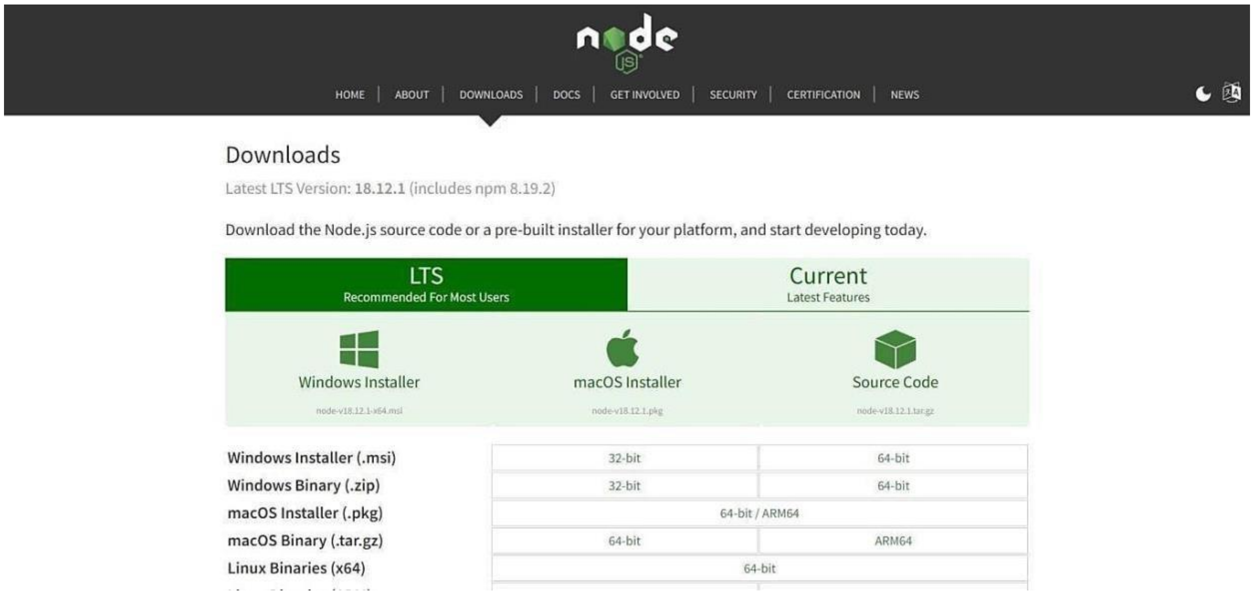
i. Good sensitivity to Combustable gas in wide range .
ii. High sensitivity to LPG, Propane and Hydrogen .
iii. Long life and low cost.
iv. Simple drive circuit.

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%	

User Acceptance Testing:



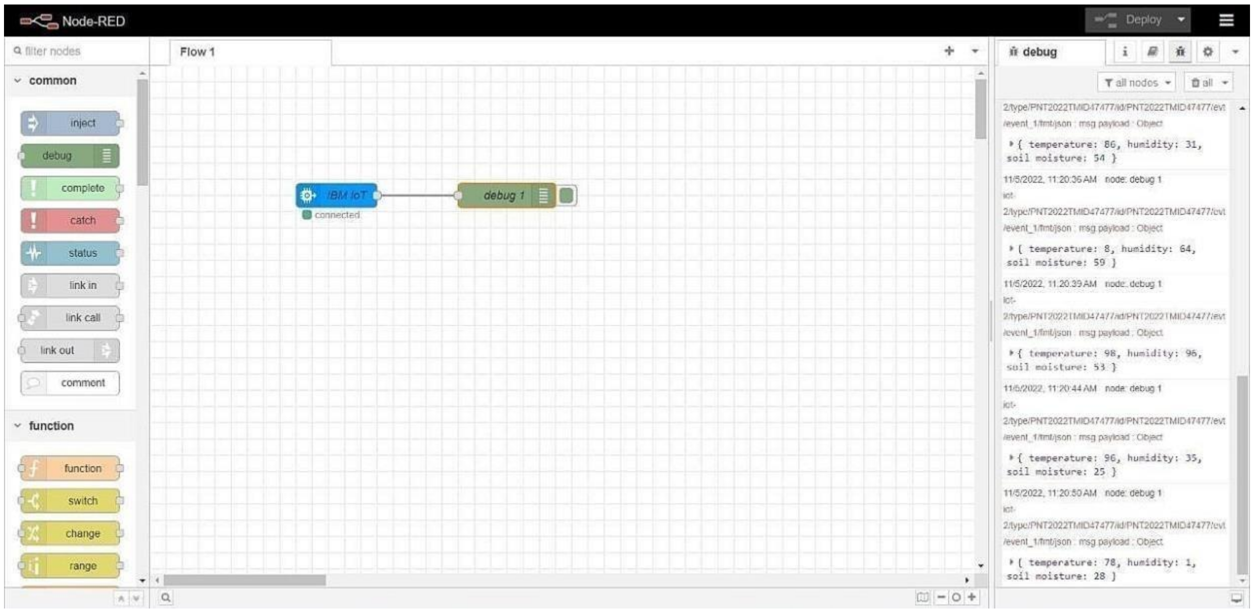
Downloads

Latest LTS Version: **18.12.1** (includes npm 8.19.2)

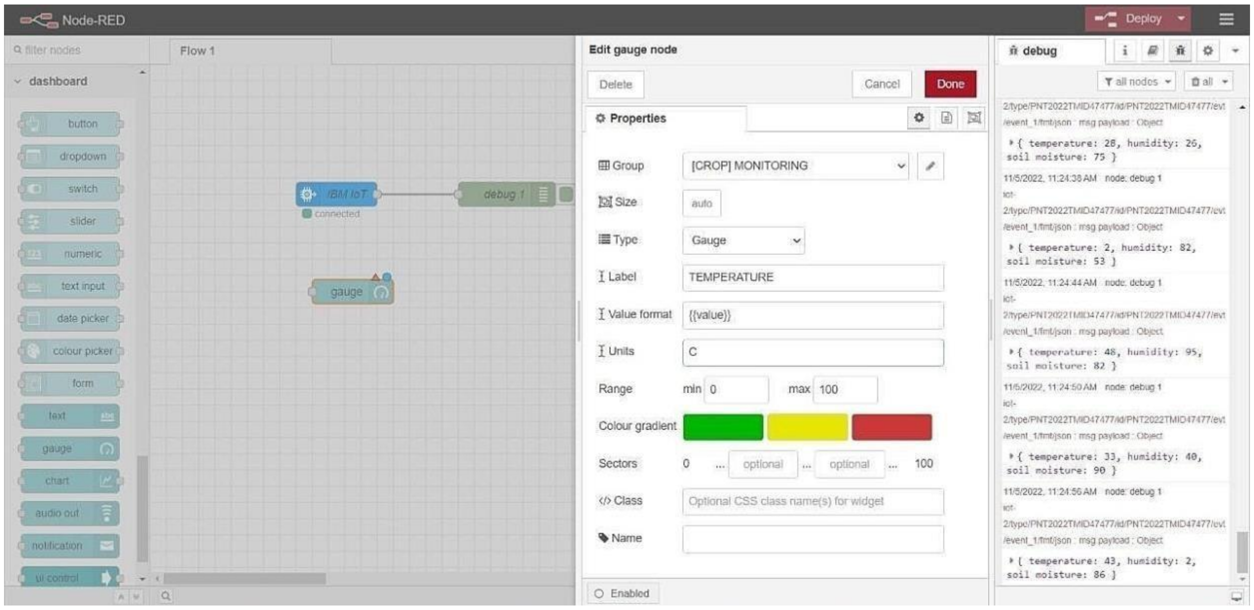
Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users		Current Latest Features
Windows Installer node-v18.12.1-x64.msi	macOS Installer node-v18.12.1.pkg	Source Code node-v18.12.1.tar.gz

Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit / ARM64	
macOS Binary (.tar.gz)	64-bit	ARM64
Linux Binaries (x64)	64-bit	

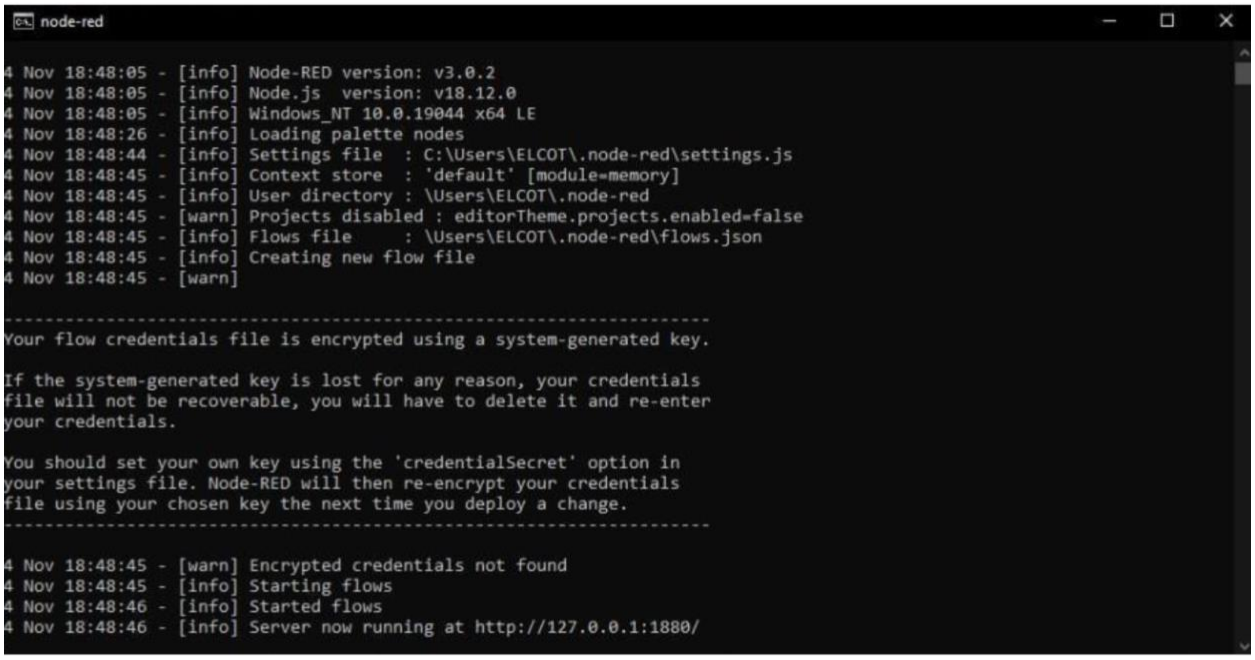


```
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 86, humidity: 31,
soil moisture: 54 }
11/5/2022, 11:20:36 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 8, humidity: 64,
soil moisture: 59 }
11/5/2022, 11:20:39 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 98, humidity: 96,
soil moisture: 53 }
11/5/2022, 11:20:44 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 96, humidity: 35,
soil moisture: 25 }
11/5/2022, 11:20:50 AM   node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 78, humidity: 1,
soil moisture: 28 }
```



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

```
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 28, humidity: 26,
soil moisture: 75 }
11/5/2022, 11:24:38 AM  node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 2, humidity: 82,
soil moisture: 53 }
11/5/2022, 11:24:44 AM  node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 48, humidity: 95,
soil moisture: 82 }
11/5/2022, 11:24:50 AM  node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 33, humidity: 40,
soil moisture: 90 }
11/5/2022, 11:24:56 AM  node: debug 1
iot-
2/type/PNT2022TMID47477/id/PNT2022TMID47477/evt
/event_1/fmt/json : msg.payload : Object
▸{ temperature: 43, humidity: 2,
soil moisture: 86 }
```



```
4 Nov 18:48:05 - [info] Node-RED version: v3.0.2
4 Nov 18:48:05 - [info] Node.js  version: v18.12.0
4 Nov 18:48:05 - [info] Windows_NT 10.0.19044 x64 LE
4 Nov 18:48:26 - [info] Loading palette nodes
4 Nov 18:48:44 - [info] Settings file  : C:\Users\ELCOT\.node-red\settings.js
4 Nov 18:48:45 - [info] Context store  : 'default' [module=memory]
4 Nov 18:48:45 - [info] User directory : \Users\ELCOT\.node-red
4 Nov 18:48:45 - [warn] Projects disabled : editorTheme.projects.enabled=false
4 Nov 18:48:45 - [info] Flows file     : \Users\ELCOT\.node-red\flows.json
4 Nov 18:48:45 - [info] Creating new flow file
4 Nov 18:48:45 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

4 Nov 18:48:45 - [warn] Encrypted credentials not found
4 Nov 18:48:45 - [info] Starting flows
4 Nov 18:48:46 - [info] Started flows
4 Nov 18:48:46 - [info] Server now running at http://127.0.0.1:1880/
```

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 achieving better crop yields thus leading to the economic 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 starving. 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, Watson simulator, 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 animal land fire can be detected by cameras and if it comes towards farm then system will be directly activated through wireless networks. Wild animals can also be detected by using wireless networks such as laser wireless sensors and by sensing this laser or sensor’s security system will be activated.

APPENDIX

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(commamd=="lighton"):
        print('lighton')
    elif(command=="lightoff"):
        print('lightoff')
    elif(command=="motoron"):
        print('motoron')
    elif(command=="motoroff"):
        print('motoroff')

myConfig = {
    "identity": {
        "orgId": "lkagkz",
        "typeId": "lav15",
        "deviceId": "141414"
    },
    "auth": {
        "token": "8883419179"
    }
}

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()

   #This is the model ID of a publicly available General model. You may use any other public or custom
model ID.
   request = service_pb2.PostModelOutputsRequest(
      model_id='a6100c6f4fb74e79ad8b57b1db2f0235',

inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes)
)
      )])
   response = stub.PostModelOutputs(request, metadata=metadata)
   print(response)
   if response.status.code != status_code_pb2.SUCCESS:
      raise Exception("Request failed, status code: " + str(response.status.code))
   detect=False
   for concept in response.outputs[0].data.concepts:
      #print('%12s: %.f' % (concept.name, concept.value))
      if(concept.value>0.98):
         #print(concept.name)
         if(concept.name=="animal"):
            print("Alert! Alert! animal detected")
            playsound.playsound('alert.mp3')
            picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
            cv2.inwrite(picname+'.jpg',frame)
            multi_part_upload('Jade', picname+'.jpg', picname+'.jpg')
            json_document={"link":COS_ENDPOINT+'/'+'Jade'+'/'+picname+'.jpg'}
            new_document = my_database.create_document(json_document)

            if new_document.exists():
               print(f"Document successfully created.")
            time.sleep(5)
            detect=True

   moist=random.randint(0,100)
```

```
    humidity=random.randint(0,100)
    myData={'Animal':detect,'moisture':moist,'humidity':humidity}
    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()
```