**IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE**

**Category: INTERNET OF THINGS**

A PROJECT REPORT

***Submitted by***

## DHAKSHESH V

## GOKUL M

## LINGARAJA P

**BALASANKARI R**

***FROM***

**SRI VRNKATESWARAA COLLEGE OF TECHNOLOGY**

***In fulfillment of project in IBM-NALAYATHIRAN 2022 Team Id: PNT2022TMID38273***

## PROJECT GUIDES

**Industry Mentor: Mr. Dinesh**

#### Faculty Mentor: Mrs. Anburaman S

**INDEX**

1. **INTRODUCTION**
   1. **Project Overview**
   2. **Purpose 2.LITERATURE SURVEY**
   3. **Existing problem**
   4. **Problem Statement Definition 3. IDEATION & PROPOSEDSOLUTION**
   5. **Empathy Map Canvas**
   6. **Ideation & Brainstorming**
   7. **Proposed Solution**
   8. **Problem Solution fit 4.REQUIREMENT ANALYSIS**
   9. **Functional requirement**
   10. **Non-Functional requirements 5.PROJECT DESIGN**
   11. **Data Flow Diagrams**
   12. **Solution & Technical Architecture**
   13. **User Stories**
2. **PROJECT PLANNING & SCHEDULING**
   1. **Sprint Planning & Estimation**
   2. **Sprint Delivery Schedule**
3. **CODING & SOLUTION**
   1. **Feature 1**
4. **CONCLUSION**
5. **FUTURE SCOPE**
6. **APPENDIX**
   1. **Source Code**

# INTRODUCTION

## Project Overview:

#### Smart crop protection system

Smart crop protection system solutions use sensors placed in crop yields to measure humidity , temperarture , moisture and to notify farmers when crops are ready to be emptied. Over time, historical data collected by sensors can be used to identify crop patterns . The cost of these sensors is steadily decreasing, making IoT crop protection more feasible to implement and more attractive to farmer.

## Purpose:

* At present, we can see crop are being damaged due to many reasons. Our primary goal is to protect the crop from being damaged .
* Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield.
* In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops .
* So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

# LITERATURE SURVEY

## Existing Problem:

* At present, we can see crop are being damaged due to many reasons. Our primary goal is to protect the crop from being damaged.
* Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield.
* In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops
* So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

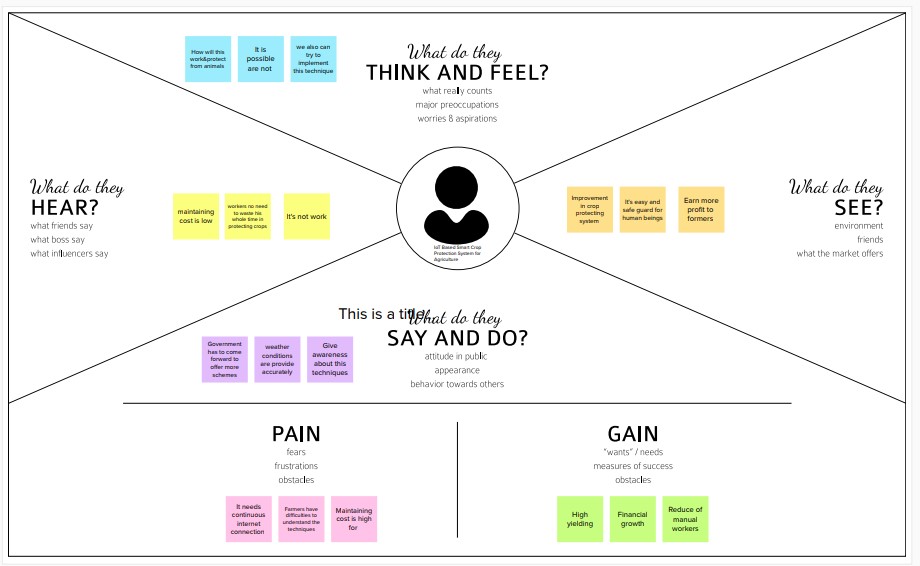
## Problem Statement Definition:

* Due to damage in crops, many farmers left farming and started doing other jobs because of loss they faced in agriculture. So our crop protection should prevent crop from being damaged and produce better yield .In agriculture fields crops are being damaged by birds, animals, insects, climate, disease, excess water, etc. Our crop protection system should stop these from damaging the crops .So, our problem statement is to design a system based on IOT application for protecting crops from birds, animals, insects, climate, disease, excess water, etc and provide high yield in agriculture to make farmers happy and people enjoy the healthy food.

## IDEATION & PROPOSED SOLUTION

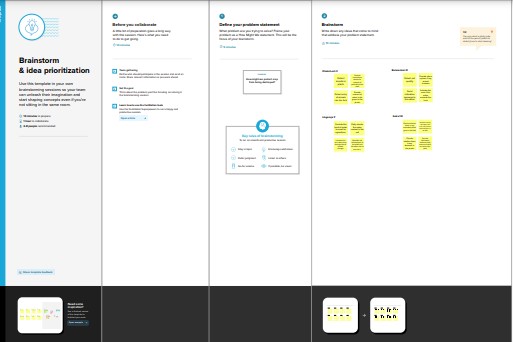
* 1. **Empathy Map Canvas:**

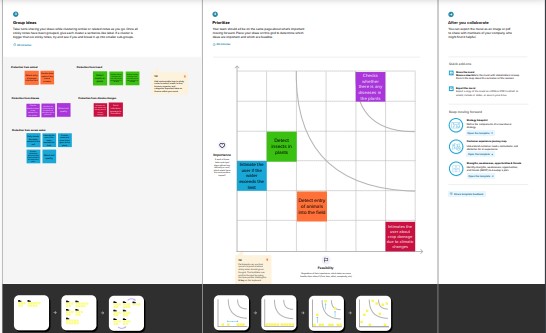
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

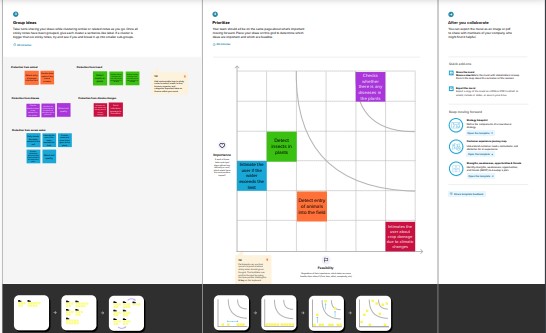


## Ideation & Brainstorming:

Ideation and Brainstorming Ideation is often closely related to the practice of brainstorming, a**specific technique that is utilized to generate new ideas**. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.





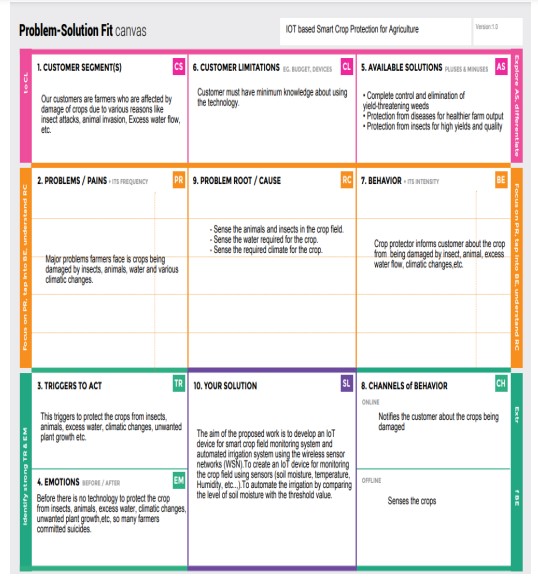


## Proposed Solution:

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE |
| 2. | Idea / Solution description | The aim of the proposed work is to develop an IOT device for smart crop field monitoring system and automated irrigation system using the wireless sensor networks(WSN) . To create an IOT device for monitoring the crop field using sensors (soil moisture ,temperature ,Humidity ,etc.,) To automate the irrigation by comparing the level of soil moisture with the threshold value . |
| 3. | Novelty / Uniqueness | Daily update about the  Condition of the land send to the farmers via mail |
| 4. | Social Impact / Customer Satisfaction | * Cost effective to the society * Modernaization to the society * High protection and High yield |

|  |  |  |
| --- | --- | --- |
| 5. | Business Model (Revenue Model) | Outcome based model Data based model Platform based model |
| 6. | Scalability of the Solution | Start small and build out |

#### PROBLEM SOLUTION FIT:

****

**4 REQUIREMENT ANALYSIS**

* 1. ***Functional Requirements:***

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement**  **(Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Form Registration through Gmail . |
| FR-2 | User Confirmation | Confirmation via Email . |
| FR-3 | Interfacing with hardware | Interface the sensors with the software application so as to alert the farmers in case of any harm for crops . |
| FR-4 | Database Connection | Databases are retrieved from IBM Cloudant . |
| FR-5 | Mobile Application | Alarm and motors can be accessed from the mobile app . |

### Non-functional Requirements:

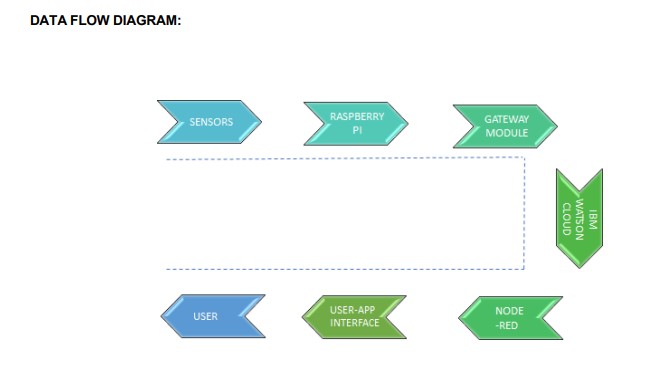
Following are the non-functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The smart crop protection alerts the farmers in caseof any obstacles and helps in protecting the crops |
| NFR-2 | **Security** | Smart Agriculture can improve the farming practicesand maintain sustainable production of crops  especially by preventing the animals into the agricultural lands through IoT enabled devices |
| NFR-3 | **Reliability** | With a proper power supply, SD card and programming the processor should be able to run 24/7 for years. The SD card and power supply will likely wear out faster than the Pi. The possible reasons behind Raspberry Pi failure can be power  breakdowns, SD card failures, and ineligible environments. |
| NFR-4 | **Performance** | Usage of an SD card module that helps to store a specified sound to scare the animals.  Crop damage due to animal attack can be sensed. Network and Design Evaluation |
| NFR-5 | **Availability** | Agriculture for different variety of crops is based on the monsoon changes, indoor and outdoor climatic temperatures, availability of rainfall and irrigation  methods. |
| NFR-6 | **Scalability** | The product shall be made available to everyone especially in remote areas for better efficiency of crop yield with the better safety of crops as well as the farmers. |

## 5 PROJECT DESIGN

* 1. **Data Flow Diagrams:**

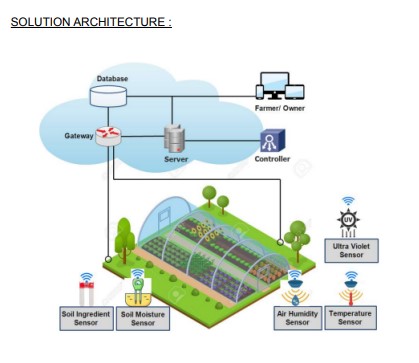
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



# SOLUTION AND TECHNICAL ARCHITECTURE

* + 1. **Summary:**

The smart crops are constructed based on the sensor application and raspberry pi . It can also act as a transceiver since it is connected to the mobile phone of the user . The overall process of the sensors and raspberry pi is monitored using real time monitor which can help data transmission.This is stored and formulated using cloud data . Through which the admin can access the data and then track the location from GPS .



# Components & Technologies:

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Component | Description | Technology |
| 1. | User Interface | How user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript / Angular Js / React Js etc. |
| 2. | Application Logic-1 | Logic for a process in the application | Java / Python |
|  | Database | Data Type, Configurations etc. | MySQL, NoSQL, etc |
| 4. | IoT | To collect the data and alert the users | IBM Watson IoT Platform, Node Red. |
| 5. | Cloud Database | Database Service on Cloud | Cloudant DB |

* + 1. **Application Characteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology |
| 1. | Open Source framework | List the open-source frameworks used | Technology of Opensource framework |
| 2. | Security implementation | List all the security / access controls implemented, use of firewalls etc ., | e.g. SHA-256, Encryptions, IAM Controls, OWASP etc. |
| 3. | Scalable Architecture | Justify the scalability of architecture (3  – tier, Micro-services) | Technology used |
| 4. | Availability | Justify the availability of application (e.g. use of load balancers, distributedservers etc.) | Technology used |
| 5. | Performance | Design consideration for the performance of the application (number of requests per sec, use of  Cache, use of CDN’s) etc. | Technology used |

## 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(Farmer) | Maintain fields | USN-1 | As a user, I can monitor the growth of  crops and protect the crops against animals | I can maintain the fields with less labor | High | Sprint-1 |
|  | Analyzing problem | USN-2 | As a user, I collect the required  information about the problems on agriculture fields | I can ask my field owner directly. | low | Sprint-2 |
| Project Designers | Identifying the problem and  provide solutions | USN-3 | As a user, I can sense the water level and flame in the field using sensor and monitor using IOT | I can perform this actions via IoT. | Medium | Sprint-1 |
| Customer field Maintainer | Problem solution | USN-4 | As a user, areas can be monitored from a remote place | Checking Process | Medium | Sprint-3 |
|  | Final process | USN-5 | This proposed smart IOT-based crop protection device is found to be  cost-effective and efficient | I can take necessary action if required | High | Sprint-4 |

# PROJECT PLANNING AND SCHEDULING

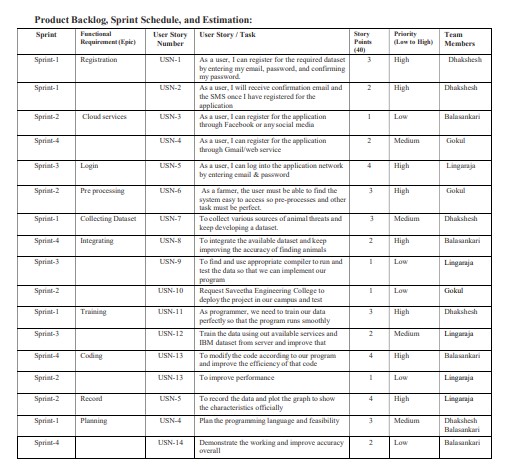
* 1. **SPRINT PLANNING & SCHEDULING:**

|  |  |  |
| --- | --- | --- |
| **TITLE** | **DESCRIPTION** | **DATE** |
| **Literature Survey & Information Gathering** | Literature survey on  the selected project is done by gathering information about related details on technical papers and  web browsing. | 28 SEPTEMBER  2022 |
| **Prepare Empathy Map** | Prepared Empathy Map  Canvas to capture the user Pains & Gains list of problem statements. | 24 SEPTEMBER  2022 |
| **Ideation** | List the organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility and importance. | 25 SEPTEMBER  2022 |
| **Proposed Solution** | Prepared the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc. | 23 SEPTEMBER  2022 |
| **Problem Solution Fit** | Prepared problem - solution ﬁt document. | 30 SEPTEMBER  2022 |
| **Solution Architecture** | Prepare solution architecture document. | 28 SEPTEMBER  2022 |

* 1. **SPRINT DELIVERY SCHEDULE**

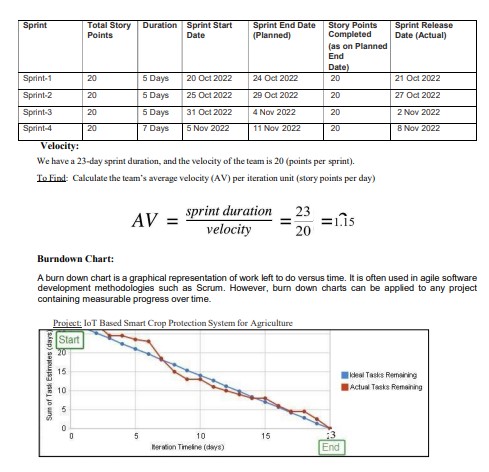
#### Product Backlog, Sprint Schedule, and Estimation

Usethe below template to create product backlog and sprint schedule.

****

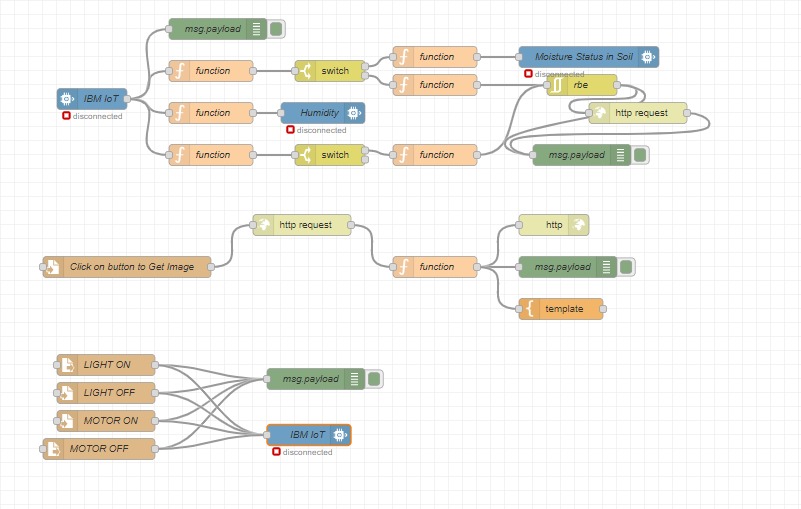
**Project Tracker, Velocity & Burndown Charts**

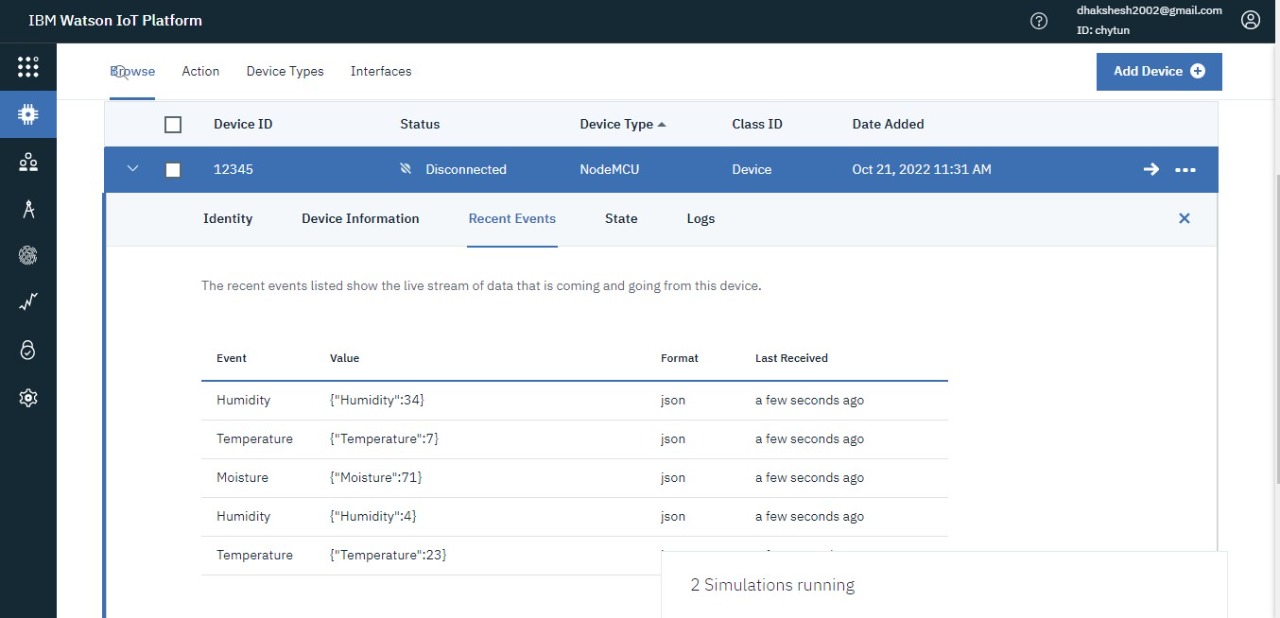
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Durati on** | **Sprint Start Date** | **Sprint End Date (Planned)** | **Story Points Completed (as on Planned**  **End Date)** | **Sprint Release Date (Actual)** |
| Sprint-1 | 20 | 5 Days | 20 Oct 2022 | 24 Oct 2022 | 20 | 21 Oct 2022 |
| Sprint-2 | 20 | 5 Days | 25 Oct 2022 | 29 Oct 2022 | 20 | 27 Oct 2022 |
| Sprint-3 | 20 | 5 Days | 31 Oct  2022 | 4 Nov 2022 | 20 | 02 Nov 2022 |
| Sprint-4 | 20 | 7 Days | 5 Nov  2022 | 11 Nov 2022 | 20 | 08Nov 2022 |



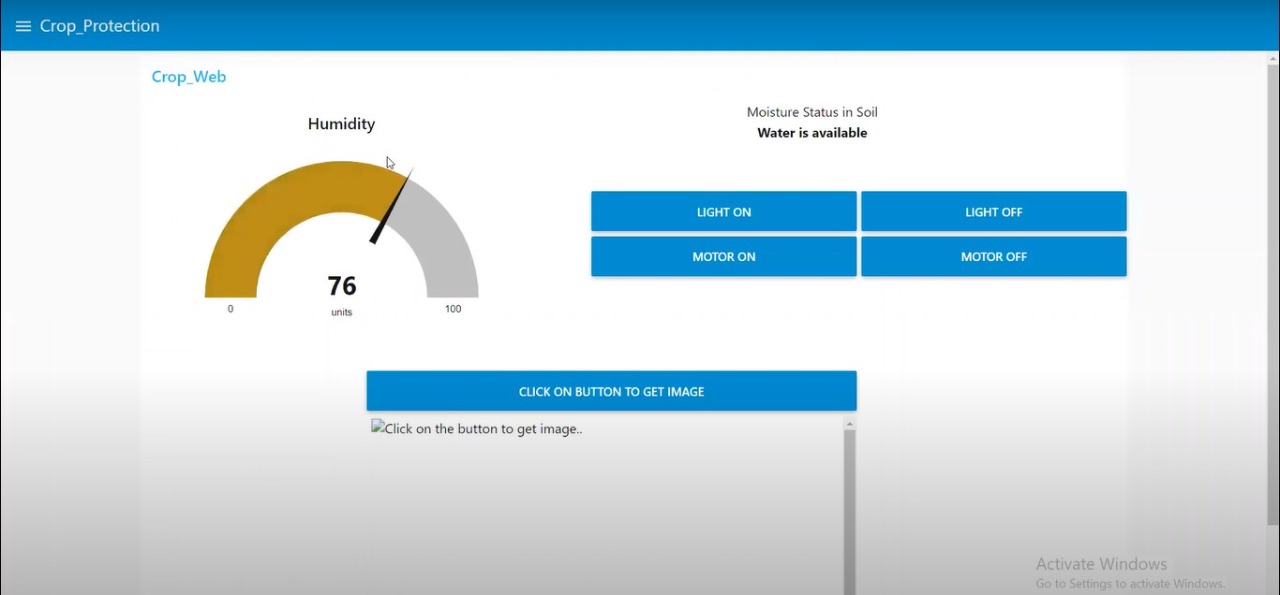
## CODING AND SOLUTIONING

**NODE RED SERVICE ASSOCIATED WITH IBM CLOUD:**





**Node red Dashboard:**



# CONCLUSION

We presented an intelligent Smart crop protection system. The

system is based on IoT sensors. It is responsible for measuring the waste level in the smart crop. When the smart crop gets affected almost there will be information received by the admin, Since the admin can access the data and location of the crop. Later send this data (through Internet) to a server for storage and processing.

This data helps to compute the optimized collection routes for the workers. In future, we would like to enhance the system for different kind of crop management system .

# FUTURE SCOPE

The advantage of this work is its contribution in making a Smart crop. Among the many challenges that a city faces, crop protection management is of utmost importance. This is because, it is directly related to food of people living in the area. We are further extending this work to address problems of seggragating different kind of crops (e.g.,paddy ,wheat ,etc., ), and identifying different agricultural department for collecting it. The optimization algorithms may be devised accordingly depending on the requirements. In future, we would like to enhance the system for different kind of crops .

# 11.APPENDIX

* 1. **SOURCE CODE:**

# PYTHON CODE TO PUBLISH DATA

import cv2

import numpy as np

import wiot.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, resource\_pb2

from clarifai\_grpc.grpc.api.status import status\_code\_pb2

#This is how you authenticate

metadata = (('authorization', 'key 0620e202302b4508b90eab7efe7475e4'),)

COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"

COS\_API\_KEY\_ID = "g5d4qO8EIgv4TWUCJj4hfEzgalqEjrDbE82AJDWlAOHo"

COS\_AUTH\_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

COS\_RESOURCE\_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/c2fa2836eaf3434bbc8b5b58fefff3f0:62e450fd-4c82-4153-ba41-ccb53adb8111::"

clientdb = cloudant("apikey-W2njldnwtjO16V53LAVUCqPwc2aHTLmlj1xXvtdGKJBn", "88cc5f47c1a28afbfb8ad16161583f5a", url="https://d6c89f97-cf91-48b7-b14b-c99b2fe27c2f-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": "chytun",

"typeId": "NodeMCU",

"deviceId": "12345"

},

"auth": {

"token": "12345678"

}

}

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

client.connect()

database\_name = "sample"

my\_database = clientdb.create\_database(database\_name)

if my\_dtabase.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 = cv3.cvtColor(frame, cv2.COLOR\_BGR@GRAY)

imS= cv2.resize(frame, (960,540))

cv2.inwrite('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='e9359dbe6ee44dbc8842ebe97247b201',

inputs=[resources\_pb2.Input(data=resources\_pb2.Data(image=resources\_pb2.Image(base64=file\_bytes))

)])

response = stub.PostModelOutputs(request, metadata=metadata)

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('Dhakshesh', picname+'.jpg', picname+'.jpg')

json\_document={"link":COS\_ENDPOINT+'/'+'Dhakshesh'+'/'+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", daya=myData, qos=0, onPublish=None)

print("Publish Ok..")

client.commandCallback = myCommandCallback

cv2.imshow('frame',imS)

if cv2.waitKey(1) & 0xFF == ord('q'):

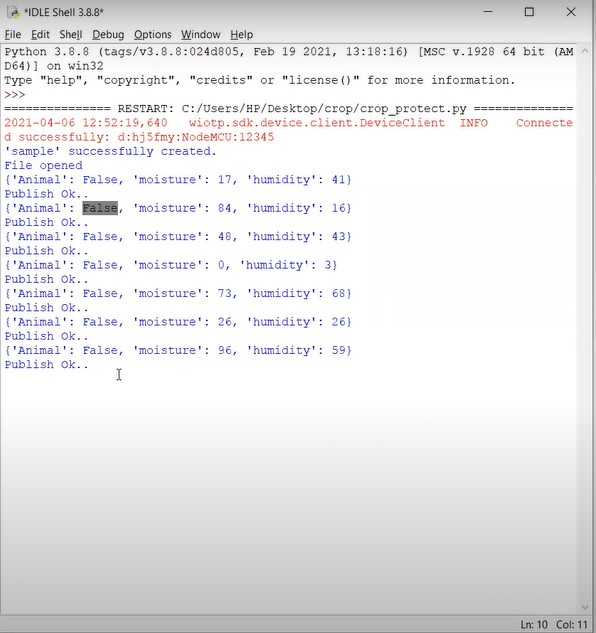
break

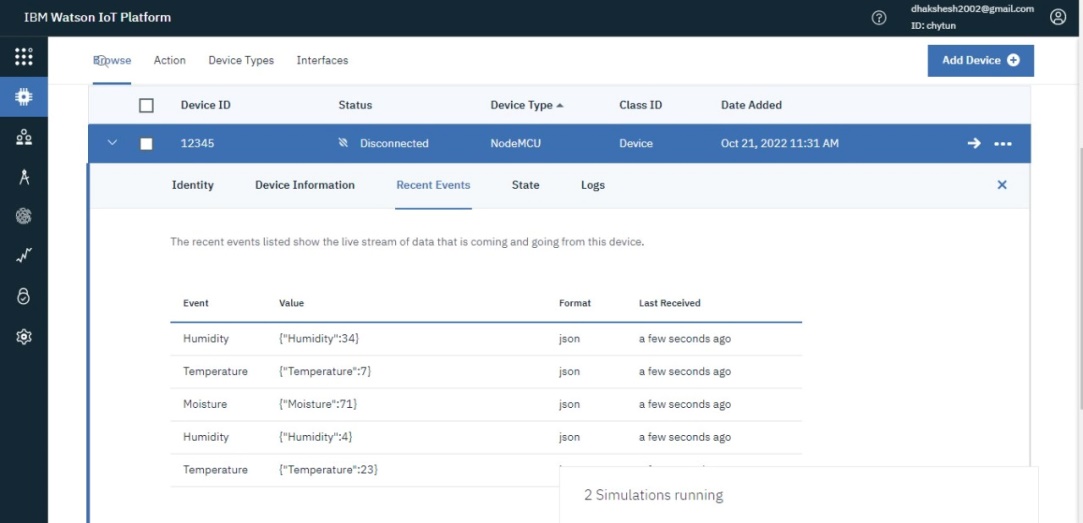
client.disconnect()

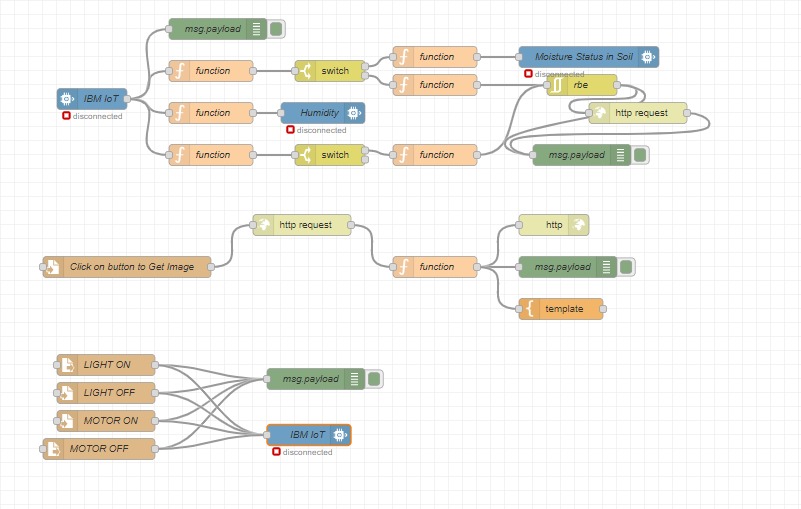
cap.release()

cv2.destroyAllWindows()

# OUTPUT







**TECH TO SPEECH:**

from ibm\_watson import TextToSpeechV1

from ibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

import playsound

authenticator = IAMAuthenticator('v9n8Zn4r5VpcMVz\_HyRY0DrS13jSzph2IEFioVj4-vmT')

text\_to\_speech = TextToSpeechV1(

authenticator=authenticator

)

text\_to\_speech.set\_service\_url('https://api.eu-gb.text-to-speech.watson.cloud.ibm')

with open('alert.mp3', 'wb') as audio\_file:

audio\_file.write(

text\_to\_speech.synthesize(

'Alert! Alert! Animal Detected.',

voice='en-US\_ALLisonV3Voice',

accept='audio/mp3'

).get\_result().content)

playsound.playsound('alert.mp3')

