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

**Category:INTERNETOFTHINGS**

PROJECT REPORT

***Submittedby***

**RAMESH S**

**RAHUL P**

**RAJESH S**

**SIVABALAN M**

***InfulfillmentofprojectinIBM-NALAYATHIRAN2022TeamId:PNT2022TMID38273***

**INDEX**

1. **INTRODUCTION**
   1. **ProjectOverview**
   2. **Purpose2.LITERATURESURVEY**
   3. **Existingproblem**
   4. **Problem Statement Definition3.IDEATION&PROPOSEDSOLUTION**
   5. **EmpathyMapCanvas**
   6. **Ideation&Brainstorming**
   7. **ProposedSolution**
   8. **Problem Solution fit4.REQUIREMENTANALYSIS**
   9. **Functionalrequirement**
   10. **Non-Functional requirements5.PROJECTDESIGN**
   11. **DataFlowDiagrams**
   12. **Solution&TechnicalArchitecture**
   13. **UserStories**
2. **PROJECTPLANNING&SCHEDULING**
   1. **SprintPlanning&Estimation**
   2. **SprintDeliverySchedule**
3. **CODING & SOLUTION**
   1. **Feature1**
4. **CONCLUSION**
5. **FUTURE SCOPE**
6. **APPENDIX**
   1. **SourceCode**

# INTRODUCTION

## ProjectOverview:

#### 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.

# LITERATURESURVEY

## ExistingProblem:

* 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.

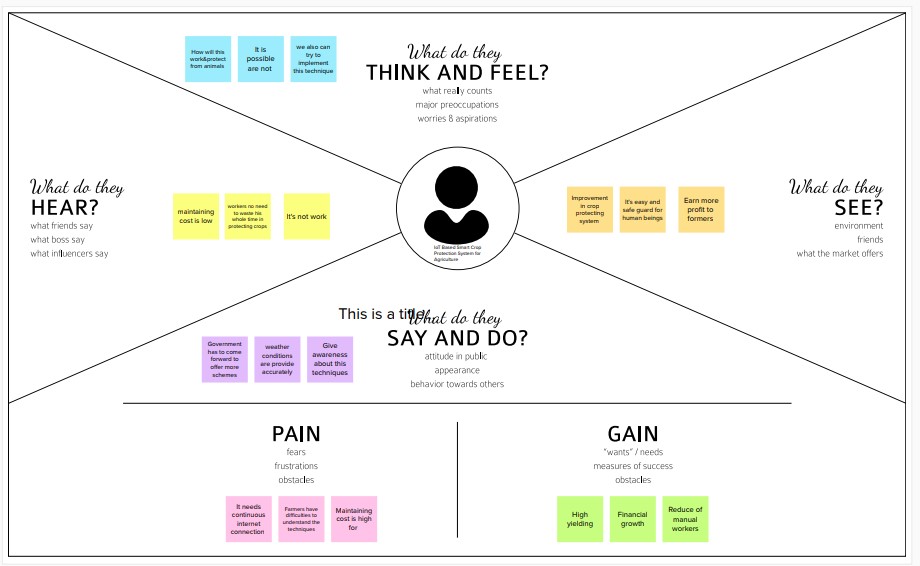
## ProblemStatement Definition:

* 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&PROPOSEDSOLUTION

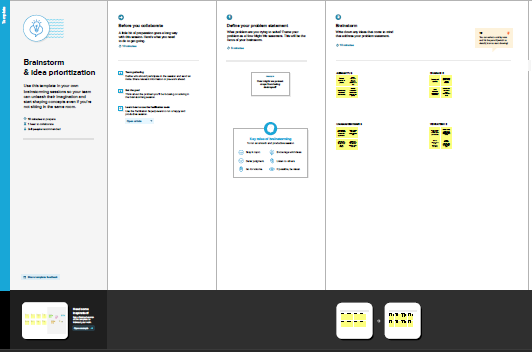
* 1. **EmpathyMapCanvas:**

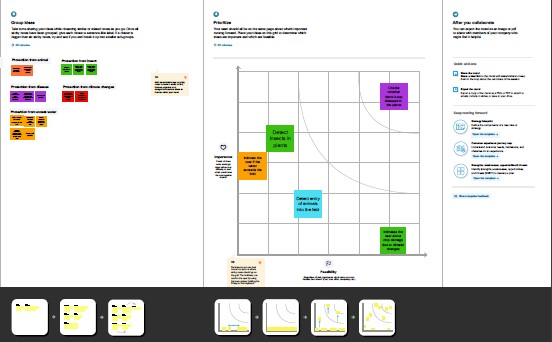
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’sbehavioursandattitudes.Itisausefultooltohelpsteamsbetterunderstandtheirusers.Creating an effective solution requires understanding the true problem and the personwho is experiencing it. The exercise of creating the map helps participants consider thingsfromtheuser’sperspectivealongwithhisorhergoalsandchallenges.



## 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.



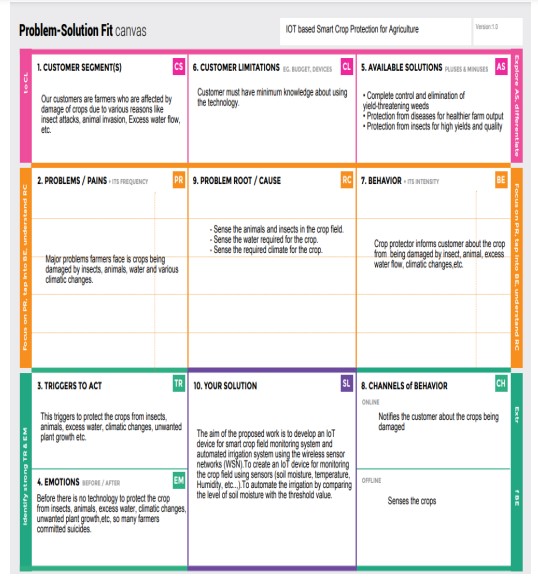


## ProposedSolution:

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | ProblemStatement(Problemtobesolved) | IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE |
| 2. | Idea/Solutiondescription | 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 /CustomerSatisfaction | * Cost effective to the society * Modernaization to the society * High protection and High yield |

|  |  |  |
| --- | --- | --- |
| 5. | Business Model(RevenueModel) | Outcome based model Data based model Platform based model |
| 6. | ScalabilityoftheSolution | Start small and build out |

#### PROBLEMSOLUTION FIT:

****

**4REQUIREMENTANALYSIS**

* 1. ***FunctionalRequirements:***

Followingarethefunctionalrequirementsoftheproposedsolution.

|  |  |  |
| --- | --- | --- |
| **FRNo.** | **FunctionalRequirement**  **(Epic)** | **SubRequirement(Story/Sub-Task)** |
| FR-1 | UserRegistration | Registration through FormRegistrationthroughGmail . |
| FR-2 | UserConfirmation | ConfirmationviaEmail . |
| FR-3 | Interfacingwithhardware | Interfacethesensorswiththesoftwareapplicationsoastoalertthefarmersincaseofanyharmforcrops . |
| FR-4 | DatabaseConnection | DatabasesareretrievedfromIBMCloudant . |
| FR-5 | MobileApplication | Alarmandmotorscanbeaccessedfromthemobileapp . |

### Non-functionalRequirements:

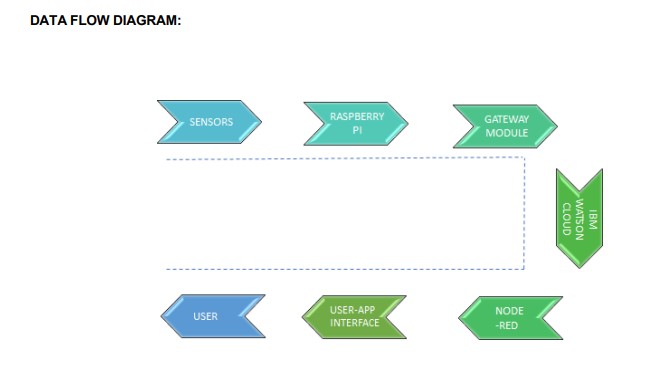
Followingarethenon-functionalrequirementsoftheproposedsolution.

|  |  |  |
| --- | --- | --- |
| **FRNo.** | **Non-FunctionalRequirement** | **Description** |
| NFR-1 | **Usability** | Thesmartcropprotectionalertsthefarmersincaseofanyobstaclesandhelpsinprotecting thecrops |
| NFR-2 | **Security** | SmartAgriculturecanimprovethefarmingpracticesandmaintainsustainableproductionofcrops  especially by preventing the animals into theagriculturallandsthroughIoTenableddevices |
| NFR-3 | **Reliability** | With a proper power supply, SD card andprogrammingtheprocessorshouldbeabletorun24/7 for years. The SD card and power supply willlikely wear out faster than the Pi. The possiblereasonsbehindRaspberryPifailurecanbepower  breakdowns,SDcardfailures,andineligibleenvironments. |
| NFR-4 | **Performance** | UsageofanSDcardmodulethathelpstostoreaspecifiedsoundtoscaretheanimals.  Cropdamageduetoanimalattackcanbesensed.NetworkandDesignEvaluation |
| NFR-5 | **Availability** | Agriculturefordifferentvarietyofcropsisbasedonthe monsoon changes, indoor and outdoor climatictemperatures,availabilityofrainfallandirrigation  methods. |
| NFR-6 | **Scalability** | The product shall be made available to everyoneespecially in remote areas for better efficiency ofcropyieldwiththebettersafetyofcropsaswell asthefarmers. |

## 5PROJECTDESIGN

* 1. **DataFlowDiagrams:**

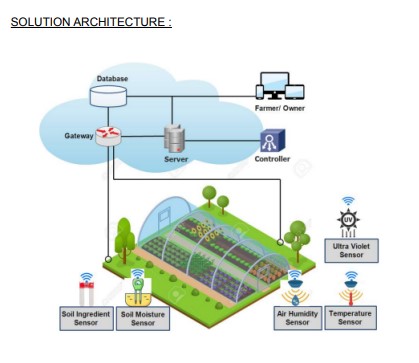
A Data Flow Diagram (DFD) is a traditional visual representation of the informationflows within a system. A neat and clear DFD can depict the right amount of the systemrequirement graphically. It shows how data enters and leaves the system, what changes theinformation,andwheredataisstored.



# SOLUTIONANDTECHNICALARCHITECTURE

* + 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. | UserInterface | Howuserinteractswithapplicatione.g.  WebUI,MobileApp, Chatbotetc. | HTML,CSS,JavaScript/AngularJs/ReactJsetc. |
| 2. | Application Logic-1 | Logicforaprocessintheapplication | Java/Python |
|  | Database | DataType,Configurationsetc. | MySQL,NoSQL,etc |
| 4. | IoT | To collect the data and alert the users | IBM Watson IoT Platform, Node Red. |
| 5. | Cloud Database | Database ServiceonCloud | Cloudant DB |

* + 1. **ApplicationCharacteristics:**

|  |  |  |  |
| --- | --- | --- | --- |
| S.No | Characteristics | Description | Technology |
| 1. | Open Source framework | Listtheopen-sourceframeworksused | TechnologyofOpensourceframework |
| 2. | Security implementation | Listallthesecurity/accesscontrolsimplemented,useoffirewallsetc ., | e.g. SHA-256, Encryptions,IAMControls,OWASPetc. |
| 3. | Scalable Architecture | Justifythescalabilityofarchitecture(3  –tier,Micro-services) | Technologyused |
| 4. | Availability | Justifytheavailabilityofapplication(e.g.use of load balancers, distributedserversetc.) | Technologyused |
| 5. | Performance | Design consideration for theperformance of the application(numberofrequestspersec,useof  Cache,useofCDN’s)etc. | Technologyused |

## UserStories

Usethebelowtemplatetolistalltheuserstoriesfortheproduct.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **UserType** | **FunctionalRequirement(Epic)** | **UserStory**  **Number** | **UserStory/Task** | **Acceptancecriteria** | **Priority** | **Release** |
| Customer(Farmer) | Maintain fields | USN-1 | Asauser,Icanmonitorthegrowthof  cropsandprotectthecropsagainstanimals | Icanmaintainthefieldswithlesslabor | High | Sprint-1 |
|  | Analyzing problem | USN-2 | As auser,Icollecttherequired  informationabouttheproblemsonagriculturefields | Icanaskmyfieldownerdirectly. | low | Sprint-2 |
| ProjectDesigners | Identifying theproblemand  providesolutions | USN-3 | Asauser,Icansensethewaterlevelandflame in the field using sensor andmonitorusingIOT | IcanperformthisactionsviaIoT. | Medium | Sprint-1 |
| Customer field Maintainer | Problem solution | USN-4 | Asauser,areascanbemonitoredfromaremoteplace | CheckingProcess | Medium | Sprint-3 |
|  | Final process | USN-5 | ThisproposedsmartIOT-basedcropprotectiondeviceisfoundtobe  cost-effectiveandefficient | Icantakenecessaryactionifrequired | High | Sprint-4 |

# PROJECTPLANNINGANDSCHEDULING

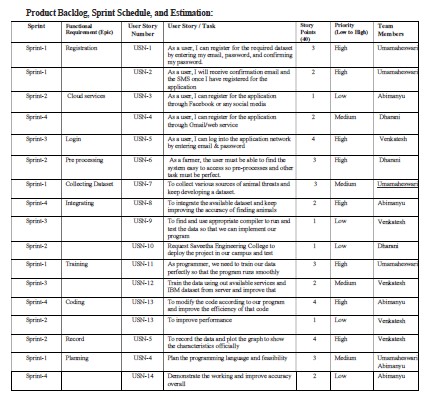
* 1. **SPRINTPLANNING&SCHEDULING:**

|  |  |  |
| --- | --- | --- |
| **TITLE** | **DESCRIPTION** | **DATE** |
| **LiteratureSurvey &InformationGathering** | Literature survey on  the selected project isdonebygatheringinformationaboutrelateddetailsontechnical papers and  webbrowsing. | 28 SEPTEMBER  2022 |
| **Prepare EmpathyMap** | PreparedEmpathy Map  Canvastocapture 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 |
| **ProposedSolution** | Preparedtheproposedsolution document, whichincludesthenovelty,feasibilityofidea,businessmodel,socialimpact,scalabilityofsolution,etc. | 23 SEPTEMBER  2022 |
| **ProblemSolutionFit** | Prepared problem -solutionﬁtdocument. | 30 SEPTEMBER  2022 |
| **Solution Architecture** | Prepare solution architecture document. | 28 SEPTEMBER  2022 |

* 1. **SPRINTDELIVERYSCHEDULE**

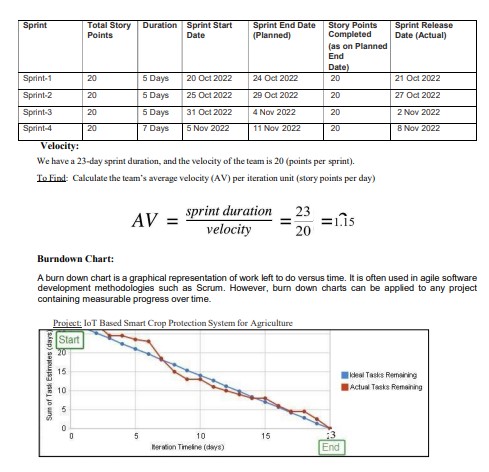
#### ProductBacklog,SprintSchedule,andEstimation

Usethe below template to create product backlog and sprint schedule.

****

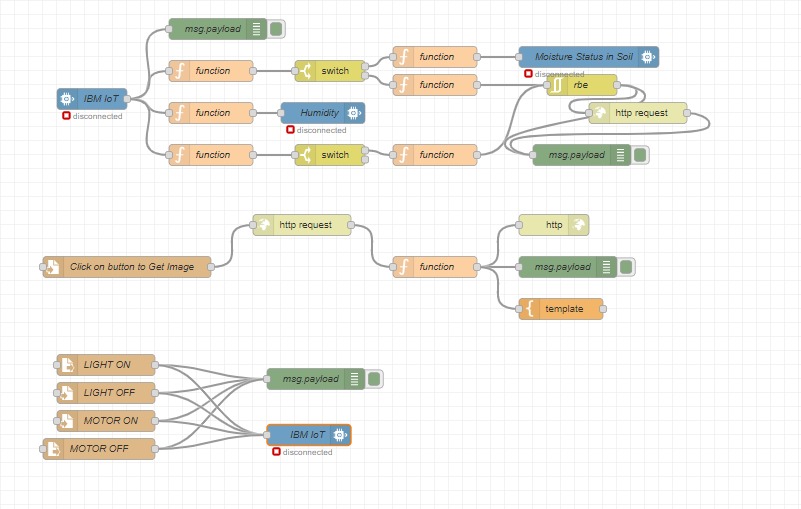
**ProjectTracker,Velocity&BurndownCharts**

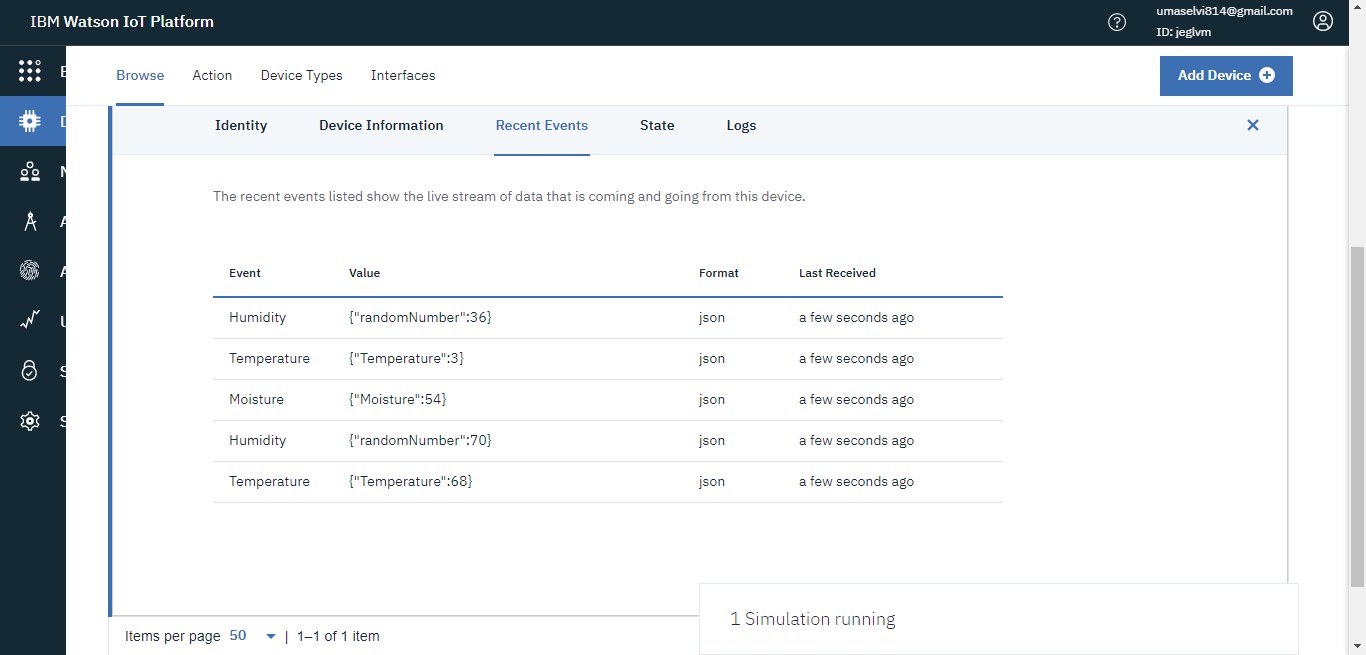
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **TotalStoryPoints** | **Duration** | **SprintStartDate** | **SprintEndDate(Planned)** | **StoryPointsCompleted(asonPlanned**  **EndDate)** | **SprintReleaseDate(Actual)** |
| Sprint-1 | 20 | 5Days | 20Oct2022 | 24Oct2022 | 20 | 21Oct2022 |
| Sprint-2 | 20 | 5Days | 25Oct2022 | 29Oct2022 | 20 | 27Oct2022 |
| Sprint-3 | 20 | 5Days | 31Oct  2022 | 4Nov2022 | 20 | 02Nov2022 |
| Sprint-4 | 20 | 7Days | 5Nov  2022 | 11Nov2022 | 20 | 08Nov2022 |



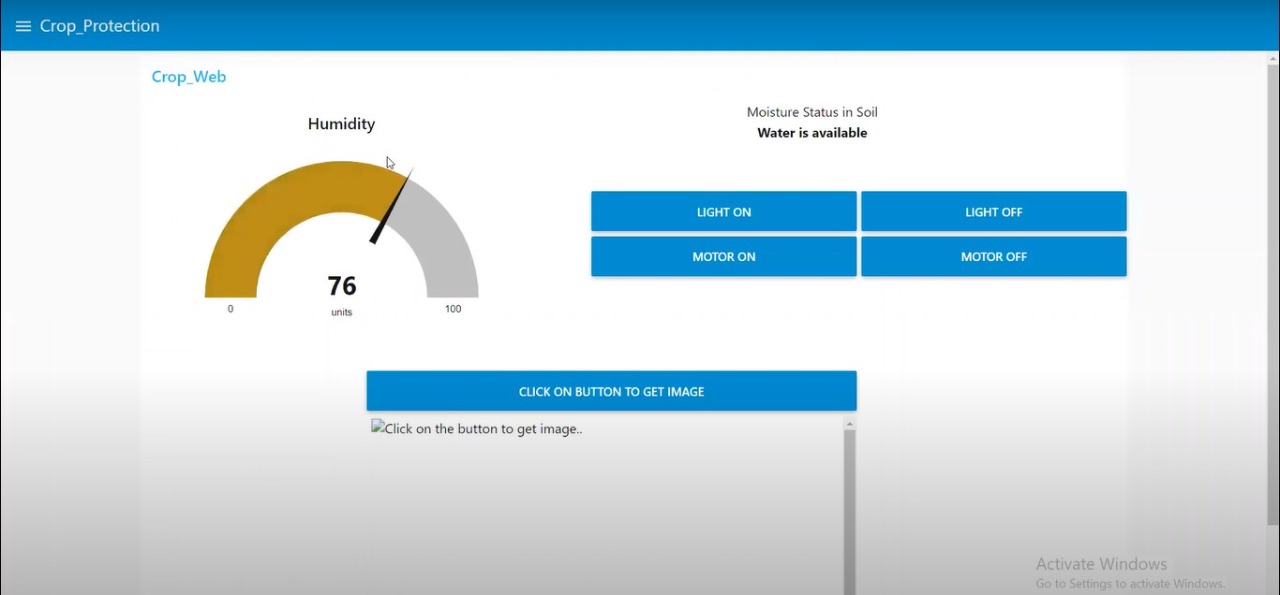
## CODINGANDSOLUTIONING

**NODEREDSERVICEASSOCIATEDWITHIBMCLOUD:**





**NoderedDashboard:**



# CONCLUSION

We presented an intelligent Smart crop protection system. The

system is based on IoT sensors. It is responsible formeasuring 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 thisdata (through Internet) to a server for storage and processing.

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

# FUTURESCOPE

The advantage of thiswork is its contribution in making a Smart crop. Among themany challenges that a city faces, crop protection management is ofutmost importance. This is because, it is directly related tofood of people living in the area. We are further extendingthis work to address problems of seggragating different kindof crops (e.g.,paddy ,wheat ,etc., ), and identifying differentagricultural department for collecting it. The optimization algorithms may bedevised accordingly depending on the requirements.In future, we would like to enhance the systemfor different kind of crops .

# 11.APPENDIX

* 1. **SOURCECODE:**

# PYTHONCODETOPUBLISHDATA

import cv2

importnumpy as np

importwiot.sdk.device

importplaysound

import random

import time

importdatetime

import ibm\_boto3

fromibm\_botocore.client import Config, ClientError

#CloudantDB

fromcloudant.client import Cloudant

fromcloudant.error import CloudantException

fromcloudant.result import Result, ResultByKey

fromclarifai\_grpc.channel.clarifai\_channel import ClarifaiChannel

fromclarifai\_grpc.grpc.api import service\_pb2\_grpc

stub = service\_pb2\_grpc.V2Stub(clarifaiChannel.get.grpc\_channel())

fromclarifai\_grpc.grpc.api import service\_pb2, resource\_pb2

fromclarifai\_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))

exceptClientError as be:

print("CLIENT ERROR: {0}\n".format(be))

except Exception as e:

print("Unable to complete multi-part upload: {0}".format(e))

defmyCommandCallback(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)

ifmy\_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)

ifresponse.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)

ifnew\_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'):

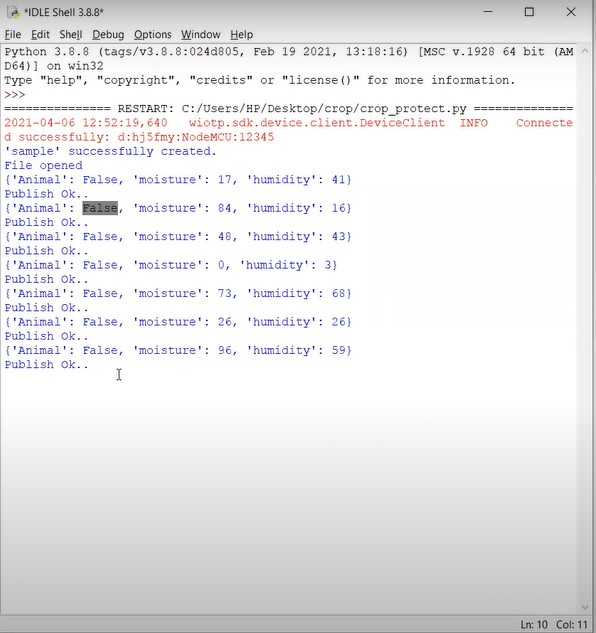
break

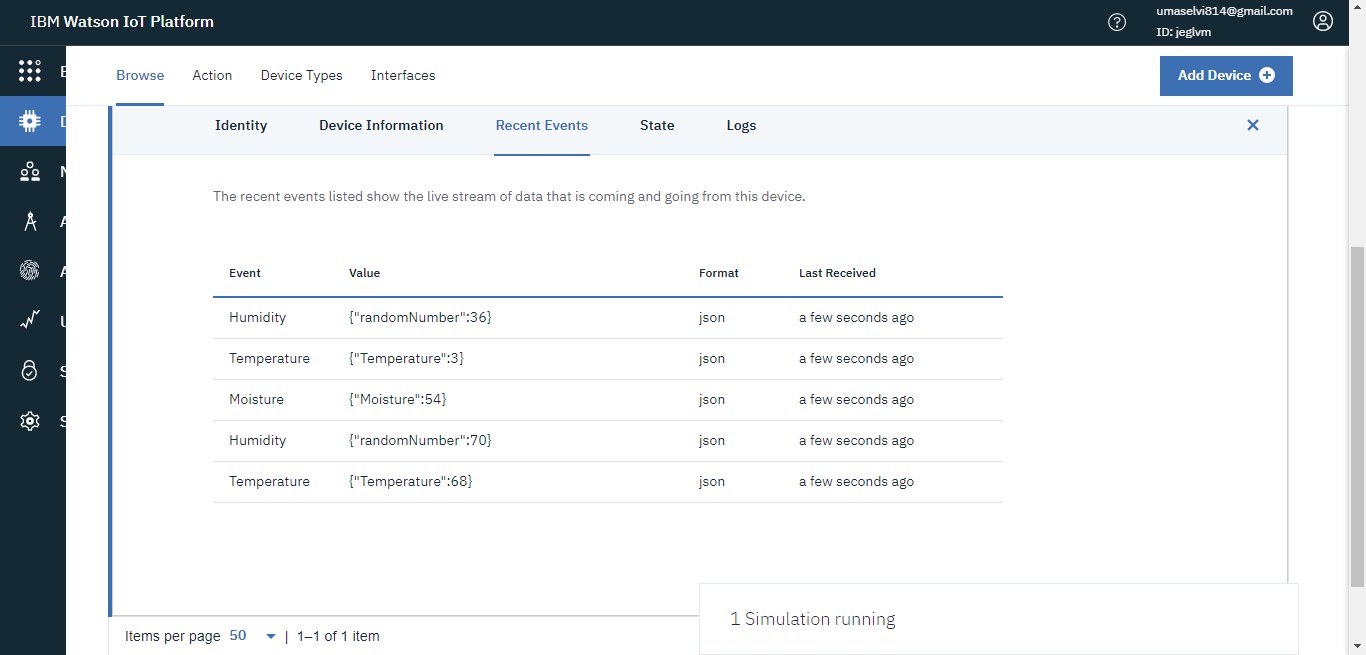
client.disconnect()

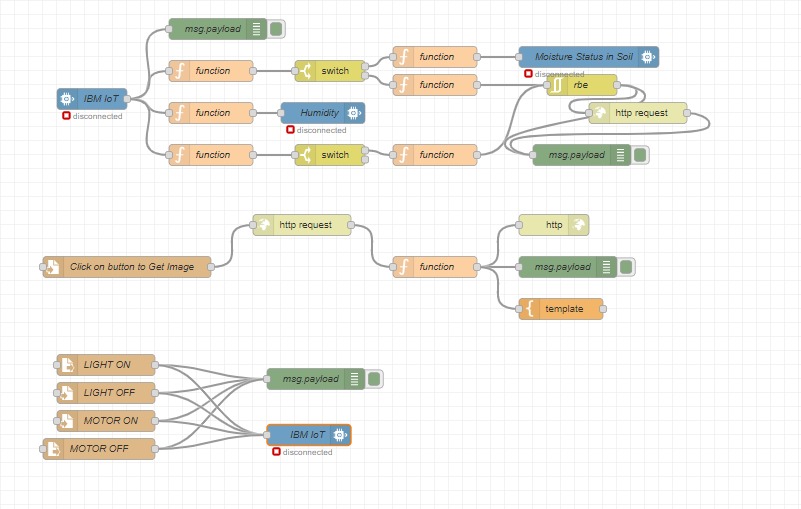
cap.release()

cv2.destroyAllWindows()

# OUTPUT







**TECH TO SPEECH:**

fromibm\_watson import TextToSpeechV1

fromibm\_cloud\_sdk\_core.authenticators import IAMAuthenticator

importplaysound

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')

