

## **1. INTRODUCTION**

### **1.1 Project Overview**

The title of our project is “**IOT based Smart Crop Protection**”. The Overview of our Project is to Safeguard the farm from climatic changes like soil erosion, landslide, and birds, animals etc.,.So, that we are making a IOT based project to protect the farm from climatic changes and haunting animals. For that we are just making a cloud based project and placing IOT based sensor. over which it will produce sounds and notification and provide results on IOT- MIT app. From which we can protect our farm. And it will provide better yield for us.

### **1.2 Purpose**

- ⇒ The main purpose of our project is to protect the farm from climatic changes animals, birds , pests.
- ⇒ And to make the crop to grow better and provide better yield.

## **2. LITERATURE SURVEY**

### **2.1 Existing problem**

- ⇒ “Food” is the important thing, which is needed for everyone to survive in this world. For that farmers are doing their own part in a effective manner, during which they have to face some problems such as:
- ⇒ There are increasing pressures from climate change, soil erosion and biodiversity loss and from consumers’ changing tastes in food and concerns about how it is produced.
- ⇒ And the natural world that farming works with – plants, pests and diseases – continue to pose their own challenges beyond that, they have to
- ⇒ Stay resilient against global economic factors.

- ⇒ Inspire young people to stay in rural areas and become future farmers
- ⇒ The effects of climate change affect farmers’ ability to grow the food we all need.

Increasingly volatile weather and more extreme events – like floods and droughts – change growing seasons, limit the availability of water, allow weeds, pests and fungi to thrive, and can reduce crop productivity.

## **2.2 References**

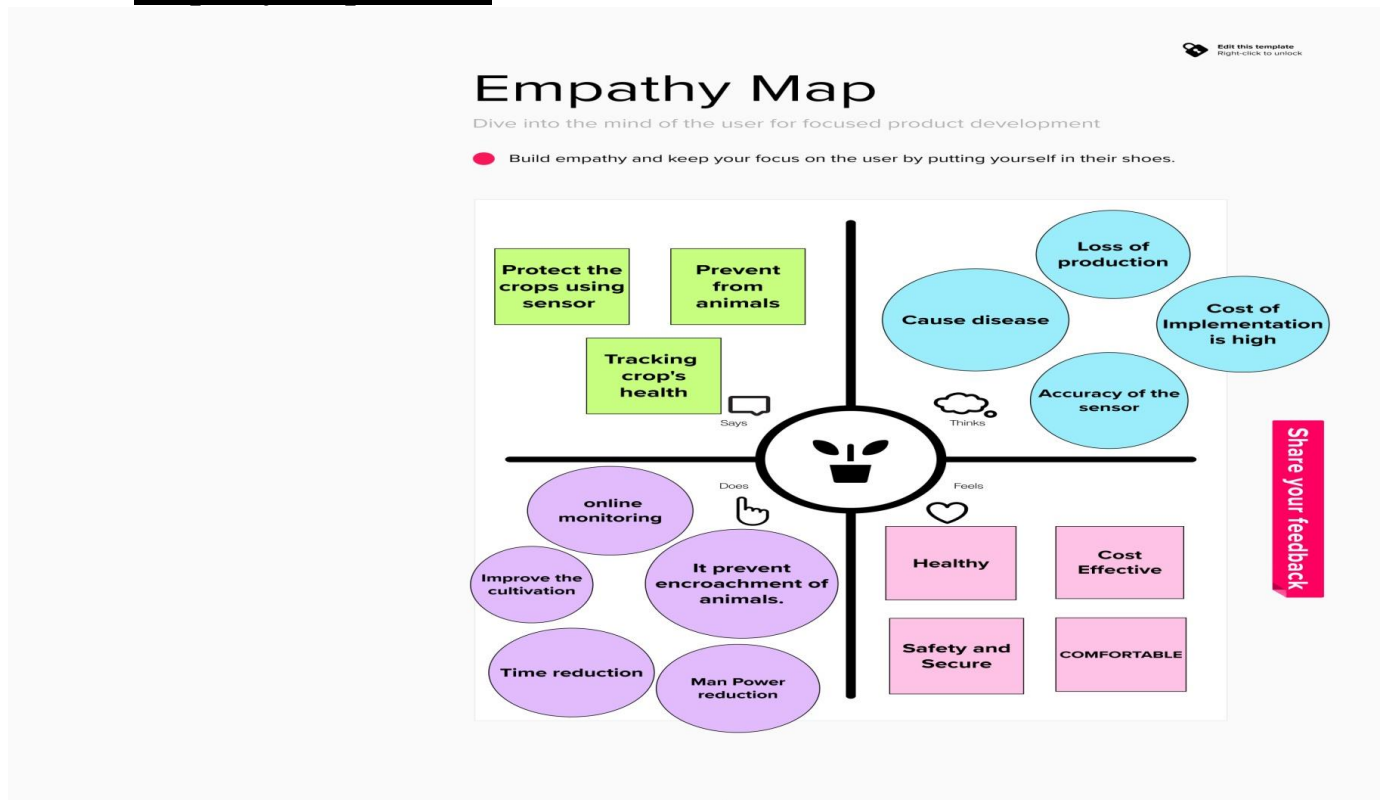
- ⇒ [https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20\(1\).pdf](https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service%20s%20(1).pdf)
- ⇒ [https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20\(1\).pdf](https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20(1).pdf)
- ⇒ <https://openweathermap.org/>
- ⇒ <https://smartinternz.com/assets/docs/Sending%20Http%20request%20to%20Open%20weather%20map%20web site%20to%20get%20the%20weather%20forecast.pdf>
- ⇒ <https://www.youtube.com/watch?v=cicTw4SEdxk>
- ⇒ [https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20\(1\).pdf](https://smartinternz.com/assets/docs/Smart%20Home%20Automation%20using%20IBM%20cloud%20Service s%20(1).pdf)
- ⇒ <https://github.com/rachuriharish23/ibmsubscribe>

## **2.3 Problem Statement Definition**

- ⇒ Agriculture is one of the Area which required urgent attention and advancement for high yield and efficient utilization of resources.
- ⇒ In this paper an approach of smart crop monitoring is presented through Internet of things (IOT).
- ⇒ A 4 Level framework is proposed namely sensing devices, sensor data level, base station level, edge computing and cloud data level for smart crop monitoring.
- ⇒ In this project, farm is going to get protected from humidity, temperature, and animals. With the help of IOT cloud module.
- ⇒ The agricultural form is been monitored with the help of MIT app and then, the data will be collected and stored it in cloud.
- ⇒ It will monitor and sense the humidity level and movement of animals and will send the message as notification to the user.

### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 Empathy Map Canvas



#### 3.2 Ideation & Brainstorming

##### What do they think and feel?

As its name may imply, smart farming is the use of technology in animal agriculture, and it's something that's been around since the Industrial Revolution. The biggest difference between then and now, though? "Motorized devices are being replaced with IOT".

##### What do they hear?

Smart farming is about using the new technologies which have arisen at the dawn of the Fourth Industrial Revolution in the areas of agriculture and cattle production to increase production quantity and quality, by making maximum use of resources and minimizing the environmental impact.

### **What do they see?**

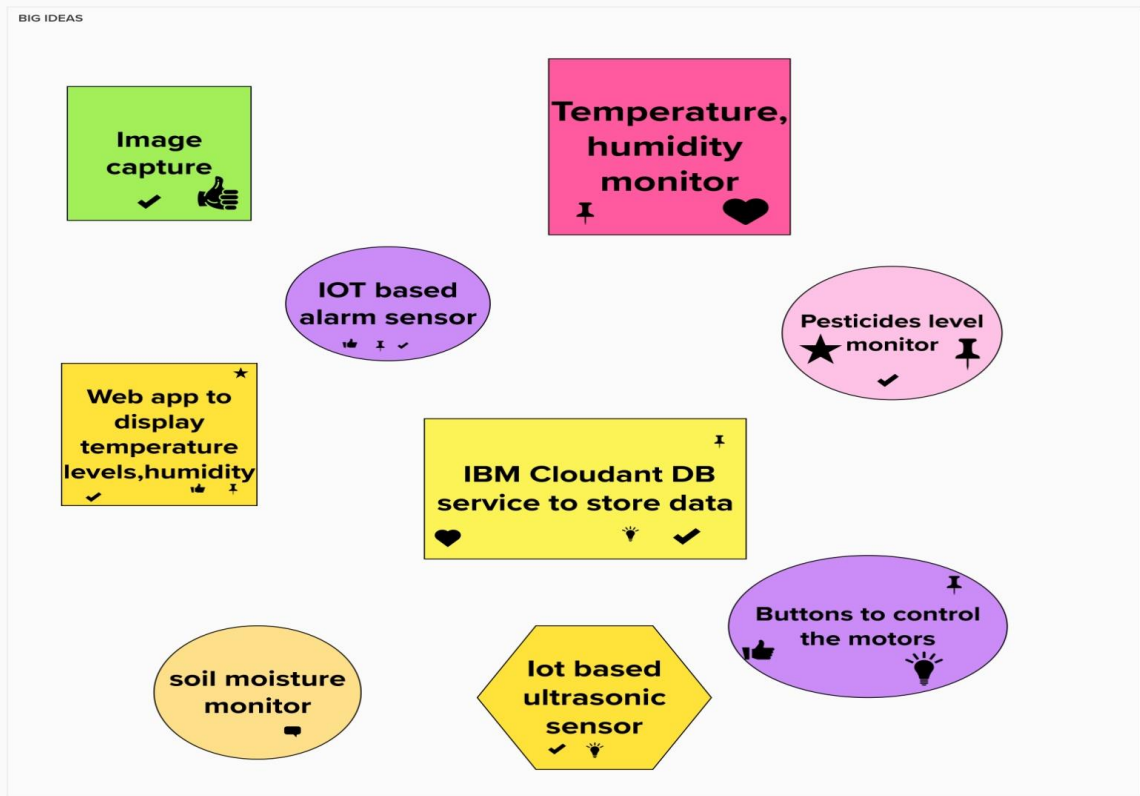
Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology – including big data, the cloud and the internet of things (IoT) – for tracking, monitoring, automating and analyzing operations.

### **What do they say and do?**

- ⇒ The aim of this technology is to **make the most of all the data** collected by various tools, by converting them into real **sources of information** in order to then define ways of simplifying agricultural work. It also allows for **accurate and predictive analysis** of all situations that may affect the farms, such as weather conditions (temperature, humidity, etc.) and sanitary or economic situations, for example. This makes it easier to organize the supply of energy, water, livestock feed and fertilizer.
- ⇒ In its most advanced form, smart farming facilitates the exchange of information between different farms, creating a real network of connected farms accessible from a smartphone or a computer.

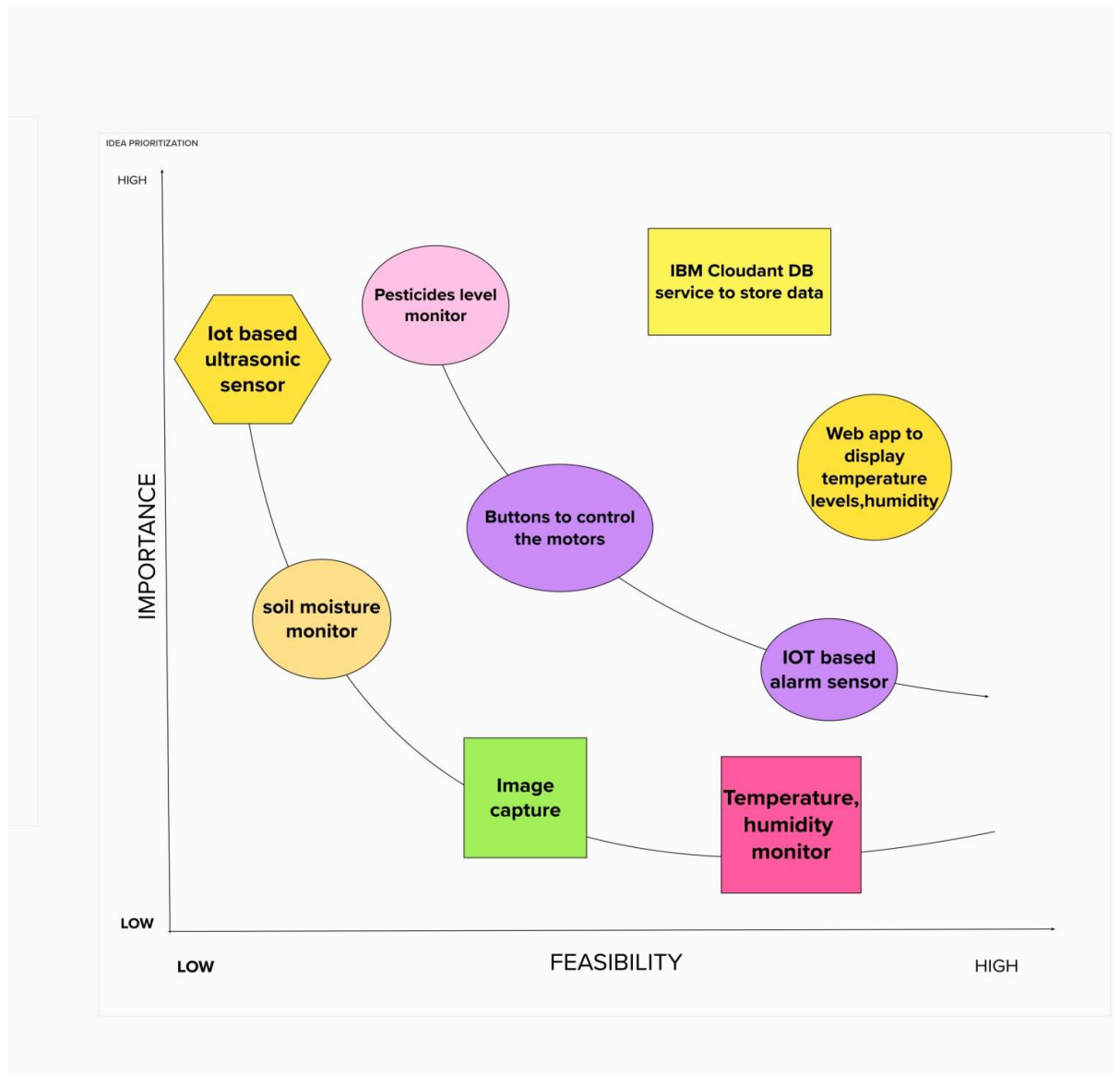
### **BRAINSTORM:**

## BRAINSTORM



## PRIORITIZATON:

PRIORITIZE



### 3.3 Proposed Solution

**Project Design Phase-I  
Proposed Solution Template**

Date	19 September 2022
Team ID	PNT2022TMID10974
Project Name	Project – IoT Based Smart Crop Protection System
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)	Crops in farms are many times ravaged by local animals like Buffaloes, Cows, Goats, Birds, and fire, etc... This leads to huge losses for the farmers. So, here we propose automatic crop protection system from animals and fire.
2.	Idea / Solution description	This System which assists the farmer in the irrigation process. We use IR Sensors to detect the Wild animals, Soil moisture and connected to Arduino microcontrollers.
3.	Novelty / Uniqueness	The smart farm protection system gives reliable security and safety to crops by monitoring 24/7.
4.	Social Impact / Customer Satisfaction	It increases the crop productivity and soil fertility. Peoples are live a healthy life.
5.	Business Model (Revenue Model)	It gives more protection and increase productivity of the crops at low cost.
6.	Scalability of the Solution	It increases the Protection of the crops in the field by indicating the Farmer through the Sensors.

### 3.3 Problem Solution fit

Problem-Solution fit canvas 2.0		IOT Based Smart Crop Protection System	
Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer? The Farmers are our customers, who are looking for the increase in crop protection.	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> What constraints prevent your customers from taking action or limit their choices of solutions? Our smart crop protection system is on budget and it would work only with internet connection and it is available on all smart devices.	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? When the mobile is on silent, it will trigger the alarm. By that alarm sound, the farmer will alert.
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>JB/P</span> Which jobs-to-be-done (or problems) do you address for your customers? This Smart crop protection requires quite a number of jobs like, when the problem is occur it needs to notify the owner and it also need to proper maintaining of crops accurately.	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> What is the real reason that this problem exists? What is the back story behind the need to do this job? If the animals are attacked the crops, it will cause huge loss to the owner. This system helps identify the problem with sensor. It will also set the alarm with it, so the farmer will always be alert. So the crop field is continuously monitored by sensor and provide proper solution.	<b>7. BEHAVIOUR</b> <span>BE</span> What does your customer do to address the problem and get the job done? The customer could get help from the help option in the settings of the application and if they are facing any issues they can make a report in that option and the authorities would look into the problem.
Focus on JB/P, tap into C	<b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act? i.e. seeing their neighbour installing For Example: if both customer use this device, any one of their farms are in danger it will notify to the respective owners. It does not create any confusion in sense. So the farmers maintain their farms safely.	<b>10. YOUR SOLUTION</b> <span>SL</span> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour. Our solution to crop protection is to create a crop tracker to observe the crops exact situation and with that the sensor will notify to the farmer. It will more secure to the crops to increase production.	<b>8. CHANNELS of BEHAVIOUR</b> This system will monitor 24/7 without the help of any workers. It will sense the accurate problem at exact time fr
	<b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> How do customers feel when they face a problem or a job and afterwards? At first farmer facing lot of consequences due to animals damaging their crops. After using this device they can maintain their crops effectively and it is possible to monitor soil quality, humidity, temperature and others.		

Commercial-NoDerivatives 4.0 license

AMALTAMA

## 4 REQUIREMENT ANALYSIS

### 3.4 Functional requirement



#### Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Requirements	Crop Protection Automatic Sprinkler System Monitors Soil Moisture ,Humidity and Temperature
FR-2	User Registration	Manual Registration Registration through webpage Registration through Form Registration through Gmail
FR-3	User Confirmation	Confirmation via Phone Confirmation via Email Confirmation via OTP
FR-4	Payment Options	Cash on Delivery Net Banking/UPI Credit/Debit/ATM Card
FR-5	Product Delivery and Installation	Door Step delivery Take away Free Installation and 1 year Warranty
FR-6	Product Feedback	Through Webpage Through Phone calls Through Google forms

### 3.5 Non-Functional requirements

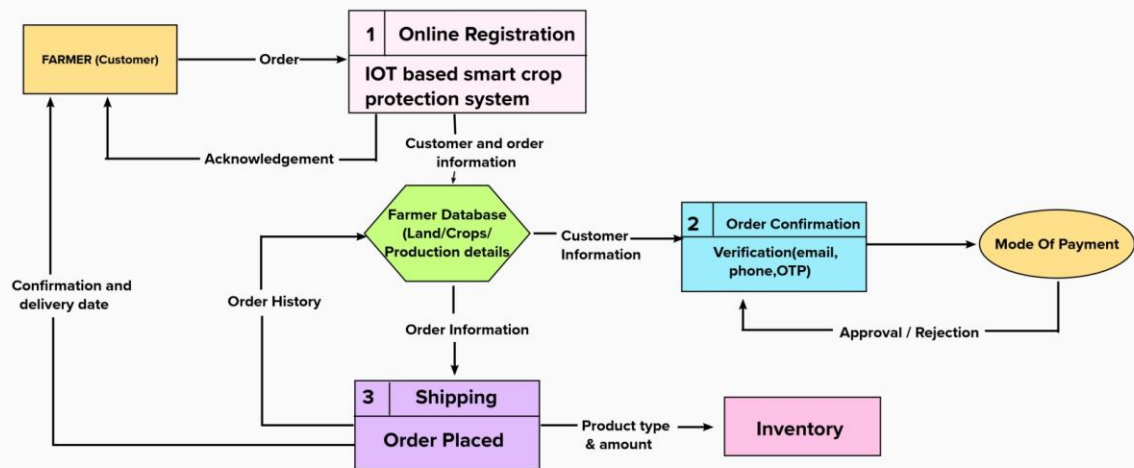
#### Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Have a clear and self-explanatory manual. Easier to use Even an illiterate farmer have to use the product without any difficulties
NFR-2	Security	Application has to be secured with 2 step authorisation Passwords and passkeys will be assigned as per the users need.
NFR-3	Reliability	Hardware requires a regular checking and service Software may be updated periodically Immediate alert is provided in case of any system failure
NFR-4	Performance	The application must have a good user interface It should have a minimal energy requirement It has to save water and energy
NFR-5	Availability	All the features will be available when the user requires. It depends on the need of the farmer and the customization the user has done.
NFR-6	Scalability	The product has to cover all the space of land irrespective of the size or area of a farm field.

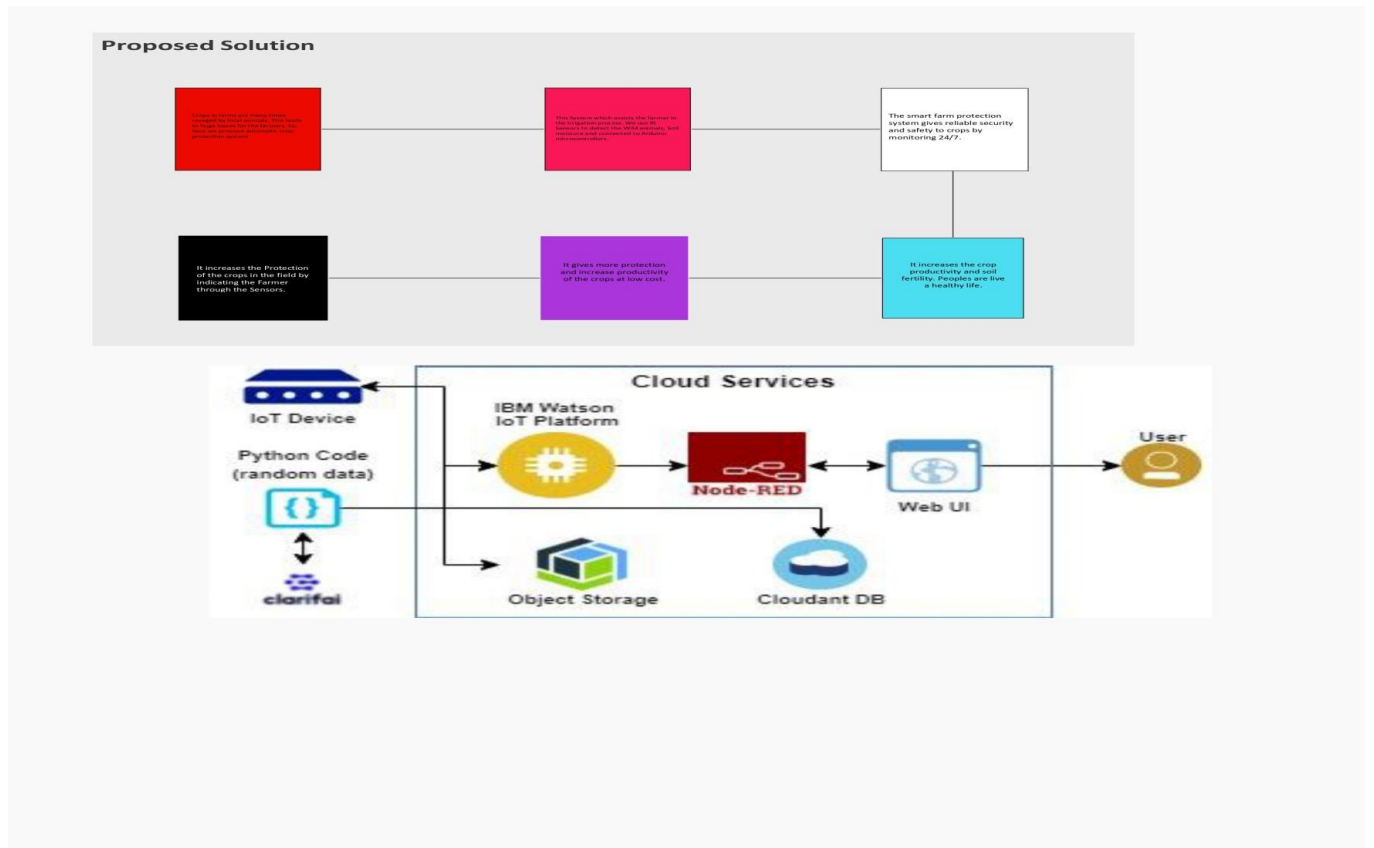
## 4. PROJECT DESIGN

### 4.1 Data Flow Diagrams



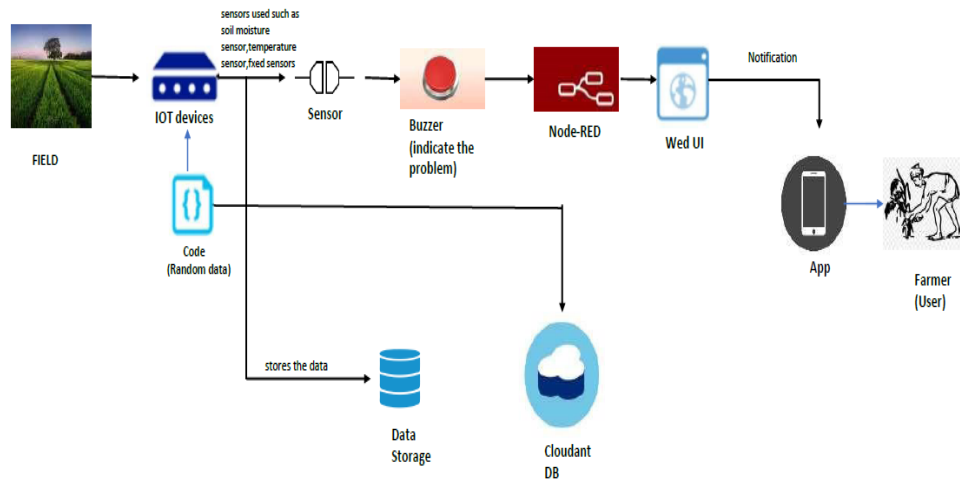
## 4.2 Solution & Technical Architecture

### SOLUTION ARCHITECTURE:



### TECHNICAL ARCHITECTURE:

**Solution Architecture Diagram:**



### 4.3 User Stories

**USER STORIES:**

User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance criteria	priority
Customer (Mobile user)	Download the database	USN-1	As a user I can register for the application by entering my email, password and confirming my password.	I can access my account/ dashboard	High
	Register	USN-2	As a user I can register for the application by entering my email, password and confirming my password.	I can receive confirmation email and click confirm	High
	Login	USN-3	As a user I will receive confirmation email once I have registered for the application.	I can register and access the dashboard with Facebook login	Low
	Upload the image	USN-4	As a user I must upload the image to identify the problem and works on it.		Medium
Customer (Web user)	The functional requirements are same as	Same as mobile user	Same as mobile user.	Same as mobile user	High when compare

## 5. PROJECT PLANNING & SCHEDULING

### Project Planning Phase Project Planning Template

Date	5 November 2022
Team ID	PNT2022TMID10974
Project Name	Project –IoT Based Smart Crop Protection For Agriculture
Maximum Marks	8 Marks

**Product Backlog, Sprint Schedule, and Estimation (4 Marks)**

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	IBM Cloud Services	USN-1	Create a Cloud Account in IBM	10	High	Bhrinta P Divya A Janavigha P Jayasurya S
Sprint-1	Software	USN-2	Install the Python IDE	5	Medium	Bhrinta P Divya A Janavigha P Jayasurya S
Sprint-1	Clarifai	USN-3	Create an Account in Clarifai (To detect the animals and birds we are using an open-source platform Clarifai.)	5	High	Bhrinta P Divya A Janavigha P Jayasurya S
Sprint-2	IBM Watson Platform	USN-4	Create IBM Watson IoT Platform and Device (It acts as the mediator to connect the web application to IoT device)	5	High	Bhrinta P Divya A Janavigha P Jayasurya S
Sprint-2	Node Red Services	USN-5	Create Node Red Services (To Create a Web Application)	5	High	Bhrinta P Divya A Janavigha P Jayasurya S

Sprint-2	<u>Cloudant</u> DB	USN-6	Create a Database in <u>Cloudant</u> DB (To Store the Image URL, Launch the <u>Cloudant</u> DB)	5	High	<u>Bhrintha</u> P <u>Divya</u> A <u>Janavigha</u> P Jayasurya S
----------	--------------------	-------	---	---	------	--

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Cloud Object Storage	USN-7	Create a Cloud Object Storage Service	5	High	<u>Bhrintha</u> P <u>Divya</u> A <u>Janavigha</u> P Jayasurya S
Sprint-4	Python Code	USN-8	Develop a Python Script	10	High	<u>Bhrintha</u> P <u>Divya</u> A <u>Janavigha</u> P Jayasurya S

**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	7 Days	24 Oct 2022	31 Oct 2022	15	31 Oct 2022
Sprint-2	15	7 Days	01 Nov 2022	07 Nov 2022	15	07 Nov 2022
Sprint-3	10	6 Days	08 Nov 2022	13 Nov 2022	10	13 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	20 Nov 2022	10	20 Nov 2022

6.

## 6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)  
Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	IBM Cloud Services	USN-1	Create a Cloud Account in IBM	10	High	Bhishha P Divya A Janavigha P Jayasriya S
Sprint-1	Software	USN-2	Install the Python IDE	5	Medium	Bhishha P Divya A Janavigha P Jayasriya S
Sprint-1	Clarifai	USN-3	Create an Account in Clarifai (To detect the animals and birds we are using an open-source platform Clarifai.)	5	High	Bhishha P Divya A Janavigha P Jayasriya S
Sprint-2	IBM Watson Platform	USN-4	Create IBM Watson IoT Platform and Device (It acts as the mediator to connect the web application to IoT device)	5	High	Bhishha P Divya A Janavigha P Jayasriya S
Sprint-2	Node Red Services	USN-5	Create Node Red Services (To Create a Web Application)	5	High	Bhishha P Divya A Janavigha P Jayasriya S
Sprint-2	Cloudant DB	USN-6	Create a Database in Cloudant DB (To Store the Image URL, Launch the Cloudant DB)	5	High	Bhishha P Divya A Janavigha P Jayasriya S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Cloud Object Storage	USN-7	Create a Cloud Object Storage Service	5	High	Bhishha P Divya A Janavigha P Jayasriya S
Sprint-4	Python Code	USN-8	Develop a Python Script	10	High	Bhishha P Divya A Janavigha P Jayasriya S

## 6.2 Sprint Delivery Schedule

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	7 Days	24 Oct 2022	31 Oct 2022	15	31 Oct 2022
Sprint-2	15	7 Days	01 Nov 2022	07 Nov 2022	15	07 Nov 2022
Sprint-3	10	6 Days	08 Nov 2022	13 Nov 2022	10	13 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	20 Nov 2022	10	20 Nov 2022

Velocity:

Sprint-1 and Sprint-2

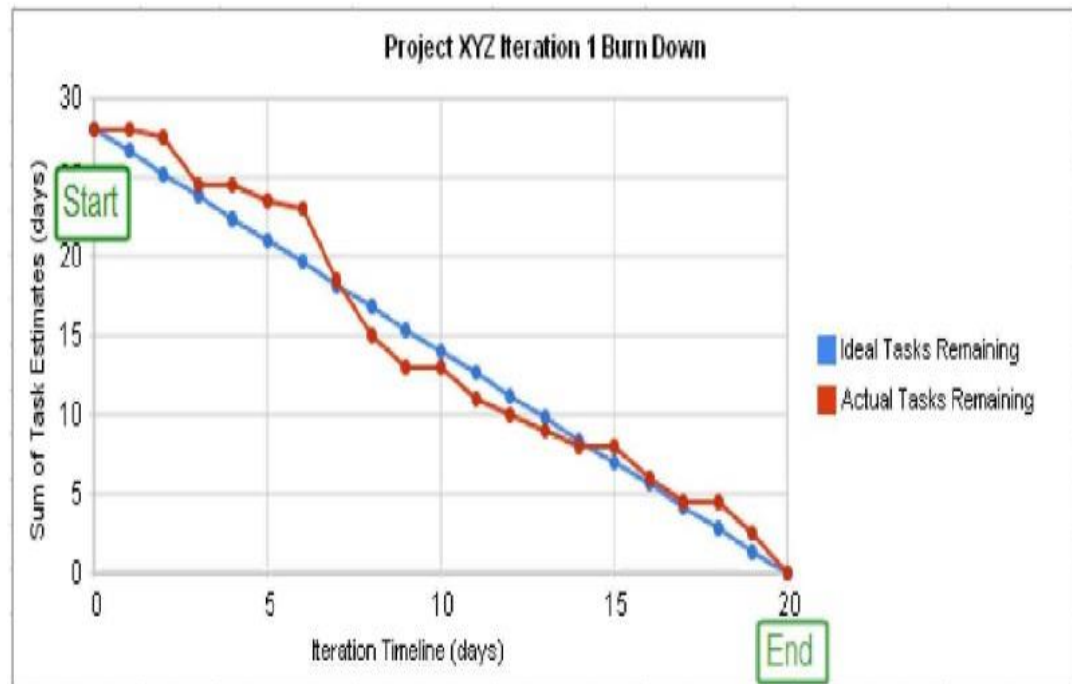
$$AV = \text{Sprint Duration} / \text{Velocity} = 15/7 = 2.14$$

Sprint-3 and Sprint-4

$$AV = \text{Sprint Duration} / \text{Velocity} = 10/6 = 1.6$$

## BURNDOWN CHART:

## Burndown Chart





## 7. CODING & SOLUTIONING

### 7.1 Feature 1

```
1 Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
2 Type "help", "copyright", "credits" or "license()" for more information.
3 import cv2
4 import numpy as np
5 import wiot.sdk.device
6 import playsound
7 import random
8 import time
9 import datetime
10 import ibm_boto3
11 from ibm_botocore.client import Config, ClientError
12
13 #CloudantDB
14 from cloudant.client import Cloudant
15 from cloudant.error import CloudantException
16 from cloudant.result import Result, ResultByKey
17 from clarifai_grpc.channel.clarifai_channel import ClarifaiChannel
18 from clarifai_grpc.grpc.api import service_pb2_grpc
19 stub = service_pb2_grpc.V2Stub(ClarifaiChannel.get_grpc_channel())
20 from clarifai_grpc.grpc.api import service_pb2, resource_pb2
21 from clarifai_grpc.grpc.api.status import status_code_pb2
22
23 #This is how you authenticate
24 metadata = (('authorization', 'key 0620e202302b4508b90eab7efe7475e4'),)
25 COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
26 COS_API_KEY_ID = "g5d4q08EIgV4TWUCJj4hFzgaIqEjrDbE82AJDWIAOHo"
27 COS_AUTH_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"
28 COS_RESOURCE_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/c2fa2836eaf3434bbc8b5b58fefff3f0:62e450fd-4c82-4153-ba41-ccb53adb8111::"
29 clientdb = cloudant(("apikey-W2njldnwtj016V53LAVUCqPwc2aHTLmlj1xXvtdGKJ8n", "88cc5f47c1a28afbfb8ad16161583f5a", url="https://d6c89f97-cf91-48b7-b14b-c99b2fe27c2f-bluemix.clouda
30 clientdb.connect()
31
32 #Create resource
33 cos = ibm_boto3.resource("s3",
34                         ibm_api_key_id=COS_API_KEY_ID,
35                         ibm_service_instance_id=COS_RESOURCE_CRN,
36                         ibm_auth_endpoint=COS_AUTH_ENDPOINT,
37                         config=Config(signature_version="oauth"),
```

```

37         config=Config(signature_version="oauth"),
38         endpoint_url=COS_ENDPOINT
39     )
40 def = multi_part_upload(bucket_name, item_name, file_path):
41     try:
42         print("Starting file transfer for {0} to bucket: {1}\n".format(item_name, bucket_name))
43         #set 5 MB chunks
44         part_size = 1024 * 1024 * 5
45         #set threshold to 15 MB
46         file_threshold = 1024 * 1024 * 15
47         #set the transfer threshold and chunk size
48         transfer_config = ibm_boto3.s3.transfer.TransferConfig(
49             multipart_threshold=file_threshold,
50             multipart_chunksize=part_size
51         )
52         #the upload_fileobj method will automatically execute a multi-part upload
53         #in 5 MB chunks size
54         with open(file_path, "rb") as file_data:
55             cos.Object(bucket_name, item_name).upload_fileobj(
56                 Fileobj=file_data,
57                 Config=transfer_config
58             )
59         print("Transfer for {0} Complete!\n".format(item_name))
60     except ClientError as be:
61         print("CLIENT ERROR: {0}\n".format(be))
62     except Exception as e:
63         print("Unable to complete multi-part upload: {0}".format(e))
64
65 def myCommandCallback(cmd):
66     print("Command received: %s" % cmd.data)
67     command=cmd.data['command']
68     print(command)
69     if(command=="lighton"):
70         print('lighton')
71     elif(command=="lightoff"):
72         print('lightoff')
73     elif(command=="motoron"):
74         print('motoron')

```

```

74     print('motoron')
75     elif(command=="motoroff"):
76         print('motoroff')
77     myConfig = {
78         "identity": {
79             "orgId": "chytun",
80             "typeId": "NodeMCU",
81             "deviceId": "12345"
82         },
83         "auth": {
84             "token": "12345678"
85         }
86     }
87     client = wiot.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
88     client.connect()
89
90     database_name = "sample"
91     my_database = clientdb.create_database(database_name)
92     if my_database.exists():
93         print(f'({database_name}) successfully created.')
94     cap=cv2.VideoCapture("garden.mp4")
95     if(cap.isOpened()==True):
96         print('File opened')
97     else:
98         print('File not found')
99
100    while(cap.isOpened()):
101        ret, frame = cap.read()
102        gray = cv3.cvtColor(frame, cv2.COLOR_BGR@GRAY)
103        imS= cv2.resize(frame, (960,540))
104        cv2.imwrite('ex.jpg',imS)
105        with open("ex.jpg", "rb") as f:
106            file_bytes = f.read()
107        #This is the model ID of a publicly available General model. You may use any other public or custom model ID.
108        request = service_pb2.PostModelOutputsRequest(
109            model_id='e9359dbe6ee44dbc8842ebe97247b201',
110            ...         inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))

```

```

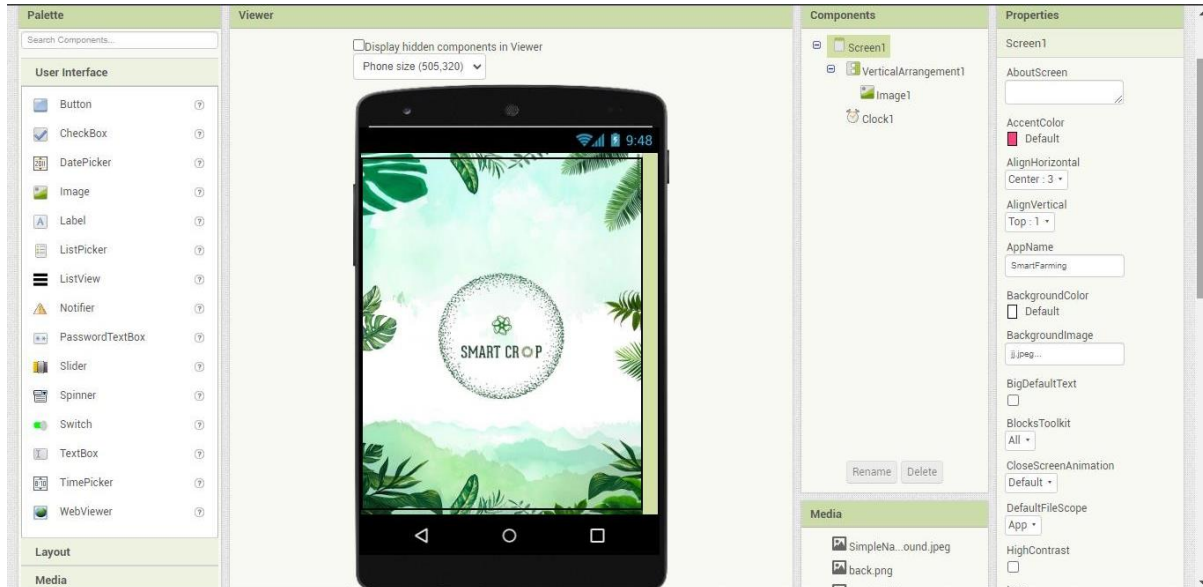
110    ...         inputs=[resources_pb2.Input(data=resources_pb2.Data(image=resources_pb2.Image(base64=file_bytes))
111    ...         ))]
112    response = stub.PostModelOutputs(request, metadata=metadata)
113    if response.status.code != status_code_pb2.SUCCESS:
114        raise Exception("Request failed, status code: " + str(response.status.code))
115    detect=False
116    for concept in response.outputs[0].data.concepts:
117        #print('%12s: %.f' % (concept.name, concept.value))
118        if(concept.value>0.98):
119            #print(concept.name)
120            if(concept.name=="animal"):
121                print("Alert! Alert! animal detected")
122                playsound.playsound('alert.mp3')
123                picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
124                cv2.imwrite(picname+'.jpg',frame)
125                multi_part_upload('Dhakshesh', picname+'.jpg', picname+'.jpg')
126                json_document={"link":COS_ENDPOINT+'/'+'Dhakshesh+'/'+'picname+'.jpg'}
127                new_document = my_database.create_document(json_document)
128                if new_document.exists():
129                    print(f"Document successfully created.")
130                time.sleep(5)
131                detect=True
132                moist=random.randint(0,100)
133                humidity=random.randint(0,100)
134                myData={'Animal':detect,'moisture':moist,'humidity':humidity}
135                print(myData)
136                if(humidity!=None):
137                    client.publishEvent(eventId="status",msgFormat="json", daya=myData, qos=0, onPublish=None)
138                    print("Publish Ok..")
139                client.commandCallback = myCommandCallback
140                cv2.imshow('frame',imS)
141                if cv2.waitKey(1) & 0xFF == ord('q'):
142                    break
143                client.disconnect()
144                cap.release()

```

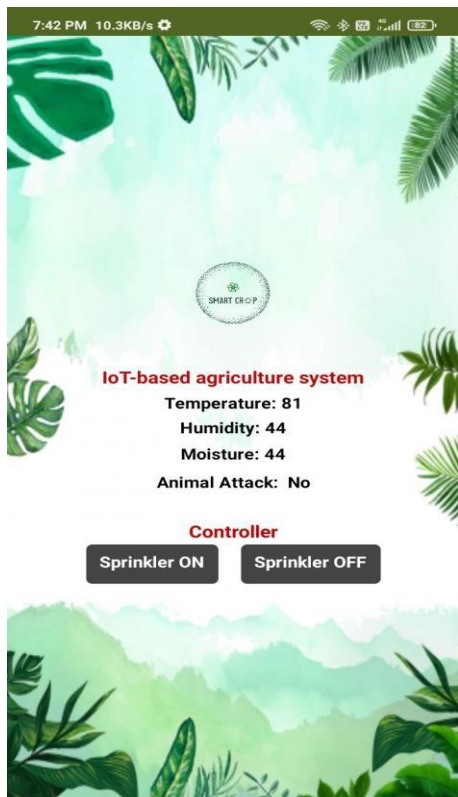
```
*IDLE Shell 3.8.8*
File Edit Shell Debug Options Window Help
Python 3.8.8 (tags/v3.8.8:024d805, Feb 19 2021, 13:18:16) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/HP/Desktop/crop/crop_protect.py =====
2021-04-06 12:52:19,640 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:hj5fmy:NodeMCU:12345
'sample' successfully created.
File opened
{'Animal': False, 'moisture': 17, 'humidity': 41}
Publish Ok..
{'Animal': False, 'moisture': 84, 'humidity': 16}
Publish Ok..
{'Animal': False, 'moisture': 48, 'humidity': 43}
Publish Ok..
{'Animal': False, 'moisture': 0, 'humidity': 3}
Publish Ok..
{'Animal': False, 'moisture': 73, 'humidity': 68}
Publish Ok..
{'Animal': False, 'moisture': 26, 'humidity': 26}
Publish Ok..
{'Animal': False, 'moisture': 96, 'humidity': 59}
Publish Ok..
I
Ln: 10 Col: 11
```

## 7.2 Feature 2

### MIT app inventor to design the app:



### Customizing the app interface to display the values:



## 8 TESTING

### 8.2 Test Cases

#### ▲ Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	11	4	2	2	19
Duplicate	1	1	2	0	4
External	2	3	0	1	6
Fixed	10	2	3	20	35
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	1	3
Won't Fix	0	5	2	1	8
Totals	24	15	13	25	77

#### Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	47	0	2	45

Security	3	0	0	3
Outsource Shipping	2	0	0	2
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	5
Version Control	3	0	1	2

## **9 RESULTS**

Thus the IOT based Smart Crop Protection has been build successfully with the help of MIT app, Node.Js, and node red. And the output has been tested and verified using MIT app.

### **ADVANTAGES :**

- ⇒ Sensors in Agriculture invented to meet the increasing demand for food with minimum resources such water, fertilizers and seeds.
- ⇒ They are easy to operate and use and easy to maintain.
- ⇒ Sensors are cheaper in price and best in quality.
- ⇒ They can used for measuring pollution and global warming for their fields and crops.

### **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.

## **10 CONCLUSION:**

Smart farming reduces the ecological footprint of farming. Minimized or site-specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases

## **11 FUTURE SCOPE**

IoT smart agriculture products are designed to help monitor crop fields using sensors and by automating irrigation systems. As a result, farmers and associated brands can easily monitor the field conditions from anywhere without any hassle.

## **12 APPENDIX**

→ **GitHub Link**

<https://github.com/IBM-EPBL/IBM-Project-6209-1658824607>

→ **Source Code**

<https://github.com/IBM-EPBL/IBM-Project-6209-1658824607/blob/main/Final%20Deliverables/Final%20Code/Source%20code>

**Youtube link**

<https://youtu.be/99y4MEHmvqM>

Prepared by:

- P.Bhrintha
- S.Jayasurya
- P.Janavigha
- A.Divya



**THANK YOU!**