IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE

Submitted by

KARTHIKEYAN A (923819104019)

ABEESH K (923819104001)

PANDIYARAJAN A (923819104033)

SURIYAPRAKASH P (923819104046)

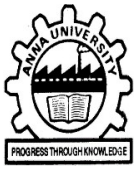
BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

MANGAYARKARASI COLLEGE OF ENGINEERING,

MADURAI 625 402



ANNA UNIVERSITY:: CHENNAI 600 025

TABLE OF CONTENTS

CHAPTER-1

Introduction

Agriculture is a pillar of India's economy and deserves security. Security and protection are required at the very initial stage, like protection from attacks of rodents or insects in the fields and as well as grain stores. Thosechallenges also need to be taken into account. The security systems that are used today are not smart enough to deliver real-time notification after detecting the problem. Climate change, soil erosion and loss of biodiversity also increase the pressure of farmers and a drastic decrease in the crop production.

Project Overview

An intelligent crop protection system helps the farmers in protecting the crop from the animals and birds which destroy the crop. This system also helps farmers to monitor the soil moisture levels in the field and also the temperature and humidity values near the field. The motors and sprinklers in the field can be controlled using the mobile application. The device will detect the animals and birds using the Clarified service. If any animal or bird is detected the image will be captured and stored in the IBM Cloud object storage. It also generates an alarm and avoid animals from destroying the crop The image URL will be stored in the IBM Cloudant DB service. The device will also monitor the soil moisture levels, temperature, and humidity values and send them to the IBM IOT Platform. The image will be retrieved from Object storage and displayed in the web application. A web application is developed to visualize the soil moisture, temperature, and humidity values.

Purpose

The Main motive of this project is to protect the various agriculture fields from factors such as birds, animals, insects and other climatic conditions. This system helps the crops to protect from ravaging animals also birds feeding into newly growing crops ,This system can also find the increase in water in level during extreme rain .With this system the surrounding temperature ,humidity, moisture content can also be detected.

CHAPTER-2

LITERATURE SURVEY

Existing problem

As new technologies has been introduced and utilized in modern world, there is a need to bring advancement in the sector of agriculture also. Various Researches have been undergone to enhance crop cultivation and are widely used. So as to enhance the crop productivity efficiently, it is necessary to monitor the environmental conditions in and around the field. The parameters that have to be exact monitored to enhance the yield are soil characteristics, weather conditions, moisture, temperature, etc., Internet of Things (IOT) is being utilized in a number of real time applications. The introduction of Internet of thing (IOT) along with the sensor network in farmer furbishes the traditional way of farming. Online crop monitoring the use of IOT helps the farmers to stay related to his subject from somewhere andanytime. Various sensors are used to screen and collect records about the area conditions. Collectively the about the farm circumstance is disbursed to the farmer thru GSM technology.

References

1. Official webpage of the European Smart crop protection System at:http://effis.jrc.ec.europa.eu/

2. Official webpage of the Copernicus Earth Observation Programme at:http://www.copernicus.eu

3. Forest Fires in Europe, Middle East and North Africa 2016, JRC Science for policy report,BN 978-92-79-71292-0, ISSN 1831-9424, doi:10.2760/17690, availabe at: http://effis.jrc.ec.europa.eu/media/cms_page_media/40/Smart_Crop_in_Europe_Middle_east_and_North_Africa_2016_final_pdf_JZU7He L.pd

Problem Statement Definition

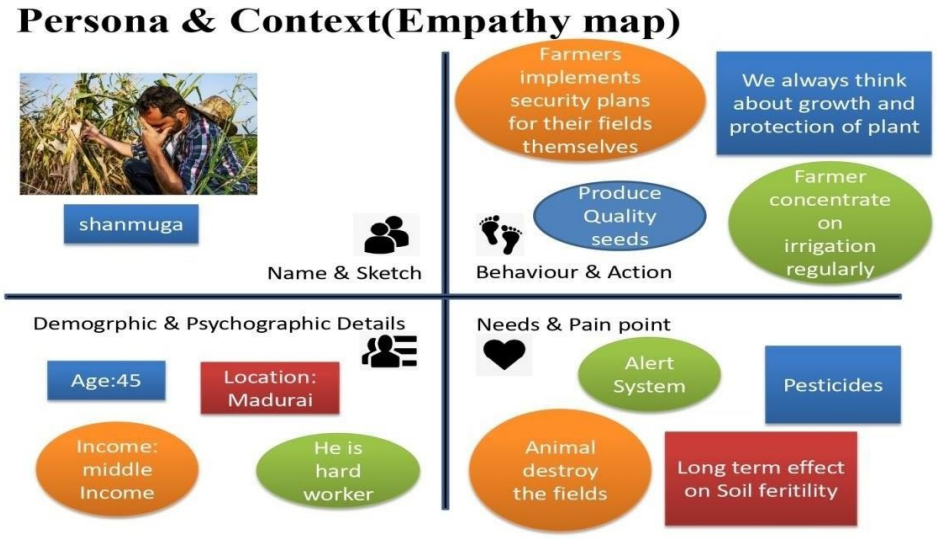
The problem of wild life attack on crops i.e., Animals,Birds cause a lot of damage to crops by running over them, eating and completely vandalizing them. This lead to poor yield of crops and significant financial loss to the owners of the farmland.

CHAPTER-3

IDEATION AND PROPOSED SOLUTION

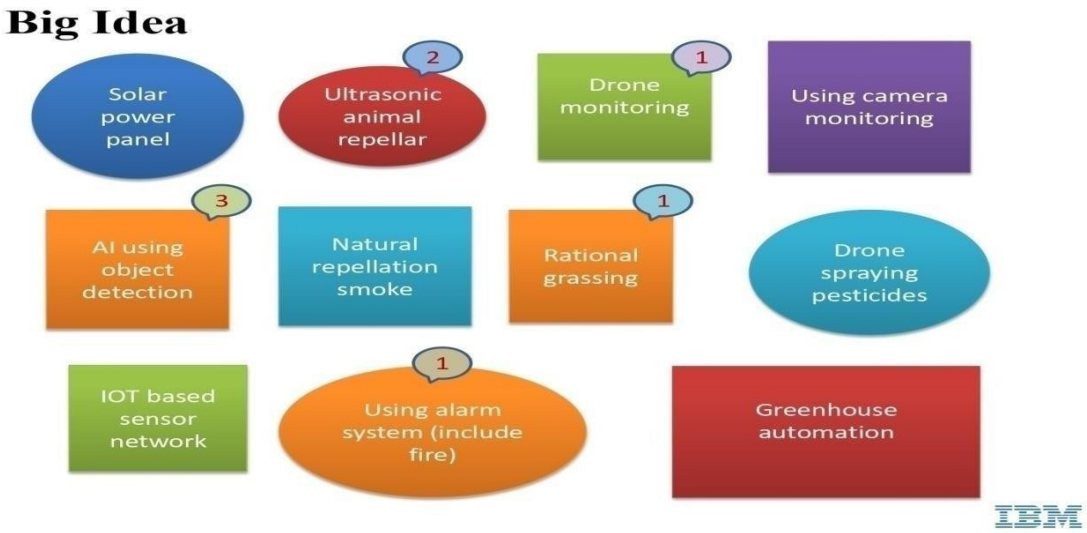
Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.



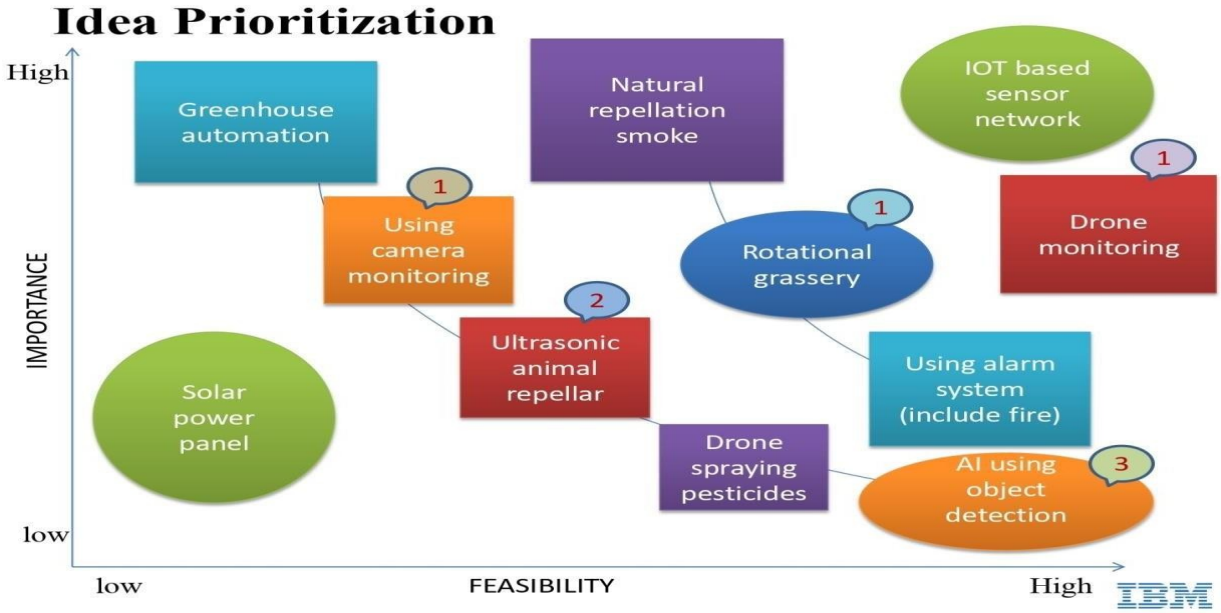
Big Ideas

It consists of all the ideas of instruments and equipments that we are going to implement in this project.

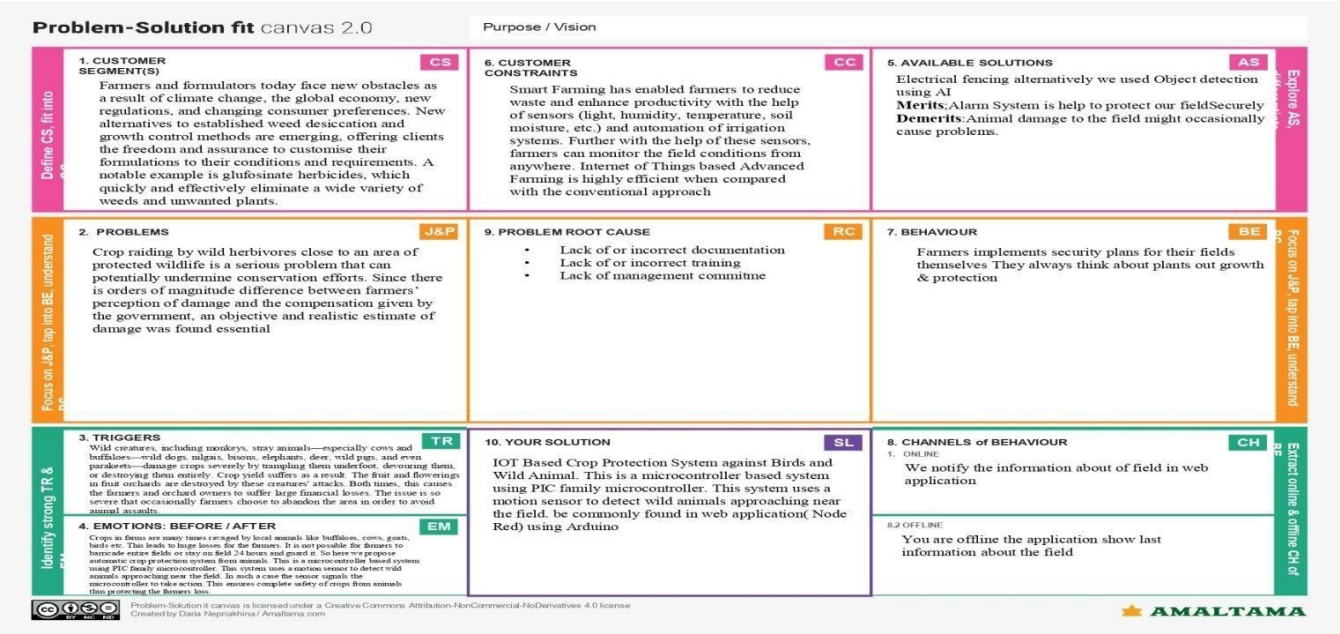


Idea Prioritization

It deals with the prioritizing of the big ideas in order of highest to lowest likes.



Problem Solution Fit



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**	Animals are some concepts. This system is microcontroller- IOT-Based Crop Protection System Against Birds and Wild based and uses a microcontroller from the PIC family. To identify wild animals entering the field, this device employs a motion sensor. using Arduino, be frequently found in mobile devices.
3.	**Novelty / Uniqueness**	Object detection using **Artificial intelligence**

4.	**Social Impact / Customer Satisfaction**	Currently, there is a significant social issue with wild animals and fires damaging crops. Since there hasn't yet been a good answer to this issue, it needs urgent attention. In light of its intent to address this issue, this project has significant social significance. With the aid of this project, farmers will be able to safeguard their orchards and fields, preventing them from suffering major monetary losses and needless labour costs in the process. Additionally, this will aid in increasing crop yields, which will benefit their ability to support themselves financially.
5.	**Business Model (Revenue Model)**	
6.	**Scalability of the Solution**	**IOT Sensor, Camera led monitoring & assessment of the soil and crop** This permits prompt preventive action to ensure crop spoiling is reduced and production is improved without intensive input of synthetic ingredients or fertilisers, increasing the farmers' income

CHAPTER-4

REQUIREMENT ANALYSIS

Functional Requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User registration	Registration through Gmail Create an account Follow the instructions
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Interface sensor	Interface sensor and the application so if animals enter the field it gives alarm.
FR-4	Accessing datasets	Datasets are retrieved from Cloudant DB
FR-5	Web application	Motors and sprinklers in the field can be controlled by web application.

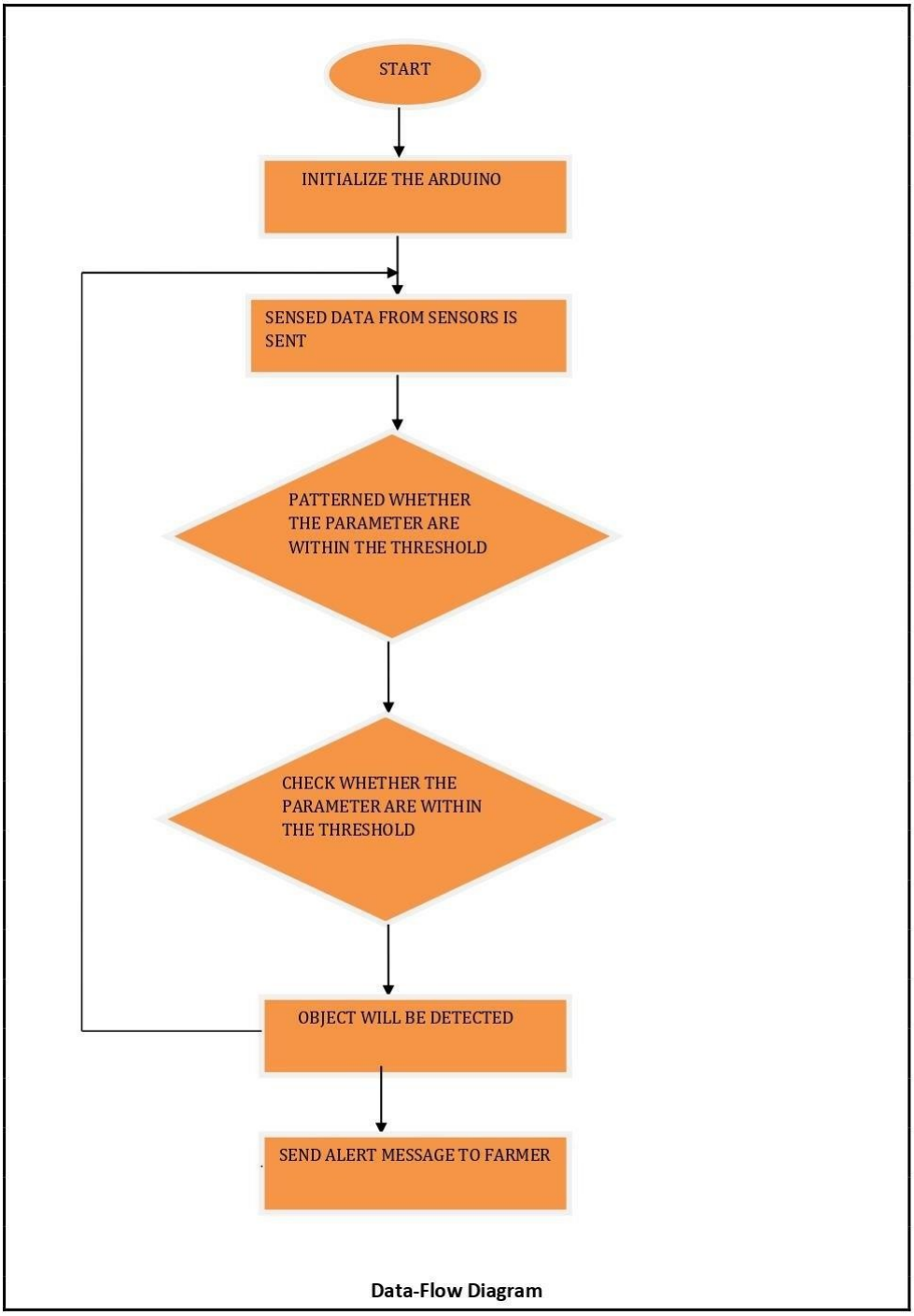
Non-Functional Requirements

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

CHAPTER-5

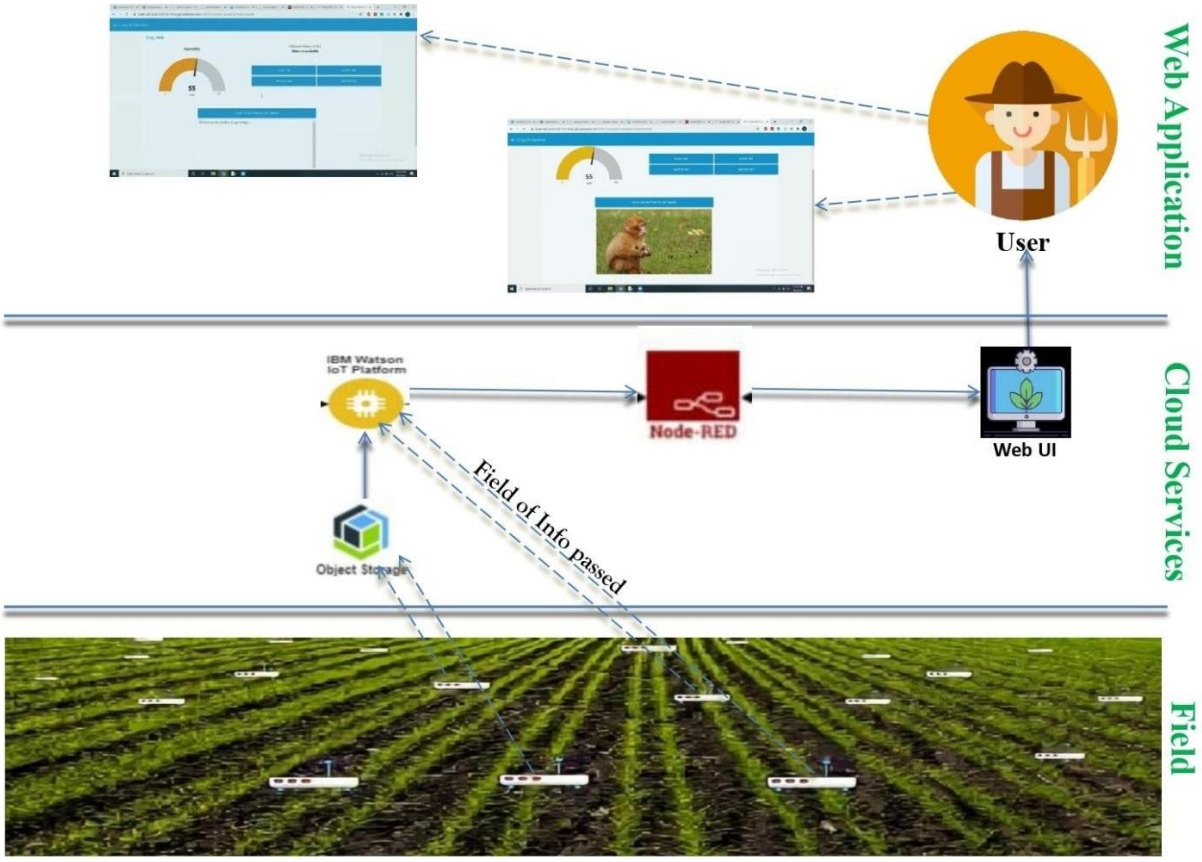
PROJECT DESIGN

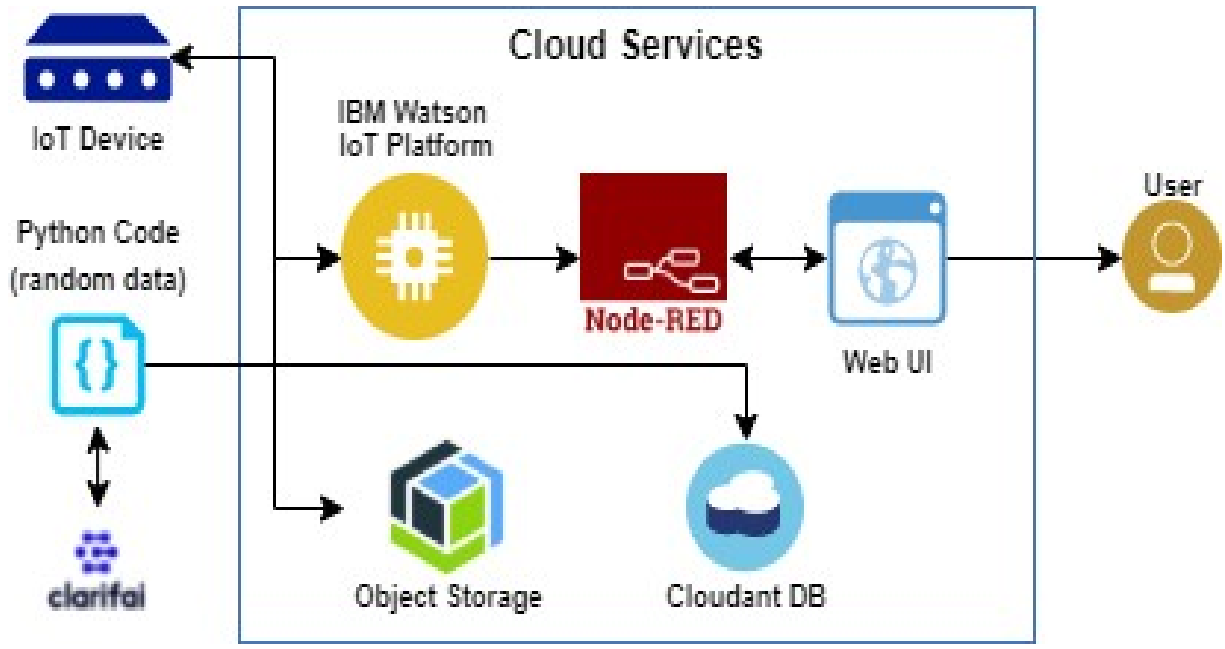
Data Flow Diagram



Data-Flow Diagram

Solution & Technical Architecture





Customer Journey Map

CHAPTER-6

PROJECT PLANNING PHASE

Sprint Planning, Schedule & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a farmer, I can register for the application by entering my email, password, and Confirming my password.	2	High	K Abeesh
Sprint-1	User Confirmation	USN-2	As a farmer, I will receive confirmation email once I have registered for the application	1	Medium	P Suriyaprakash
Sprint-1	Login	USN-3	As a farmer, I can log into the application by entering email & password	2	High	A Karthikeyan
Sprint-2	Interface Sensor	USN-1	A sensor interface is a bridge between a device and any attached sensor. The interface takes data collected by the sensor and outputs it to the attached device.	2	High	A Karthikeyan A Pandiyarajan
Sprint-3	Coding (Accessing datasets)	USN-1	Coding is a set of instructions used to manipulate information so that a certain input results in a particular output.	2	High	K Abeesh A Karthikeyan A Pandiyarajan P Suriyaprakash
Sprint-4	Web Application	USN-1	As a Farmer, I will show the current Information of the Field.	1	Medium	K Abeesh P Suriyaprakash

Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	4 Days	24 Oct 2022	27 Oct 2022	20	29 Oct 2022
Sprint-2	20	5 Days	28 Oct 2022	01 Nov 2022	20	04 Nov 2022
Sprint-3	20	8 Days	02 Nov 2022	09 Nov 2022	20	11 Nov 2022
Sprint-4	20	9 Days	10 Nov 2022	18 Nov 2022	20	19 Nov 2022

Reports From JIRA

CHAPTER-7

CODING AND SOLUTION

7.1 Feature

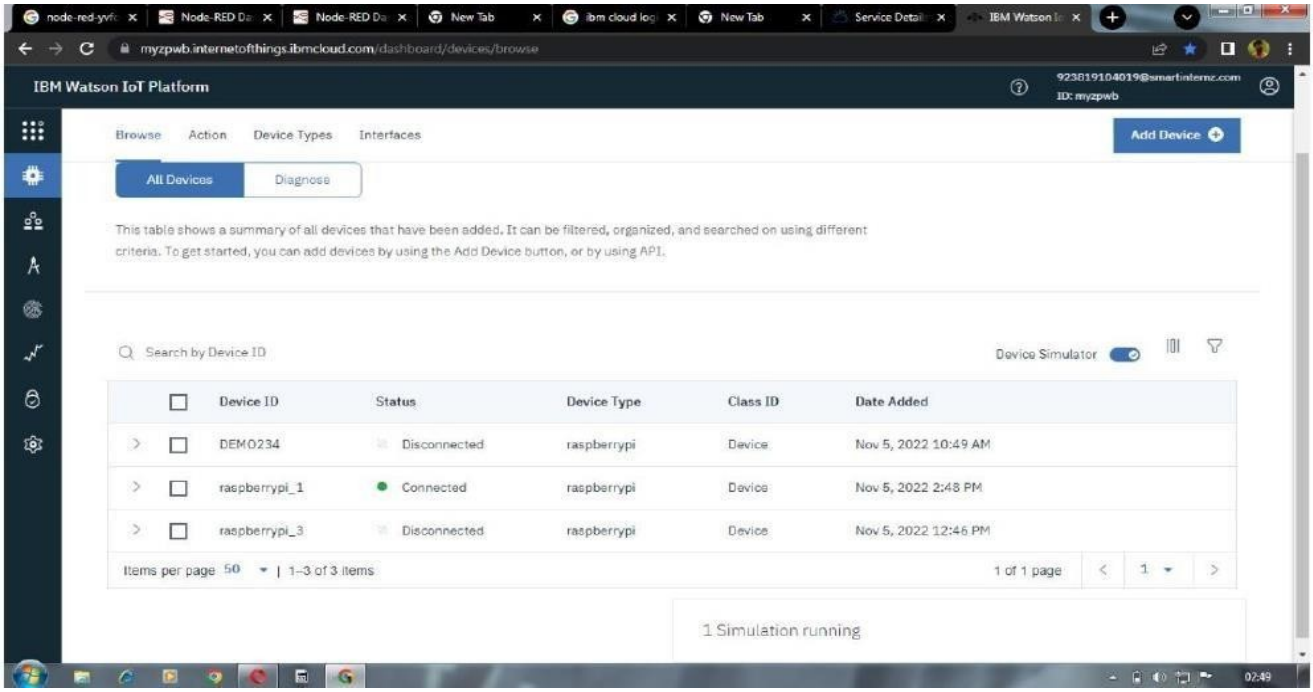
```
import cv2
import numpy as np
importwiotp.sdk.device
import playsound
import random
import time
import
datetime
import
ibm_boto3
from ibm_botocore.client import Config,
ClientErrorfrom cloudant.client import
Cloudant
from cloudant.error import CloudantException
from cloudant.result import Result, 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
# Constants for IBM COS values
COS_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list
avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints
COS_API_KEY_ID =
"9Vp5U7vuntgNduJwVKZXDn7f4wKRPaDJT5a1kiDXapEP" # eg
"W00YiRnLW4a3fTjMB-odB-2ySfTrFBIQQWanc--P3byk"
COS_AUTH_ENDPOINT =
"https://iam.cloud.ibm.com/identity/token"  COS_RESOURCE_CRN
= "crn:v1:bluemix:public:cloud-object-
storage:global:a/9b399604cf904a88a997a81684b97184:24b5d2b1-5259-4c70-83ac-
24552802d92e::"  # eg "crn:v1:bluemix:public:cloud-object-
storage:global:a/3bf0d9003abfb5d29761c3e97696b71c:d6f04d83-6c4f- 4a62-a165-
696756d63903::"

clientdb=Cloudant("apikey-v2-thovtw7l8mtpl3w17or63t37f6a4wivqvvgg4gkhi83",
```

```
"89c52b4f4fdd6459155676646ee45a3a", url="https://apikey-v2-
thovtw7l8mtpl3w17or63t37f6a4wivqvvgg4gkhi83:89c52b4f4fdd6459155676646ee45a3a@a6f904
5f-ff4b- 4a32-a9a0-35efb443e91b-bluemix.cloudantnosqldb.appdomain.cloud")
clientdb.connect()

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))part_size = 1024 * 1024 * 5
    file_threshold = 1024 * 1024 * 15
    transfer_config = ibm_boto3.s3.transfer.TransferConfig(
        multipart_threshold=file_threshold,
        multipart_chunksize=part_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:
{1}\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: %$" %
  cmd.data)command=cmd.data[
  'command' ] print(command)
  if(command=='light
     on'):
     print('lighton')
  elif(command=='lig
     htoff'):
     print('lightoff')
  elif(command=='m
     otoron'):
     print('motoron')
```

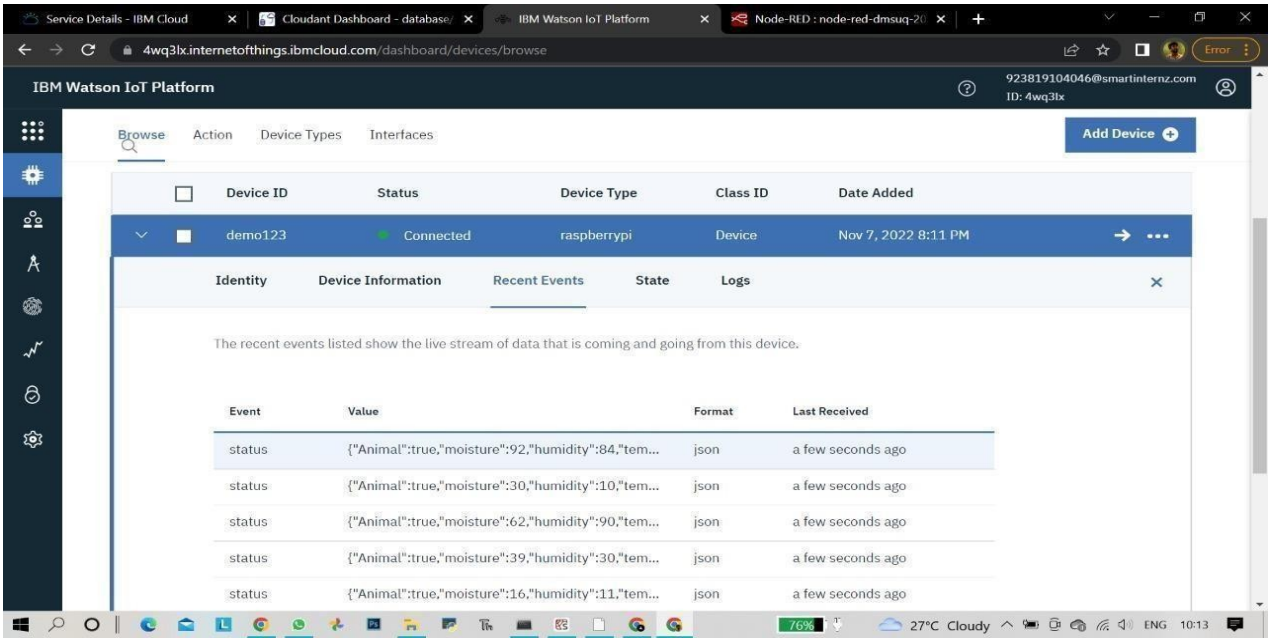
```
  elif(command=='m
     otoroff'):
     print('motoroff')
myConfig =  {
            "identity":
             { "orgId":
             "4wq3lx",
            "typeId": "raspberrypi",
            "deviceId":  "demo123"
             },
   "auth" : {
            "token": "mind1234"
                  }
            }
client= wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None) client.connect()

database_name ="sample"
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()
       print("Alert! Alert! Animal detected")
       picname=datetime.datetime.now().strftime("%y-%m-%d-
       %H-%M") cv2.imwrite(picname+'.jpg',frame)
       multi_part_upload("karthi01", picname+'.jpg', picname+'.jpg')
       json_document={"link":COS_ENDPOINT+'/'+'karthi01'+'/'+picname+'.jp
       g'} 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.destroyAllWindo
w()
```

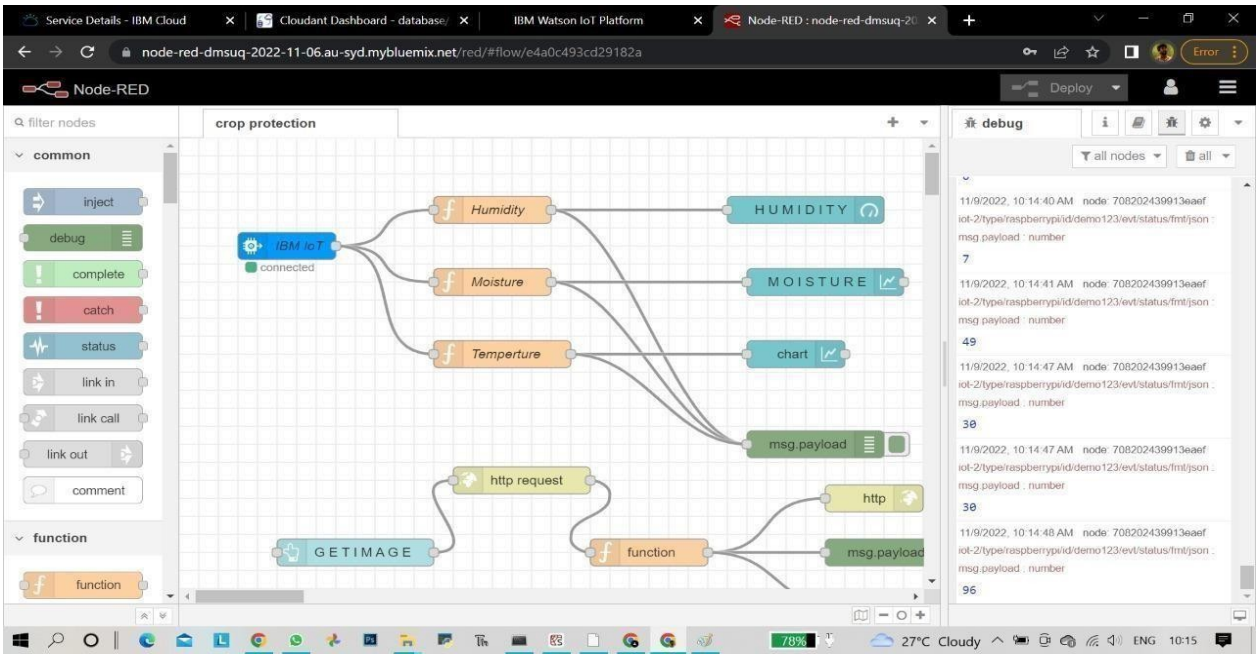
Device Details:

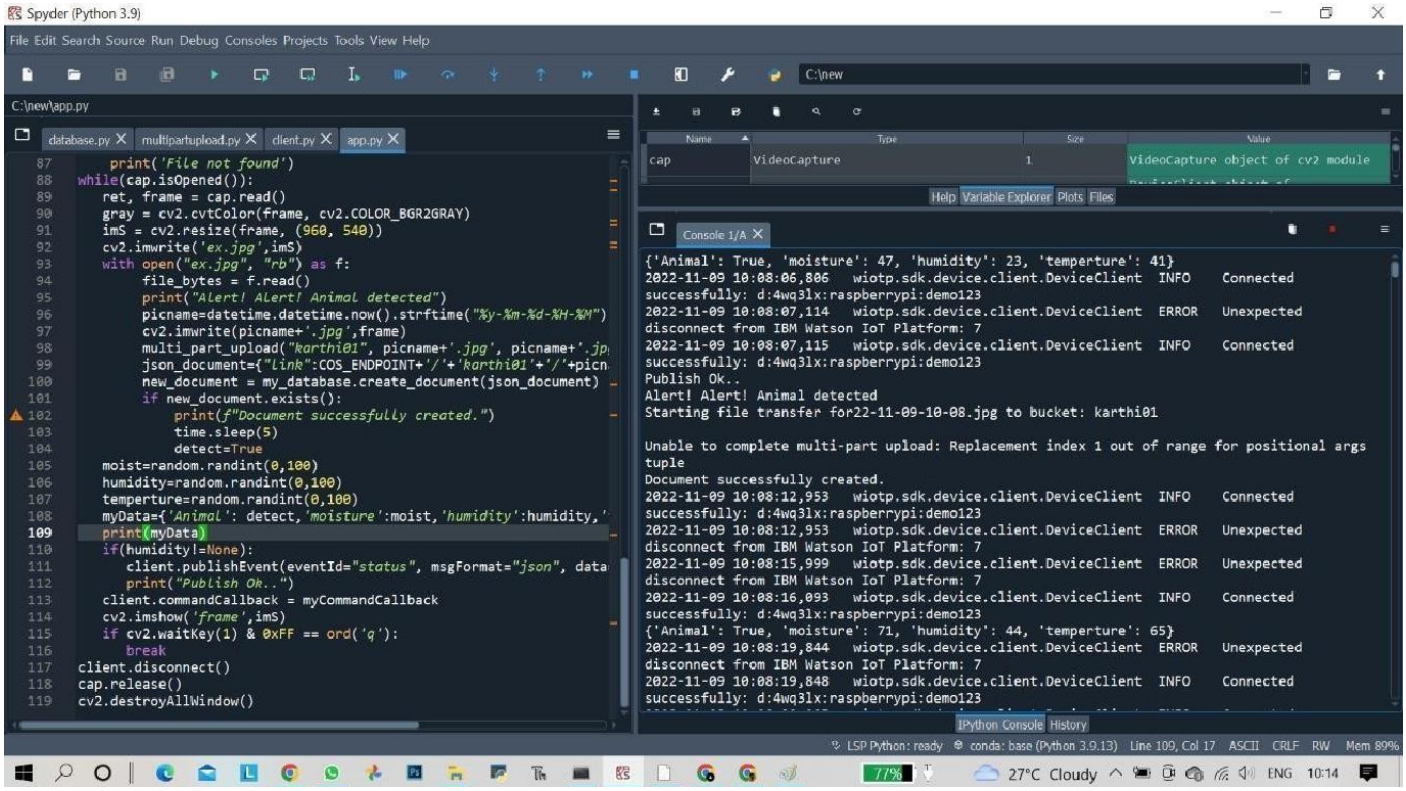


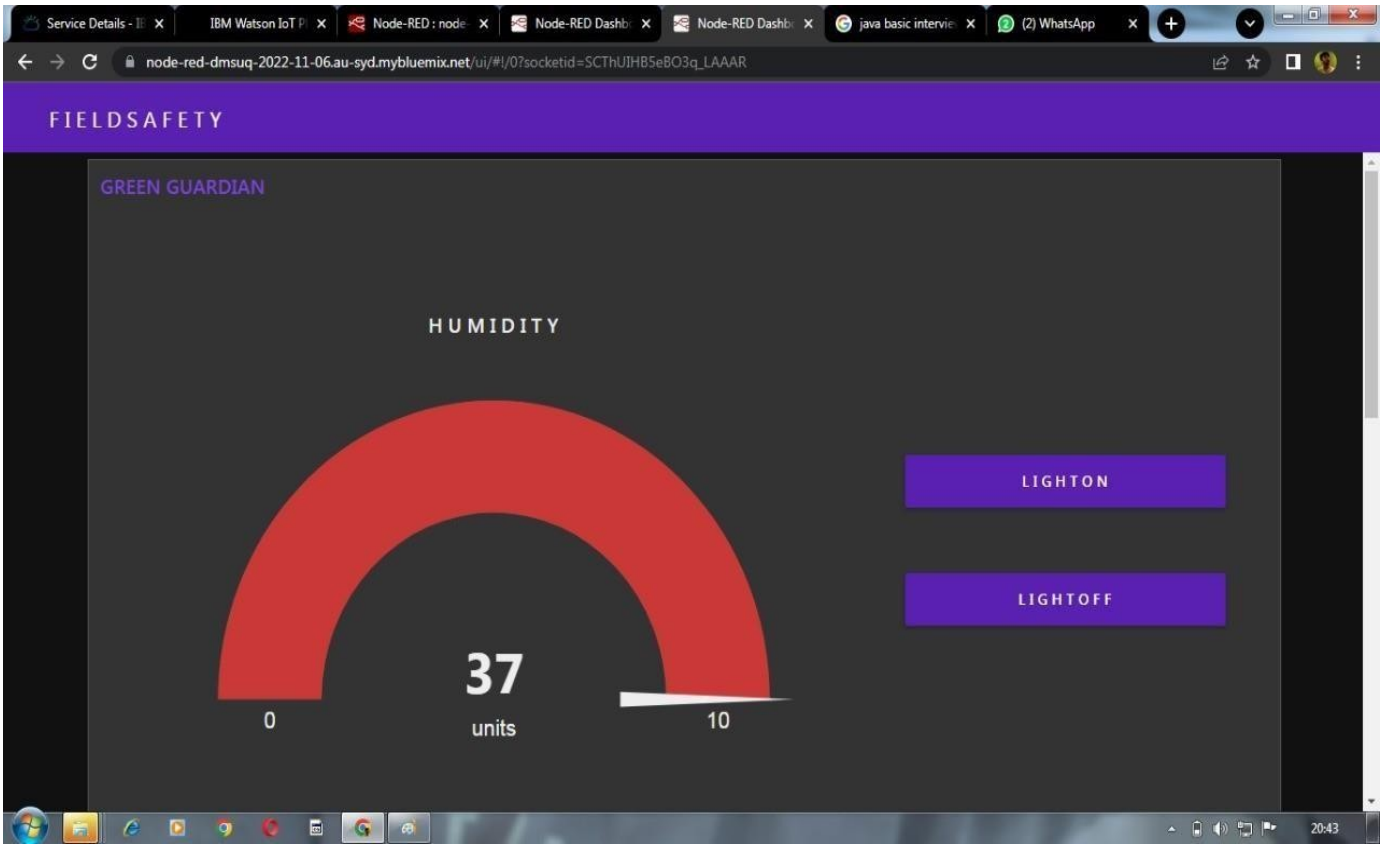
Recent Events:

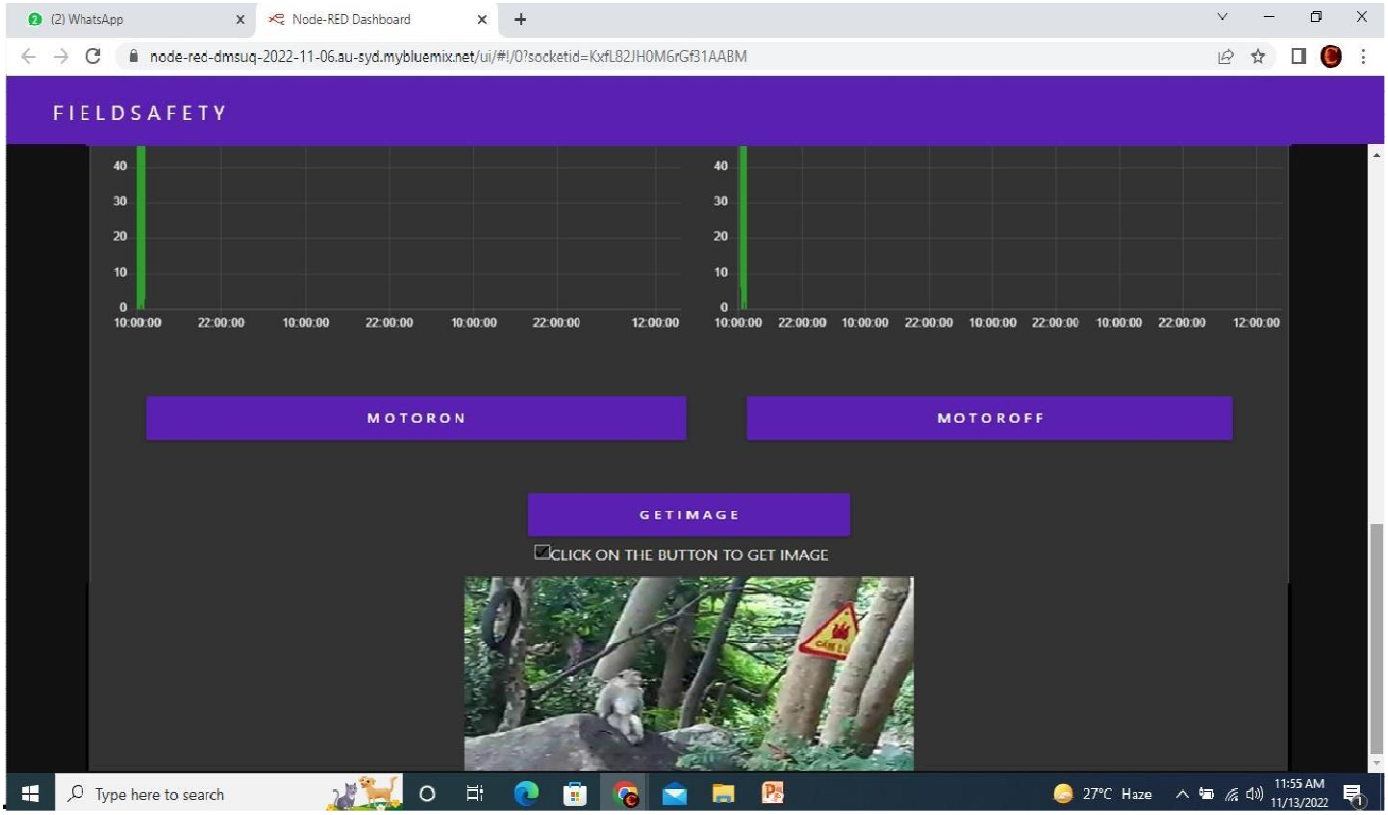


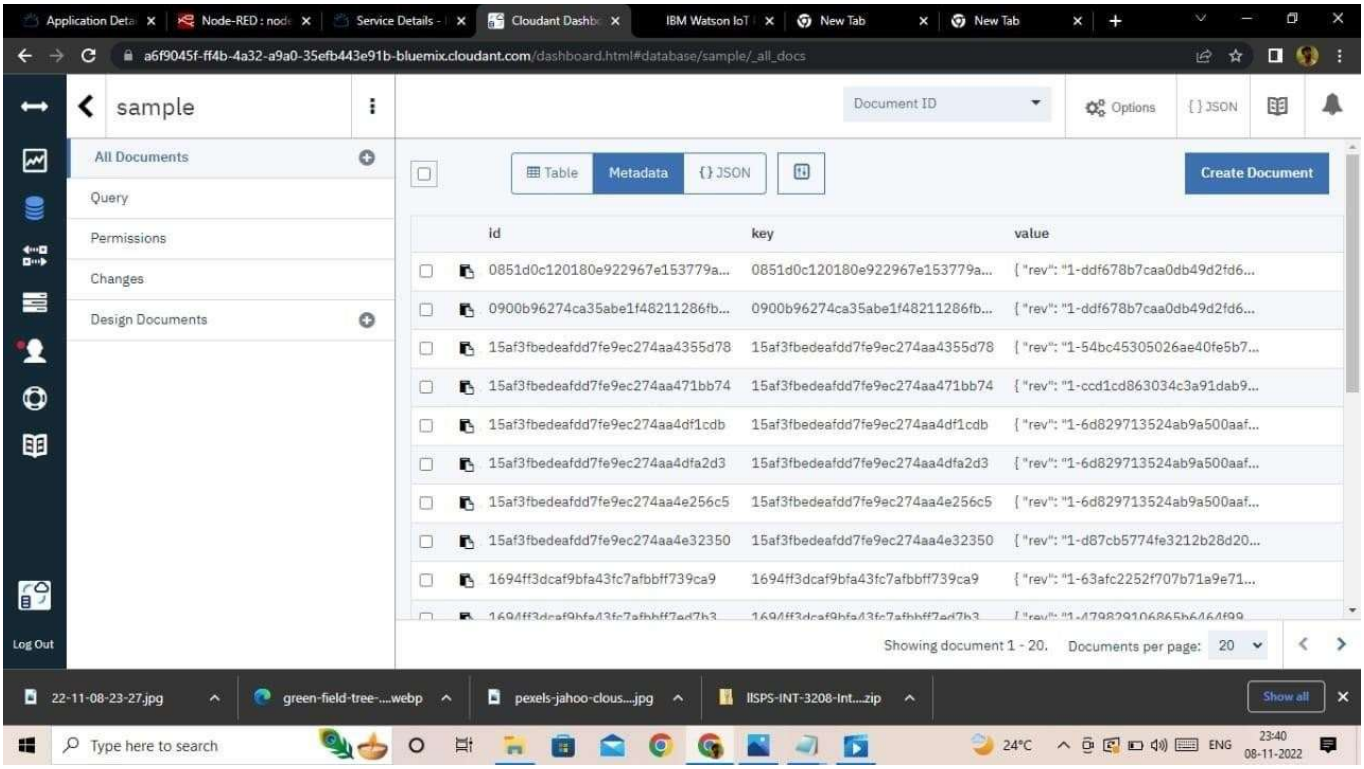
Node-Red Connection and Dashboard Design:

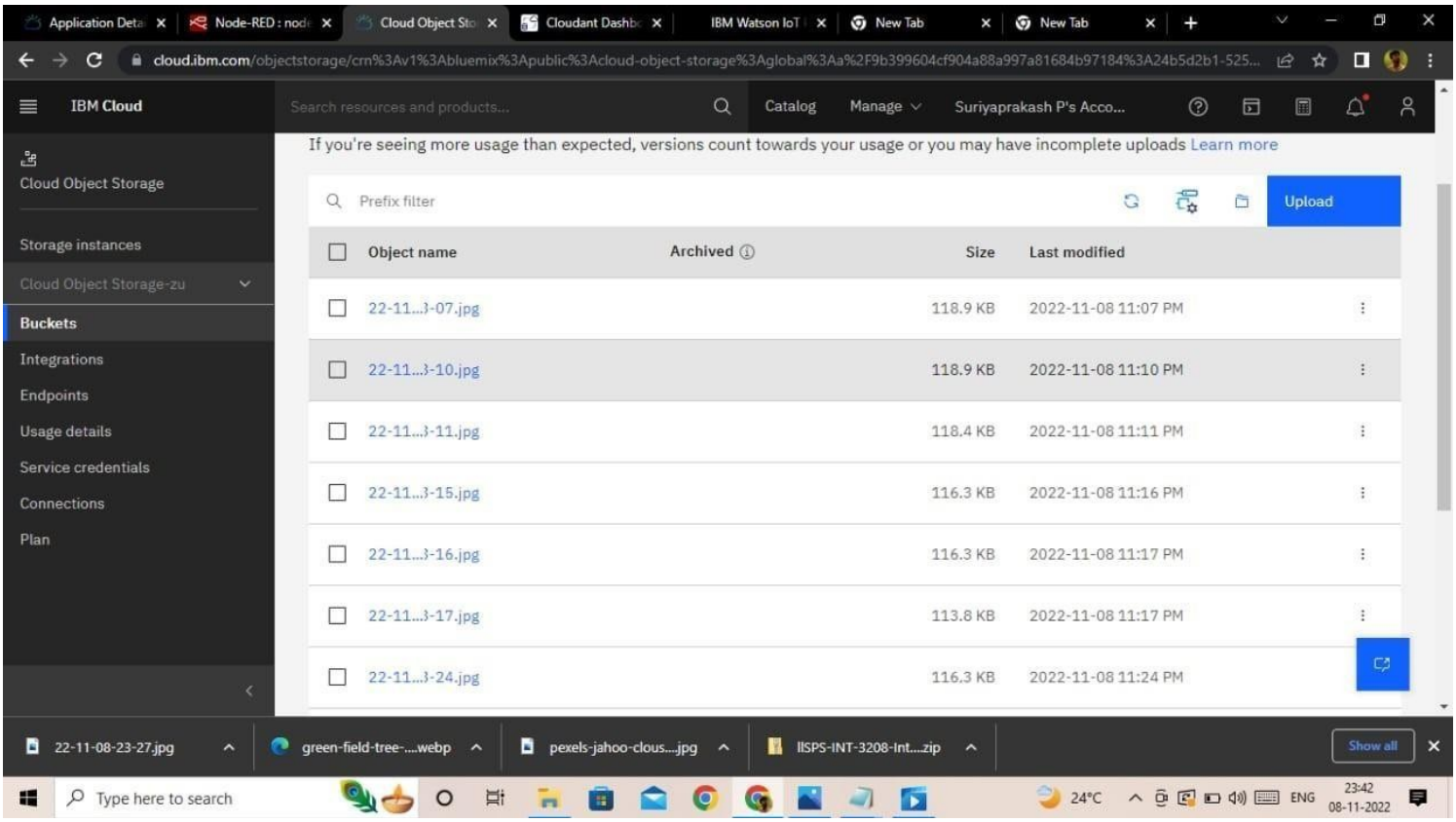



Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\new\app.py
database.py
multipartupload.py
client.py
app.py
print('File not found')
while(cap.isOpened()):
ret, frame = cap.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
imS = cv2.resize(frame, (960, 540))
cv2.imwrite('ex.jpg',imS)
with open("ex.jpg", "rb") as f:
file_bytes = f.read()
print("Alert! Alert! Animal detected")
picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
cv2.imwrite(picname+'.jpg',frame)
multi_part_upload("karthi01", picname+'.jpg', picname+'.jp
json_document={"link":COS_ENDPOINT+'/'+'karthi01'+'/'+picn
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)
temperture=random.randint(0,100)
myData={'Animal': detect,'moisture':moist,'humidity':humidity,'
print(myData)
if(humidity!=None):
client.publishEvent(eventId="status", msgFormat="json", data
print("Publish Ok..")
client.commandCallback = myCommandCallback
cv2.imshow('frame',imS)
if cv2.waitKey(1) & 0xFF == ord('q'):
break
client.disconnect()
cap.release()
cv2.destroyAllWindow()
Name
Type
Size
Value
cap
VideoCapture
1
VideoCapture object of cv2 module
Help
Variable Explorer
Plots
Files
Console 1/A
{'Animal': True, 'moisture': 47, 'humidity': 23, 'temperture': 41}
2022-11-09 10:08:06,806 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:4wq3lx:raspberrypi:demo123
2022-11-09 10:08:07,114 wiotp.sdk.device.client.DeviceClient ERROR Unexpected
disconnect from IBM Watson IoT Platform: 7
2022-11-09 10:08:07,115 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:4wq3lx:raspberrypi:demo123
Publish Ok..
Alert! Alert! Animal detected
Starting file transfer for22-11-09-10-08.jpg to bucket: karthi01
Unable to complete multi-part upload: Replacement index 1 out of range for positional args tuple
Document successfully created.
2022-11-09 10:08:12,953 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:4wq3lx:raspberrypi:demo123
2022-11-09 10:08:12,953 wiotp.sdk.device.client.DeviceClient ERROR Unexpected
disconnect from IBM Watson IoT Platform: 7
2022-11-09 10:08:15,999 wiotp.sdk.device.client.DeviceClient ERROR Unexpected
disconnect from IBM Watson IoT Platform: 7
2022-11-09 10:08:16,093 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:4wq3lx:raspberrypi:demo123
{'Animal': True, 'moisture': 71, 'humidity': 44, 'temperture': 65}
2022-11-09 10:08:19,844 wiotp.sdk.device.client.DeviceClient ERROR Unexpected
disconnect from IBM Watson IoT Platform: 7
2022-11-09 10:08:19,848 wiotp.sdk.device.client.DeviceClient INFO Connected
successfully: d:4wq3lx:raspberrypi:demo123
IPython Console
History
LSP Python: ready
conda: base (Python 3.9.13)
Line 109, Col 17
ASCII
CRLF
RW
Mem 89%
77%
27°C Cloudy
ENG
10:14


Service Details -
IBM Watson IoT P
Node-RED : node
Node-RED Dashb
Node-RED Dashb
java basic intervie
(2) WhatsApp
node-red-dmsuq-2022-11-06.au-syd.mybluemix.net/ui/#!/0?socketid=SCThUIHB5eBO3q_LAAAR
FIELDSAFETY
GREEN GUARDIAN
HUMIDITY
37
units
0
10
LIGHTON
LIGHTOFF
20:43


(2) WhatsApp
Node-RED Dashboard
node-red-dmsuq-2022-11-06.au-syd.mybluemix.net/ui/#!/0?socketid=KxfL82JH0M6rGf31AABM
FIELDSAFETY
40
30
20
10
0
10:00:00
22:00:00
12:00:00
MOTORON
MOTOROFF
GETIMAGE
CLICK ON THE BUTTON TO GET IMAGE
Type here to search
27°C Haze
11:55 AM
11/13/2022


Application Deta
Node-RED : node
Service Details -
Cloudant Dashb
IBM Watson IoT
New Tab
New Tab
a6f9045f-ff4b-4a32-a9a0-35efb443e91b-bluemix.cloudant.com/dashboard.html#database/sample/_all_docs
sample
All Documents
Query
Permissions
Changes
Design Documents
Log Out
Document ID
Options
{ } JSON
Table
Metadata
{ } JSON
Create Document
id
key
value
0851d0c120180e922967e153779a...
0900b96274ca35abe1f48211286fb...
15af3fbedeafdd7fe9ec274aa4355d78
15af3fbedeafdd7fe9ec274aa471bb74
15af3fbedeafdd7fe9ec274aa4df1cdb
15af3fbedeafdd7fe9ec274aa4dfa2d3
15af3fbedeafdd7fe9ec274aa4e256c5
15af3fbedeafdd7fe9ec274aa4e32350
1694ff3dcaf9bfa43fc7afbbff739ca9
Showing document 1 - 20.
Documents per page:
20
22-11-08-23-27.jpg
green-field-tree-....webp
pexels-jahoo-clous....jpg
IISPS-INT-3208-Int....zip
Show all
Type here to search
24°C
ENG
23:40
08-11-2022


Application Deta
Node-RED : node
Cloud Object Sto
Cloudant Dashbo
IBM Watson IoT
New Tab
New Tab
cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F9b399604cf904a88a997a81684b97184%3A24b5d2b1-525...
IBM Cloud
Search resources and products...
Catalog
Manage
Suriyaprakash P's Acco...
Cloud Object Storage
Storage instances
Cloud Object Storage-zu
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads Learn more
Prefix filter
Upload
Object name
Archived
Size
Last modified
22-11...3-07.jpg 118.9 KB 2022-11-08 11:07 PM
22-11...3-10.jpg 118.9 KB 2022-11-08 11:10 PM
22-11...3-11.jpg 118.4 KB 2022-11-08 11:11 PM
22-11...3-15.jpg 116.3 KB 2022-11-08 11:16 PM
22-11...3-16.jpg 116.3 KB 2022-11-08 11:17 PM
22-11...3-17.jpg 113.8 KB 2022-11-08 11:17 PM
22-11...3-24.jpg 116.3 KB 2022-11-08 11:24 PM
22-11-08-23-27.jpg
green-field-tree-....webp
pexels-jahoo-clous....jpg
IISPS-INT-3208-Int....zip
Show all
Type here to search
24°C
ENG
23:42
08-11-2022

CHAPTER-8

TESTING

8.1Test cases

Test Case	Test Scenario	Test Data	Status	Comments	Executed by
TC_OO1	Create the IBM Cloud services which are being used in this project	https://cloud.ibm.com /login	Pass	Results verified	**Pandiyarajan A**
TC_OO2	Configure the IBM Cloud services which are being used in completing this project.	https://cloud.ibm.com /login	Pass	Results verified	Abeesh k
TC_OO3	IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform	https://4wq3lx.internetofthings. ibmcloud.com/dashboard/ devices/browse	Pass	Results verified	Karthikeyan A
TC_OO4	In order to connect the IoT device to the IBM cloud create a device in the IBM Watson IoT platform	Temperature, Humidity , Soil moisture sensor values are generated randomly in simulation	Pass	Results verified	Karthikeyan A
TC_OO5	Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT platform	https://cloud.ibm.com /developer/appservice /create- app?starterKit=59c9d5 bd-4d31-3611-897a- f94eea80dc9f&default	Pass	Results verified	Suriyaprakesh p
TC_OO6	Create a Node-RED service.	Values of sensors and button for light ON/OFF is displayed	Pass	Results verified	Suriyaprakesh p
TC_OO7	publish random sensor data such as temperature, humidity level, soil moisture to the IBM IoT platform	https://www.python.org/ downloads/release /python-370/	Pass	Results verified	Abeesh k

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. 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	7	3	6	5	21
Duplicate	4	0	3	0	7
External	1	2	0	1	4
Fixed	14	1	3	8	26
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	4	2	0	6
Totals	26	11	18	19	67

3. 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	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	1	0	0	1
Exception Reporting	7	0	0	7
Final Report Output	9	0	0	9
Version Control	1	0	0	1

CHAPTER-9
RESULTS

9.1 Performance Metrics

1. Requirement Identification
    - Functional Requirements
    - Non-Functional Requirements
2. Implementation result
    - System Implementation results
    - Results of web application Implementation
3. Resource utilization results
    - Foreground activities results
    - Memory usage
    - Energy usage
4. Background activities results

CHAPTER-10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGE:

- ✓ Easily detect the problems in the field.
- ✓ Most Accurate
- ✓ Flexible Model which can give maximized outcome
- ✓ No Specific Requirements needed to implement the model

10.2 DISADVANTAGE:

- ✓ Wind can wreak havoc on sprinklers, directing water in the wrong direction.
- ✓ The cost of maintenance becomes high whether there is a repair or not.
- ✓ If there are faulty data processing equipment or sensors then it will lead to the situation where the wrong decisions are taken.

CHAPTER-11

CONCLUSION

In this project, we proposed a method for efficient crop monitoring for agricultural field. With the application of IOT the datas can be stored and retrieved from anywhere. In this proposed work, the sensor part is limited only for monitoring of crop. hence in future it can be automated for irrigation and the system can be enhanced with security of farm land under video surveillence which prevents it from obtrude intrusion.

CHAPTER-12

FUTURE SCOPE

In the current project we have implemented the project that can protect and maintain the crop. In this project the farmer monitor and controlthe field remotely. In future we can add or update few more things to this projectWe can create few more models of the same project ,so that thefarmer can have information of a entire.We can update the this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. Itreduces the power line cost. It will be a one time investment. We can add solar fencing technology to this project.We can use GSM technology to this project so that the farmers can get the information directly to his home through SMS. This helps the farmerto get information if there is a internet issues.

CHAPTER-13

APPENDIX

Github : http://bit.ly/3GythyG

Demo Link : http://bit.ly/3Ef0H1q