EPROJECT REPORT
IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
TEAM ID:PNT2022TMID37069

TABLE OF CONTENTS

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

1. Feature 1
2. Feature 2

8. TESTING

1. Test Cases
2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1.INTRODUCTION

1.1 PROJECT OVERVIEW:

Food plays a major role in human life, without food no one can live in this world. In that food cultivation agriculture plays main role, but now a days due to many climate conditions, insufficient of labours, improper irrigation, animal attacks, lack of knowledge on fertilizers among farmers and some other threats crops are not grown properly. To solve this the automation and IOT is implemented to maintain crops in the crop land so farmer can maintain crop from remote place that is anywhere at any time.

For this we have use many sensors to get readings of some factors and which is connected to IOT cloud through the gateway that is Raspberry PI, through that cloud the user can receive the sensor readings in mobile app and also he can controll the sprinkler or motor in the cropland. By use of raspberry PI also authomation is implemented by sensor readings and that is updated to the farmer.

1.2 PURPOSE:

The purpose of the project is to protect crops from various problems like climate condition, insufficient labour force, improper irrigation, animal attacks, etc. Hence the crop growth will be proper, production will be more and overcome the loses from farmer side.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM:

- **Improper Irrigation:**Crops are not irrigated properly due to insufficient labour forces. Hence crops are damaged and growth is bad.
- **improper maintanance:**Improper maintenance of crops against various environmental factors such as temperature climate, topography and soil quality which results in crop destruction.
- **Lack of knowledge on fertilizers:**Lack of knowledge among farmers in usage of fertilizers and hence crops are affected due to high ammonia, urea, potassium and high PH level fertilizers.
- **Wild attacks:**Requires protecting crops from Wild animals attacks, birds and pests.

2.2 REFERENCES:

1.2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC) 978-1-7281-4889-2/20/$31.00 ©2020 IEEE 10.1109/ICCMC48092.2020.ICCMC-00076

2.2020 Seventh International Conference on Information Technology Trends (ITT)|978-1-7281-83794/20/$31.00©2020IEEE|DOI:10.1109/ITT51279.2020.9320783

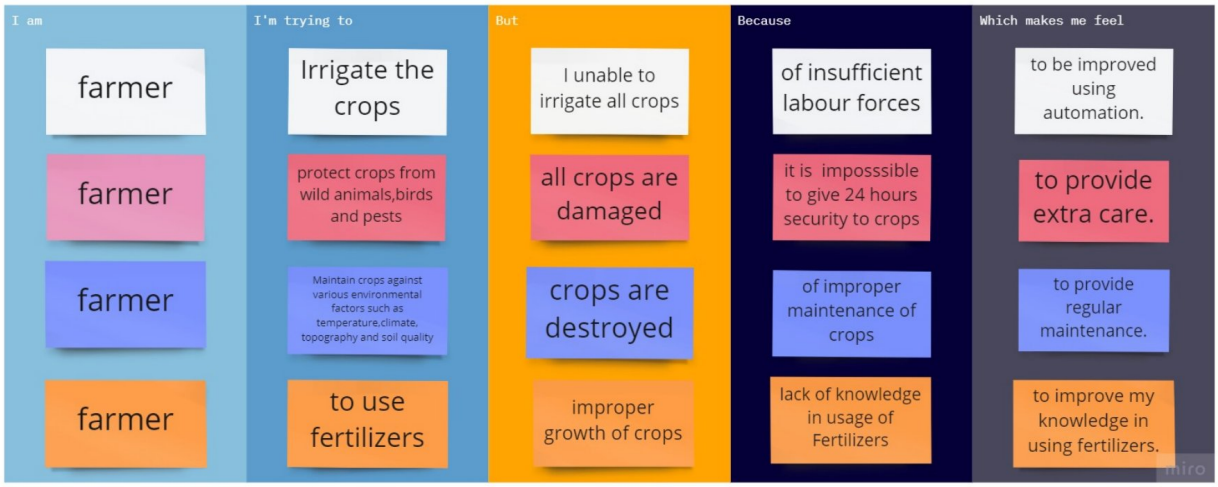
3.2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC) | 978-1-7281-1110-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/R10-HTC49770.2020.9357012

4.Solving the Flexible Job Shop Scheduling ProblemWith Makespan Optimization by Using a Hybrid Taguchi-Genetic Algorithm - Digital object identifier - 10.1109/ACCESS.2021.311403

2.3 PROBLEM STATEMENT DEFINITION:

I am Farmer .**I am trying to** irrigate my crops, protect my crops from various hard climate conditions, wild attacks, and improper maintanence. **But** crops are destroyed. **Because** of improper maintence, improper irrigation, insufficient labour force and lack of knowledge on fertilizers. **Which makes me feel** to improve protection, to implement automation and to improve my knowledge on fertilizers.

Link:
https://miro.com/welcomeonboard/SjdIMXB4S0puek1Yd2ZGQjduNDFUMlVPRDNhMW44NjBEOFhHbWUwYWRMYUxBUkNrbDV3NGL7s3x7ExBwVBozsLq13wPWwAwTSg6nMTMxNzM2MzQ2fDI=?share_link_id=928357036680

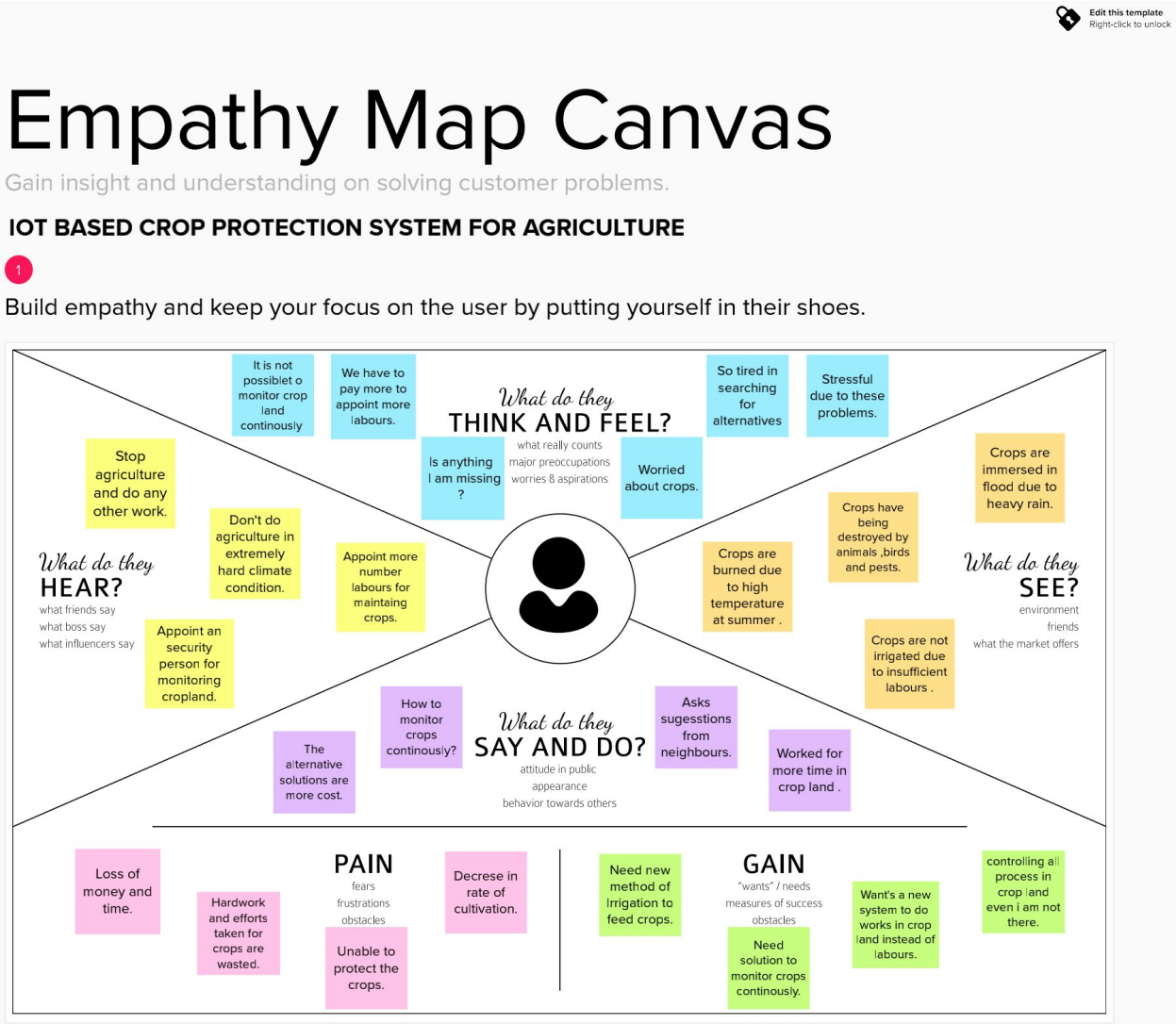


3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS:

Using empathy map we empathized the farmers problem which they are facing by putting ourself into there shoes. Hence the real problem has been came to know. The empathy map created by our team is shown below.

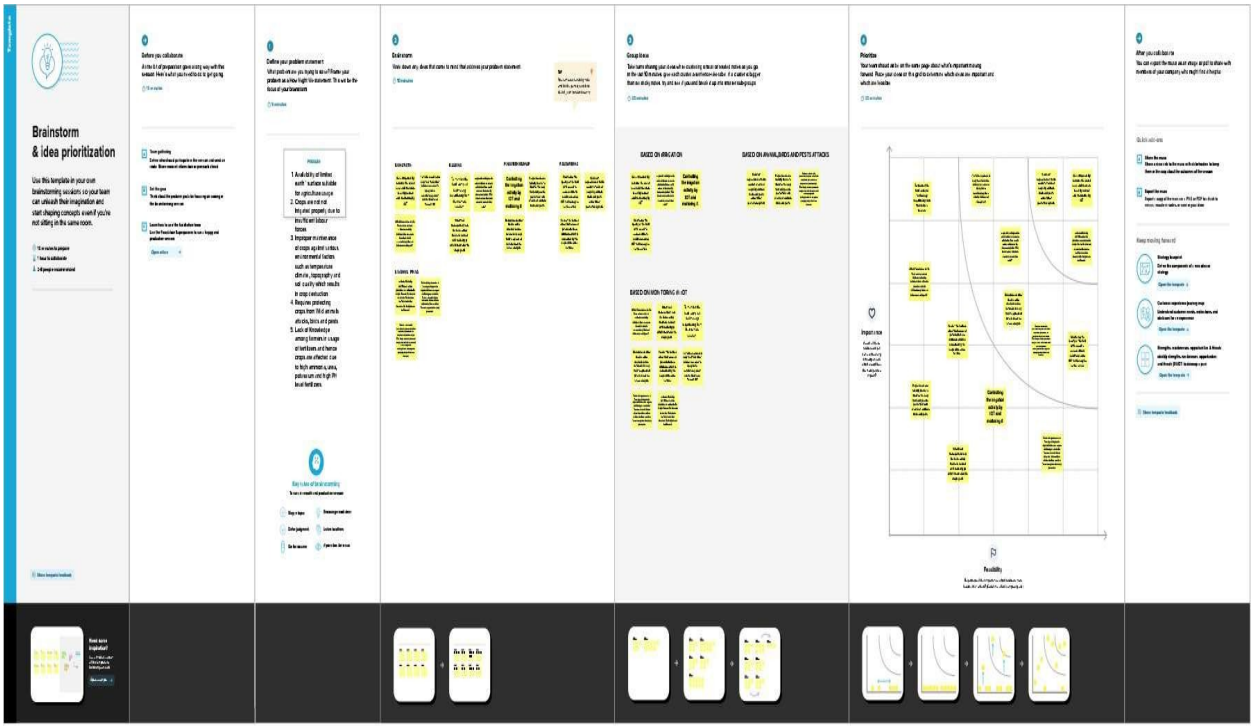
LINK:
https://app.mural.co/invitation/mural/bharathiandteam3000/1661766407564?sender=ubbb704f08dea51679dce8101&key=d851938e-20cf-49e9-83fc-55cca3669f4f



3.2 IDEATION AND BRAINSTORMING:

The problems determined by the empthy map can be overcomed by using the ideation and brainstorming. In this we team members are discussed on the problems and created some ideas while brainstorming session. and the top three ideas are choosen to solve the Farmer problem. The brainstorming and idea prioritizing Template has showed below.

Link:
https://app.mural.co/invitation/mural/bharathiandteam3000/1663332972819?sender=ubbb704f08dea51679dce8101&key=818973e5-8ca6-4ff8-afe6-de9a0d8fbd8e



3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	● Crops are not irrigated properly due to insufficient labour forces. ● Improper maintenance of crops against various environmental factors such as temperature climate, topography and soil quality which results in crop destruction. ● Lack of knowledge among farmers in usage of fertilizers and hence crops are affected due to high ammonia, urea, potassium and high PH level fertilizers. ● Requires protecting crops from Wild animals attacks, birds and pests.
2.	Idea/Solution description	● Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF the motor pump for managing the excess water level. It will be updated to authorities through IOT. ● Temperature sensor connected to microcontroller is used to monitor the temperature in the field. The optimum temperature required for crop cultivation is maintained using sprinklers. ● IOT based fertilizing methods are followed, to minimize the negative effects on growth of crops while using fertilizers. ● Image processing techniques with IOT is followed for crop protection against animal attacks.
3.	Novelty / Uniqueness	● Automatic crop maintenance and protection using embedded and IOT technology.
4.	Social Impact / Customer Satisfaction	● This proposed system provides many facilities which helps the farmers to maintain the crop field without much loss.
5.	Business Model (Revenue Model)	● This prototype can be developed as product with minimum cost with high performance .
6.	Scalability of the Solution	● This can be developed to a scalable product by using sensors and transmitting the data through Wireless Sensor Network and Analysing the data in cloud and operation is performed using robots

3.4 PROBLEM SOLUTION FIT:

1.CUSTOMER SEGMENT

- Farmers who trying to protect crops from various problems

2.PROBLEM/PAINS

- Crops are not irrigated properly.
- Improper maintenance of crops.
- Lack of knowledge among farmers in usage of fertilizers and hence crops are affected.
- Requires protecting crops from Wild animals attacks, birds and pests.

3.TRIGGERS TO ACT

- By seeing surrounding crop land with installing machineries.
- Hearing about innovative technologies and effective solutions.

4.EMOTIONS

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

5.AVAILABLE SOLUTIONS

- Automation in irrigation.
- CCTV camera to monitor and supervise the crops.
- Alarm system to give alert while animals attacks the crops.

6.CUSTOMER LIMITATION

- Limited supervision.
- Limited financial constrains.
- Lack of man power.

7.BEHAVIOUR

- Asks suggestions from surrounding peoples and implement the recent technologies.
- Consumes more time in crop land.
- Searching for an alternative solution for an existing solution.

8.CHANNELS OF BEHAVIOUR

online:

- Using different platforms /social media to describe the working and uses of smart crop protection device.

offline:

- Giving awareness among farmers about the application of the device.

9.PROBLEM ROOT/CAUSE

- Due to insufficient labour forces.
- Due to various environmental factors such as temperature climate, topography and soil quality which results in crop destruction.
- Due to high ammonia, urea, potassium and high PH level fertilizers.
- Crops are damaged and it affects growth.

10.OUR SOLUTION

- Moisture sensor is interfaced with Arduino Microcontroller to measure the moisture level in soil and relay is used to turn ON and OFF the motor pump for managing the excess water level. It will be updated to authorities through IOT.
- Temperature sensor connected to microcontroller is used to monitor the temperature in the field. The optimum temperature required for crop cultivation is maintained using sprinklers.
- IOT based fertilizing methods are followed, to minimize the negative effects on growth of crops while using fertilizers.
- Image processing techniques with IOT is followed for crop protection against animal attacks.

Link:https://ibb.co/xFnTXWx

The Above Link is for Problem solution fit template

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Hardware requirements	● Raspberry pi, Camera module, Temperature sensor, Flame sensor, PH level sensors, Water and Moisture level sensor, Motor pump and Router.
FR-2	Software requirements	● Python Software is used to program the hardware.
FR-3	Internet cloud access	● IBM cloud service.
FR-4	User interface	● Mobile Application created by MIT app inverter, NODE RED.

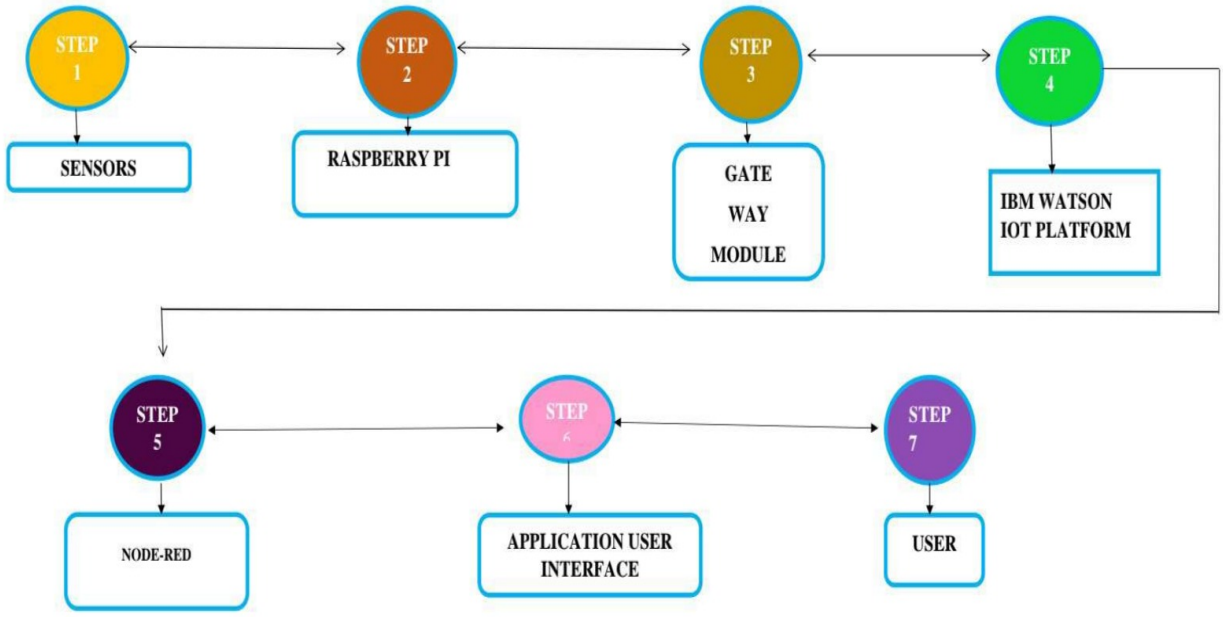
4.2 NON FUNCTIONAL REQUIREMENTS:

NFR NO	Non-Functional Requirement	Description
NRF-1	Usability	● The proposed smart crop protection device is designed using recent technology which is used to face the challenges in agriculture with automatic farming practices.
NFR-2	Security	● It should be more secure to avoid unwanted interferences
NFR-3	Reliability	● The developed system is more reliable such that it satisfies the needs of farmers and found easy handling.
NFR-4	Performance	● Performance of the device will be optimum.
NFR-5	Availability	● The device is cost effective and simple design. Hence it is easily available in market and installation is easy.
NFR-6	Scalability	● This should be developed to scalable product by using sensors and transmitting the data through wireless sensors network and analysing the data in cloud and operation is performed using robots.

5.PROJECT DESIGN

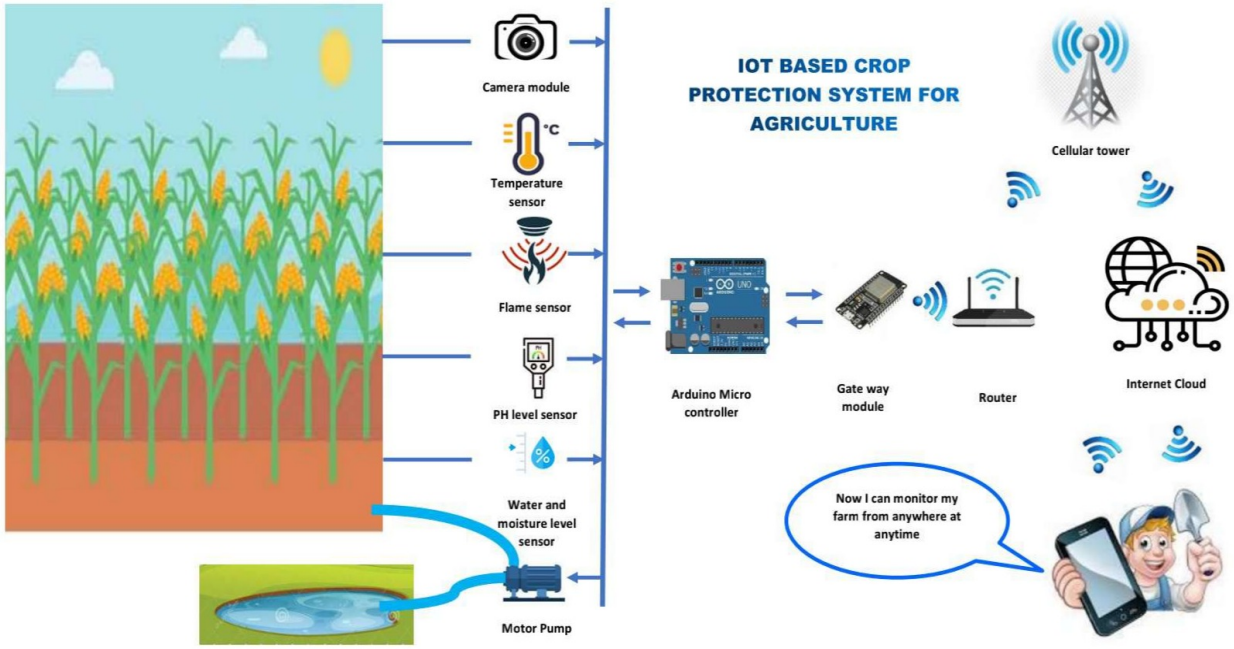
5.1 DATAFLOW DIAGRAM:

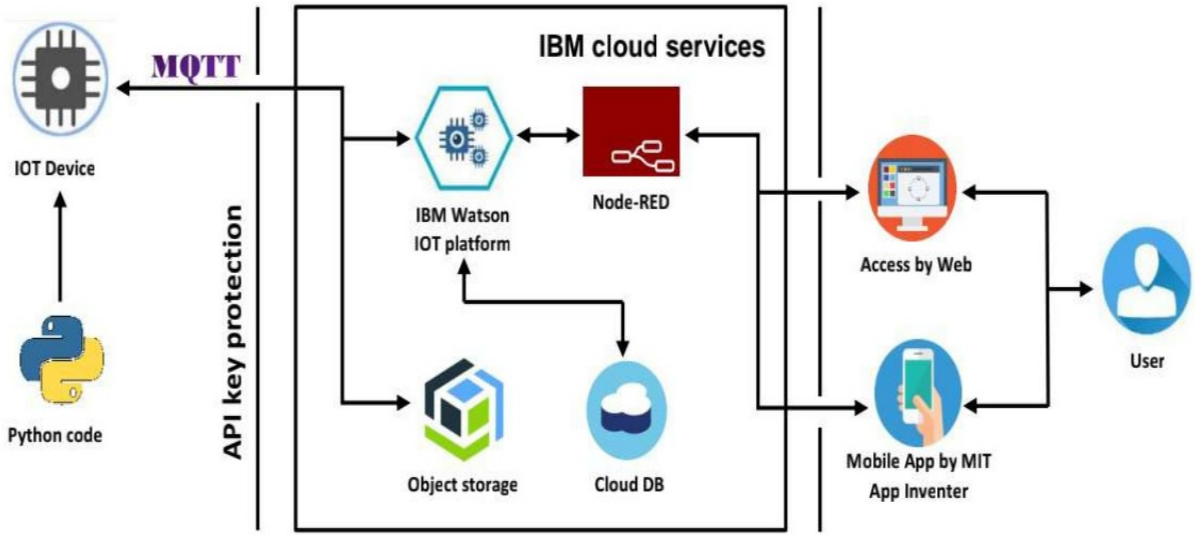
This Data flow diagram shows the data flow from IOT device to user interface application.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE:

Based on the Ideas the solution architecture and technology architecture is shown





5.3 USER STORIES:

Here the user is farmer so what are the user stories if the project is implemented are mention below.

- As a farmer I can see the farm is automated using sensors. Hence insufficient labour problem has been solved.
- As a farmer I can monitor my farm from anywhere or without going to farm.
- As a farmer I can controll the motors and sprinkler in my farm from remote place.
- an user interface applicaton is used to all datas of sensor from the farm.

6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING NO SCHEDULING:

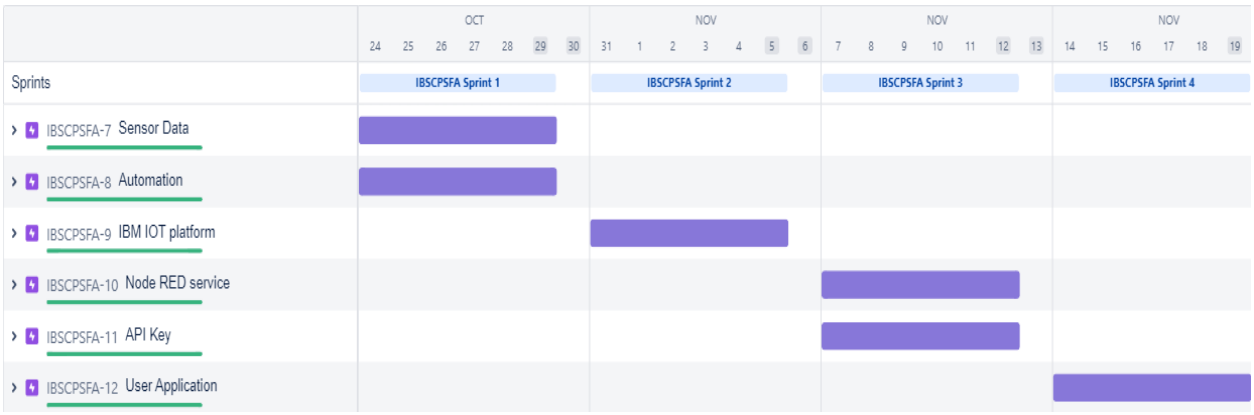
Sprint	Functional Requirement (Epic)	User Story Number	User story/task	Story Points	Priority	Team Members
Sprint 1	Sensor Data (python script)	USN-1	The Data of sensor which are feed to the Raspberry pi. Here we are using python script to generate a random sensor data.	3	High	S. BHARATHI (Team leader)
sprint 1	Automation (python script)	USN-2	Some activities are made to automation to overcome insufficient of labour force in the field. Hence that also included in python script to implement automation in the.	5	High	S. BHARATHI (Team leader)
sprint 2	IBM IOT platform	USN-3	To send the raspberry pi data to IOT platform, we create an IBM IOT platform and connect the raspberry pi to the device created in IBM IOT.	5	High	R. NAVEEN KUMAR (Team member-2)
sprint 3	Node RED service	USN-4	To access the IBM IOT platform from external application or from external UI Node red service is established.	5	High	K. LUBNA (Team Member-1)
sprint 3	API Key	USN-5	To protect the IBM IOT platform creating an API Key.	3	High	K. LUBNA (Team Member-1)
sprint 4	User Application	USN-6	To monitor and control the field sensors the User is provided with an User application created by MIT app inventor	8	High	R. SASIREKHA (Team Member-3), B. VISHNU PRIYA (Team Member-4)

6.2 SPRINT DELIVERY SCHEDULE:

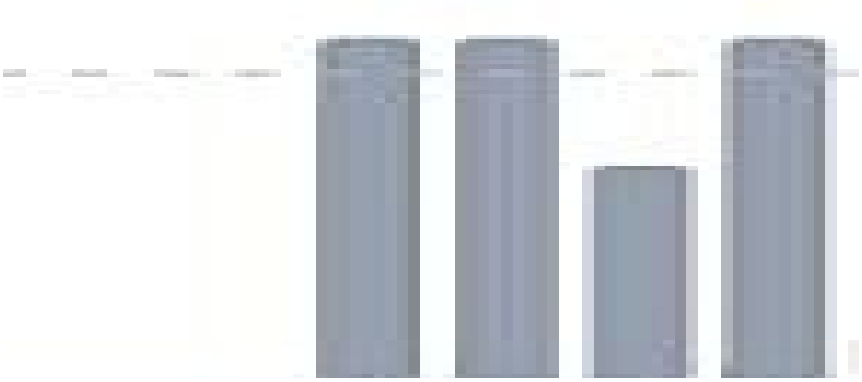
Sprint	Total story points	Duration	Sprint start Date	Sprint End Date	Story Points completed as palnned	Sprint Release Date
Sprint-1	8	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	5	6 Days	31 Oct 2022	5 Nov 2022	5	5 Nov 2022
sprint-3	8	6 Days	07 Nov 2022	12 Nov 2022	8	12 Nov 2022
sprint-4	8	6 days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

6.3 REPORTS FROM JIRA:

1.ROAD MAP:



2.BURNDOWN CHART:



3.VELOCITY:

$$\text{VELOCITY} = \frac{\textbf{29(no of sprint points)}}{\textbf{24(no of sprint days)}} = \textbf{1.2}$$

7.CODING AND SOLUTIONING

7.1 FEATURE 1:

In this project, The sensors are fixed in the farm land (Temperature Sensor, PH Sensor, Camera Module, Flame Sensor, Moisture Sensor and Water level sensor). From that sensors, values are read by use of rasperry pi and the rasperrry pi act as a gate way module to send the sensor data to cloud(Here we use python code for simulation instead of rasperry pi). And also the automation are implement in python code.

7.2 FEATURE 2:

The datas which sent by raspberry pi (Python Script) is received in IBM IOT platform is were we have created an organization and device to view the data in back end.

7.3 FEATURE 3:

The datas received in the IBM IOT are get through node red service to create an UI to visualize the data and to create data flow to user inerface application.

7.4 FEATURE 4:

The mobile application is created to view the dash board created using the Node red service. this will more user friendly.

8.TESTING

8.1 TEST CASES:

NFT RISK ASSESMENT

s.no	project name	Scope /feature	Functional changes	Hardware Changes	Software changes	Load/volume changes	Risk score	Justification
1.	IOT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE	To protect crops from various environmental issues and improper maintanence	Low	No changes	No changes	Low	Green	Already the project running with more load but it running properly . Hence separate testing is not needed for our project. Direct manual checking and running the project is going to be done. And performance metrics will be shown.

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	3	2	0	1	6
Duplicate	1	0	0	0	1
External	2	3	0	1	6
Fixed	2	1	0	1	4
Not Reproduced	0	0	0	0	0
Skipped	0	0	1	0	1
Won't Fix	0	2	0	0	2
Total	8	7	1	3	19

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	10	0	0	10
Client Application	15	0	0	15
Security	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4

9.RESULTS

9.1 PERFORMANCE METRICS:

1.IOT DEVICE(PYTHON SCRIPTS):

The sensor datas are generated randomly through the python code and the automation is implented with the help of python code. The sensor datas are published succesfully to the cloud and the aknowledgement is printed in the output. Also the akerts and automation datas are published succesfully.

2.IBM IOT PLATFORM:

The datas which published published are view in the IBM IOT platform, in that the device is connected succesfully and the datas are viewed in the recent events.

3.NODE RED SERVICE:

The datas which sent to the ibm platform are received in IBM IOT node in node red and the functions are created to view the visual data. The datas are succesfully visualized in NODE RED UI.

4.MOBILE APPLICATION:

The node red UI is created as an mobile application and the datas are seened in the Mobile application.

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Crops are protected from various climate conditons and wild attacks.
- Crops are irrigated even labours are insufficient.
- Through the IOT technology farmer can monitor the farm without going to crop land.
- Hence crops are protected so increase in production helps to get more profit for farmers.

DISADVANTAGES:

- Quality internet service is required.
- User should use that app in mobile background every time

11.CONCLUSION

Through the rasperry pi we have automated the whole system hence this is an power efficient system, that is the raspberry pi can run by minimimum of 5V. the IOT Technology is used to monitor and protect the crop from various issues like climate, insufficient labour, wild attacks and lack of knowledge in fertilizers. and also the system is cost effective.

12. FUTURE SCOPE

In future the system will be controlled and monitored through the Artificial intelliengce Hence the performance will be improved. And also robots are installed to make the system more efficient.

13.APPENDIX

PYTHON SOURCE CODE:

```
#importing Libraries:

import random
import ibmiotf.application
import ibmiotf.device
from time import sleep
import sys


#IBM Watson Device Credentials.

organization = "zf801i"
deviceType = "bharathi"
deviceId = "bharathi123"
authMethod = "token"
authToken = "123456789"


def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="sprinkler_on":
      print ("sprinkler is ON")
   else :
      print ("sprinkler is OFF")

   #print(cmd)




try:
                deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
                deviceCli = ibmiotf.device.Client(deviceOptions)


except Exception as e:
                print("Caught exception connecting device: %s" % str(e))
                sys.exit()
```

```
#Connecting to IBM watson.
deviceCli.connect()



while True:

#Getting values from sensors.

    temp_sensor = round( random.uniform(0,80),2)
    PH_sensor = round(random.uniform(1,14),3)
    camera = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    camera_reading = random.choice(camera)
    flame = ["Detected","Not Detected","Not Detected","Not Detected","Not Detected","Not Detected",]
    flame_reading = random.choice(flame)
    moist_level = round(random.uniform(0,100),2)
    water_level = round(random.uniform(0,30),2)



#storing the sensor data to send in json format to cloud.

    temp_data = { 'Temperature' : temp_sensor }
    PH_data = { 'PH_Level' : PH_sensor }
    camera_data = { 'Animal_attack' : camera_reading}
     flame_data = { 'Flame' : flame_reading }
    moist_data = { 'Moisture_Level' : moist_level}
    water_data = { 'Water_Level' : water_level}



 # publishing Sensor data to IBM Watson for every 5-10 seconds.

    success = deviceCli.publishEvent("Temperature sensor", "json", temp_data, qos=0)
    sleep(1)
    if success:
        print ("   .....publish ok.....   ")
        print ("Published Temperature = %s C" % temp_sensor,  "to IBM Watson")



     success = deviceCli.publishEvent("PH sensor", "json", PH_data, qos=0)
```

```
    sleep(1)
    if  success:
        print ("Published PH Level = %s" % PH_sensor,  "to IBM Watson")


    success = deviceCli.publishEvent("camera", "json", camera_data, qos=0)
    sleep(1)
    if success:
       print ("Published Animal attack %s " % camera_reading,  "to IBM Watson")


    success = deviceCli.publishEvent("Flame sensor", "json", flame_data, qos=0)
    sleep(1)
    if success:
        print ("Published Flame %s " % flame_reading,  "to IBM Watson")


    success = deviceCli.publishEvent("Moisture sensor", "json", moist_data, qos=0)
    sleep(1)
    if success:
         print ("Published Moisture Level = %s " % moist_level,  "to IBM Watson")


    success = deviceCli.publishEvent("Water sensor", "json", water_data, qos=0)
    sleep(1)
    if success:
        print ("Published Water Level = %s cm" % water_level,  "to IBM Watson")
        print ("")




    #Automation to control sprinklers by present temperature an to send alert message to IBM Watson.

    if (temp_sensor > 35):
       print("sprinkler-1 is ON")
          success = deviceCli.publishEvent("Alert1", "json",{ 'alert1' : "Temperature(%s) is high, sprinkerlers
are turned ON" %temp_sensor } , qos=0)
       sleep(3)
       if success:
                       print( 'Published alert1 : ',  "Temperature(%s) is high, sprinkerlers are turned ON"
%temp_sensor,"to IBM Watson")
            print("")

    else:
```

```
        print("sprinkler-1 is OFF")
        print("")


    #To send alert message if farmer uses the unsafe fertilizer to crops.

    if (PH_sensor > 7.5 or PH_sensor < 5.5):
            success = deviceCli.publishEvent("Alert2", "json",{ 'alert2' : "Fertilizer PH level(%s) is not safe,use
other fertilizer" %PH_sensor } , qos=0)
        sleep(3)
        if success:
            print('Published alert2 : ' , "Fertilizer PH level(%s) is not safe,use other fertilizer" %PH_sensor,"to
IBM Watson")
            print("")


    #To send alert message to farmer that animal attack on crops.

    if (camera_reading == "Detected"):
            success = deviceCli.publishEvent("Alert3", "json", { 'alert3' : "Animal attack on crops detected" },
qos=0)
        sleep(3)
        if success:
            print('Published alert3 : ' , "Animal attack on crops detected","to IBM Watson","to IBM Watson")
            print("")


     #To send alert message if flame detected on crop land and turn ON the splinkers to take immediate
action.

    if (flame_reading == "Detected"):
        print("sprinkler-2 is ON")
            success = deviceCli.publishEvent("Alert4", "json", { 'alert4' : "Flame is detected crops are in
danger,sprinklers turned ON" }, qos=0)
        sleep(3)
        if success:
            print( 'Published alert4 : ' , "Flame is detected crops are in danger,sprinklers turned ON","to IBM
Watson")
            print("")

    else:
        print("sprinkler-2 is OFF")
        print("")
```

```
   #To send alert message if Moisture level is LOW and to Turn ON Motor-1 for irrigation.

    if (moist_level < 20):
        print("Motor-1 is ON")
            success = deviceCli.publishEvent("Alert5", "json", { 'alert5' : "Moisture level(%s) is low, Irrigation
started" %moist_level }, qos=0)
        sleep(3)
        if success:
                print('Published alert5 : ' , "Moisture level(%s) is low, Irrigation started" %moist_level,"to IBM
Watson" )
            print("")
    else:
       print("Motor-1 is OFF")
       print("")


    #To send alert message if Water level is HIGH and to Turn ON Motor-2 to take water out.

    if (water_level > 20):
        print("Motor-2 is  ON")
         success = deviceCli.publishEvent("Alert6", "json", { 'alert6' : "Water level(%s) is high, so motor is ON
to take water out " %water_level }, qos=0)
        sleep(2)
        if success:
                  print('Published alert6 : ' , "water level(%s) is high, so motor is ON to take water out "
%water_level,"to IBM Watson" )
            print("")
    else:
       print("Motor-2 of OFF")
       print("")


    #command recived by farmer
    deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-40242-1660626623.git

DEMO VIDEO LINK:

https://www.youtube.com/watch?reload=9&app=desktop&v=E_ReEDbANHY