

Smart Farmer - IOT Enabled Smart Farming Application

Submitted by

VARUN PANDIYAN T (923819104056)

GUNASEELAN S (923819104302)

SURYA PRAKASH M (923819104047)

RAJADURAI R (923819104036)

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

MANGAYARKARASI COLLEGE OF ENGINEERING,

MADURAI 625 402



ANNA UNIVERSITY:: CHENNAI 600 025

TABLE OF CONTENTS

| CHAPTER NO | TITLE | PAGE NO |
|------------|--|---------|
| 1 | INTRODUCTION 1.1 Project Overview 1.2 Purpose | 1 |
| 2 | LITERATURE SURVEY 2.1 Existing System 2.2 References 2.3 Problem Statement Definition | 2 |
| 3 | IDEATHON & PROPOSED SOLUTION 3.1 Empathy Map Canvas 3.2 Ideathon & Brainstorming 3.3 Proposed Solution 3.4 Problem Solution Fit | 3 |
| 4 | REQUIREMENT ANALYSIS 4.1 Functional Requirement 4.2 Non-Functional Requirement | 7 |
| 5 | PROJECT DESIGN 5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 Customer Journey Map | 8 |
| 6 | PROJECT PLANNING & SCHEDULING 6.1 Sprint Planning, Schedule & Estimation 6.2 Sprint Delivery Schedule 6.3 Reports From JIRA | 11 |
| 7 | CODING & SOLUTIONING 7.1 Feature | 13 |

| | | |
|----------|---|----|
| 8 | TESTING 8.1 Test cases 8.2 User Acceptance Testing | 23 |
|----------|---|----|

| | | |
|-----------|--|----|
| 9 | RESULTS 9.1 Performance Metrics | 25 |
| 10 | ADVANTAGES & DISADVANTAGES 10.1 Advantages 10.2 Disadvantages | 26 |
| 11 | CONCLUSION | 27 |
| 12 | FUTURE SCOPE | 27 |
| 13 | APPENDIX | 28 |

CHAPTER-1

Introduction

Internet of Things Smart technology enables new digital agriculture. Today technology has become a necessity to meet current challenges and several sectors are using the latest technologies to automate their tasks. Advanced agriculture, based on Internet of Things technologies, is envisioned to enable producers and farmers to reduce waste and improve productivity by optimizing the usage of fertilizers to boost the efficiency of plants. It gives better control to the farmers for their livestock, growing crops, cutting costs, and resources.

1.1 Project Overview

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.

1.2 Purpose

We have tried to focus on different scientific applications which could be put together in the agricultural field for better accuracy with better productivity using less manpower. Moreover, we include a method for monitoring the agricultural fields from any remote location and assessing the basic condition of the field. This is the project from the motivation of the farmers working in the farmlands who are solely dependent on the rains and bore wells for irrigation of their land. In recent times, the farmers have been using irrigation techniques through manual control in which the farmers irrigate the land at regular intervals by turning the water-pump ON/OFF when required and it also contains the open weather for knowing current weather in our current location.

CHAPTER-2

LITERATURE SURVEY

2.1 Existing problem

Agriculture is the foundation of our Nation. In the past, agriculturists used to figure out the ripeness of soil and influenced presumptions to develop which kind of product. They didn't think about the dampness, level of water and especially climate conditions which are more horrifying to an agriculturist. They utilize pesticides in view of a few suspicions which lead a genuine impact to the yield if the supposition isn't right. Profitability relies upon the last phase of the harvest on which agriculturists depend.

2.2 References

1. Official Mobile application for Smart Agriculture
CropX : <https://play.google.com/store/apps/details?id=com.cropx.adaptive&hl=en>
2. Official Mobile application for Smart Farming
Gramophone : <https://play.google.com/store/apps/details?id=agstack.gramophone&hl=en>
3. S. R. Prathibha, A. Hongal, and M. P. Jyothi, "IOT BASED MONITORING SYSTEM IN SMART AGRICULTURE," pp. 5–8, 2017.
4. Tien Cao Huang and Can Nguyen Duy, "Environment Monitoring for Agricultural Application Based on Wireless Sensor Network.," April 16-17.

2.3 Problem Statement Definition

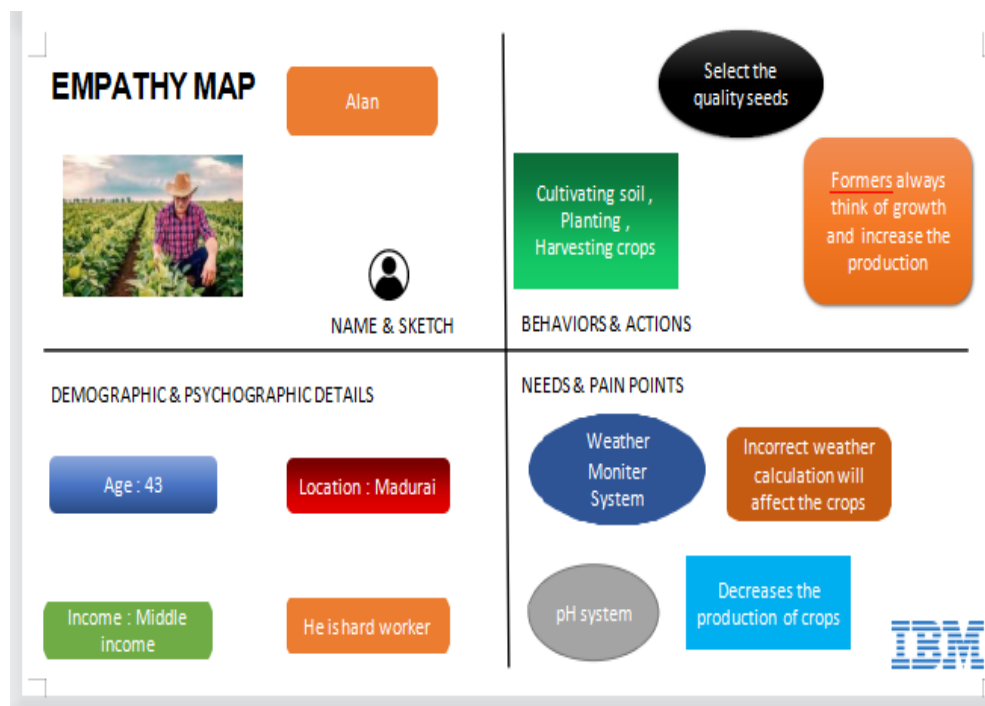
The researcher is supposed to implement an "IoT Based Smart Farming System" with various sensors, which will help to collect the data and analyse it. The proposed system collects information about different agricultural parameters (temperature, humidity, moisture) using an IoT sensor. It shows Open weather values also. Farmers can view all the parameters required for a smart farming system through the webpage and Mobile Application.

CHAPTER-3

IDEATION AND PROPOSED SOLUTION

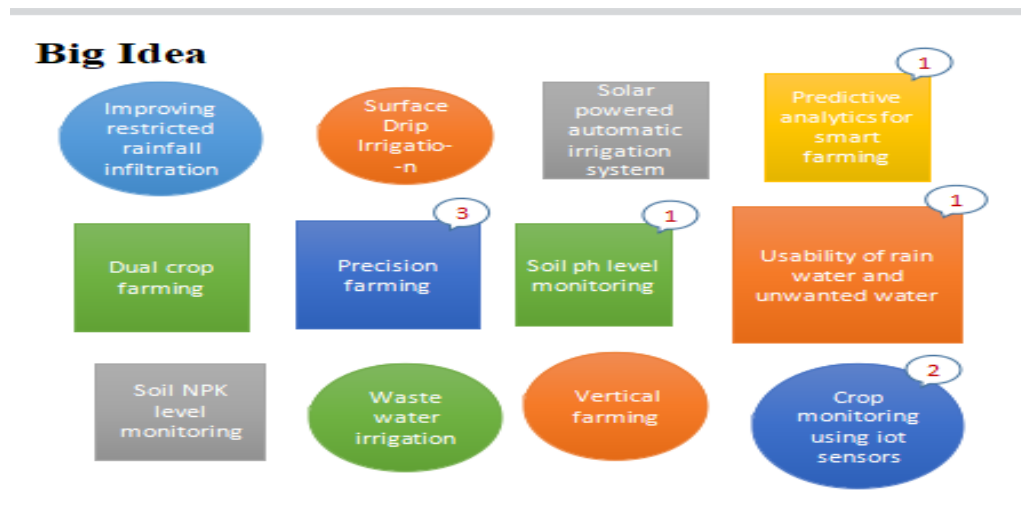
3.1 Empathy Map Canvas

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.



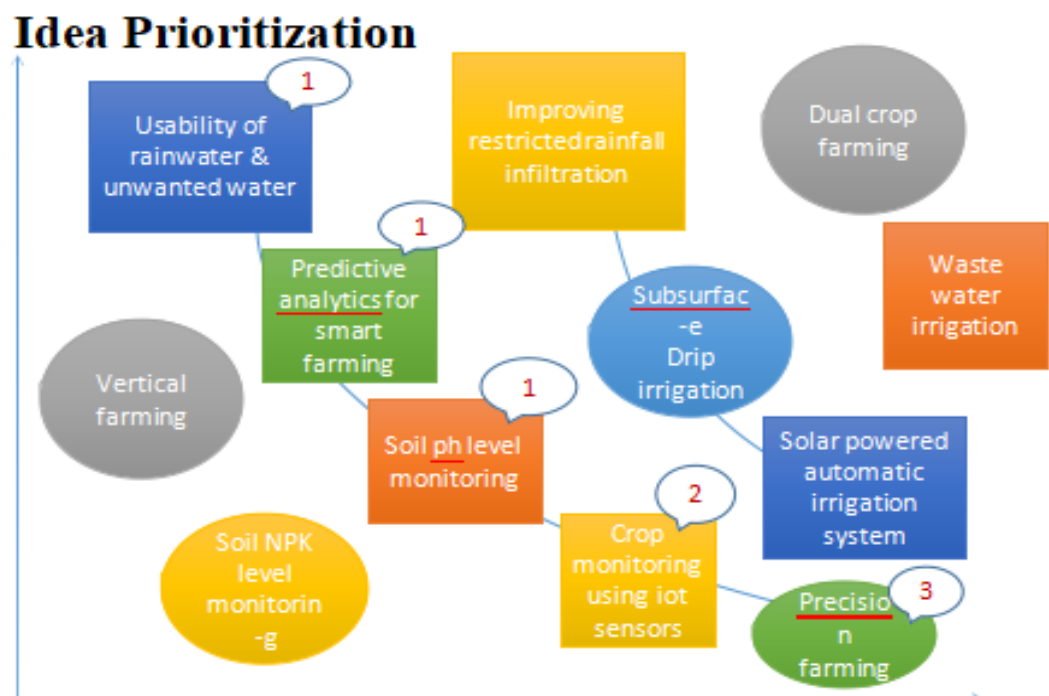
3.2.1 Big Ideas

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



3.2.2 Idea Prioritization

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

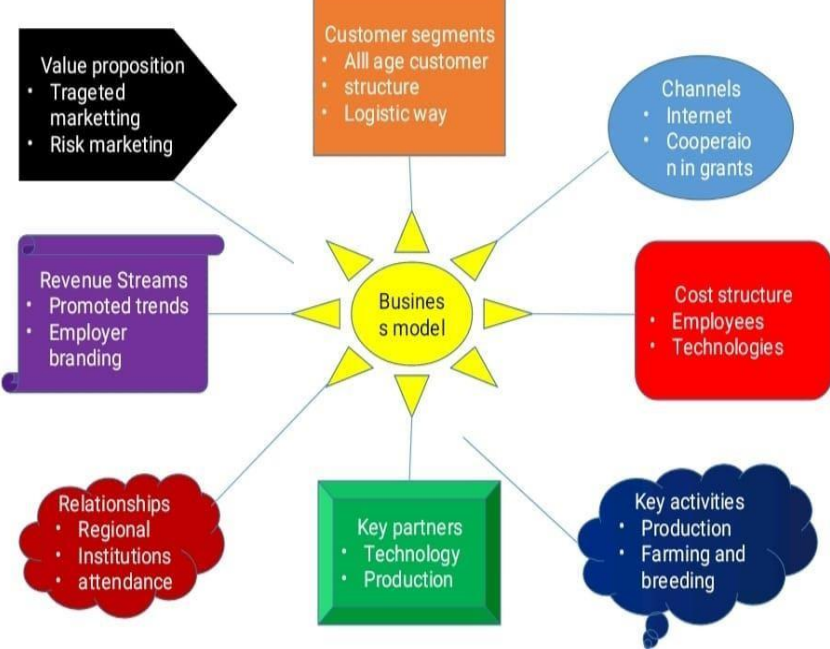


3.3 Problem Solution Fit

| Problem-Solution fit canvas 2.0 | | Purpose / Vision | |
|---|---|--|---|
| Define C.S. fit into | 1. CUSTOMER SEGMENT(S) CS Farmers can be sub-segmented under three categories micro, small, or marginal; emerging and large; or commercial farmers either based on farm surplus, gross revenue, or land under cultivation. | 5. CUSTOMER CONSTRAINTS CC IOT based Smart farming helps farmers to better understand the important factors such as water, topography, aspect, vegetation and soil types. This allows farmers to determine the best uses of scarce resources within their production environment and manage these in an environmentally and economically sustainable manner. | 3. AVAILABLE SOLUTIONS AS Spraying chemical after plant got disease alternatively we prevent plant prior from disease we used to find NPK level and Monitoring water Quality. Pros: Avoid abnormal growth of plants & improve production. Cons: We can't set general PH and Nutrient value for all the plants. Each plants have their own level. |
| | 2. PROBLEMS J&P Climate change can disrupt food availability, reduce access to food, and affect food quality. For example, projected increases in temperatures, changes in precipitation patterns, changes in extreme weather events, and reductions in water availability may all result in reduced agricultural productivity. As agriculture struggles to support the rapidly growing global population, plant disease reduces the production and quality of food, fiber and biofuel crops. Plant diseases can affect plants by interfering with several processes such as the absorbance and translocation of water and nutrients, photosynthesis, plant growth and development etc., | 6. PROBLEM ROOT CAUSE RC <ul style="list-style-type: none">• Lack of management commitment• Lack of or incorrect training• Lack of or incorrect documentation | 7. BEHAVIOUR BE Farmers always concern about to increase the growth of plants & production and essential disease control implementations in their fields. |
| Identify strong TR & EM | 3. TRIGGERS TR Weather Changes -This is in part due to mass deforestation because of urbanization. This affects every step of the farming process. Lack of information -Most farmers are illiterate and can't access farming relevant information that can help with the farming process. This is mostly because of the high illiteracy levels in the country. | 10. YOUR SOLUTION SL This system is ArduinoMicrocontroller-IOT based smart farming application and this is help to increase Production using a microcontroller from PIC family. To increasing the production and prevent plants from unnecessary disease, we use Humidity & Temperature, PH level sensor etc . Notify the collected data to Mobile device with the help of Arduino & Sensors. | 8. CHANNELS of BEHAVIOUR CH 1. ONLINE We notify the information about the agriculture on mobile application. 2. OFFLINE Users are in offline they are only know about the previous information about the field. |
| | 4. EMOTIONS : BEFORE / AFTER EM In Agriculture field farmers are always concern about monitoring the moisture level, weather & monitoring plant disease manually this cause increase in labour cost. To solve the problem proposed the system, it is help to increase Production using microcontroller from PIC family. To increasing the production and prevent plants from unnecessary disease, we use Humidity & Temperature, PH level sensor etc.. Notify the collected data to Mobile device with the help of Arduino & Sensors. | | |
| <div>Problem Solution fit canvas is licensed under a Creative Commons Attribution Non-Commercial-NoDerivatives 4.0 International License. Created by Dr. H. N</div> | | | |

3.4 Proposed Solution

| S.No. | Parameter | Description |
|-------|--|--|
| 1. | Problem Statement (Problem to be solved) | Smart Farmer – IoT Enabled Smart Farming Application |
| 2. | Idea / Solution description | This system is Arduino Microcontroller-IOT based smart farming application and this is help to increase Production using a microcontroller from PIC family. To increasing the production and prevent plants from unnecessary disease, we use Humidity & Temperature, PH level sensor etc .. Notify the collected data to Mobile device with the help of Arduino & Sensors. |
| 3. | Novelty / Uniqueness | Prevent plant prior from disease and abnormal growth we need to find NPK level and Monitoring water Quality. |
| 4. | Social Impact / Customer Satisfaction | In order to meet the needs of consumers and increase profit value, farms need to demonstrate that products offered to the market are clean products also, it helps to track and trace agrifood supply chains production process and trace the origin of agricultural products. This solution has successfully supported the tracing of food and agricultural products through QR codes, improving product quality and ensuring the clear traceability of products, thereby allowing consumers to know the product's entire history. ⁵ |

| | | |
|----|-----------------------------------|---|
| 5. | Business Model (Revenue Model) |  <p>The diagram illustrates a Business Model Canvas centered around a yellow sun labeled "Business model". Eight surrounding elements are connected to the center:</p> <ul style="list-style-type: none"> Value proposition (black arrow): Targeted marketing, Risk marketing Customer segments (orange box): All age customer structure, Logistic way Channels (blue oval): Internet, Cooperation in grants Cost structure (red box): Employees, Technologies Key activities (dark blue cloud): Production, Farming and breeding Key partners (green box): Technology, Production Relationships (red cloud): Regional, Institutions, attendance Revenue Streams (purple box): Promoted trends, Employer branding |
| 6. | Scalability of the Solution | <p>Using IoT sensors, prevent the plants from abnormal growth and unnecessary disease & Monitoring the air humidity and temperature to Improve quality and production of the crop, its beneficial for both Consumers and Farmers.</p> |

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 Functional Requirements

| FR No. | Functional Requirement(Epic) | Sub Requirement(Story/Sub-Task) |
|--------|------------------------------|--|
| FR-1 | User registration | Registration using username and password Follow the instructions |
| FR-2 | User Confirmation | Confirmation via Username Confirmation via Password |
| FR-3 | Interface sensor | Interface sensors and the applications are used to monitor the crops activities like soil moisture, temperature & humidity and send information to farmers |
| FR-4 | Accessing data sets | Data sets are retrieved from Cloudant DB |
| FR-5 | Mobile application | Motors and sprinklers in the field can be controlled by mobile application. |

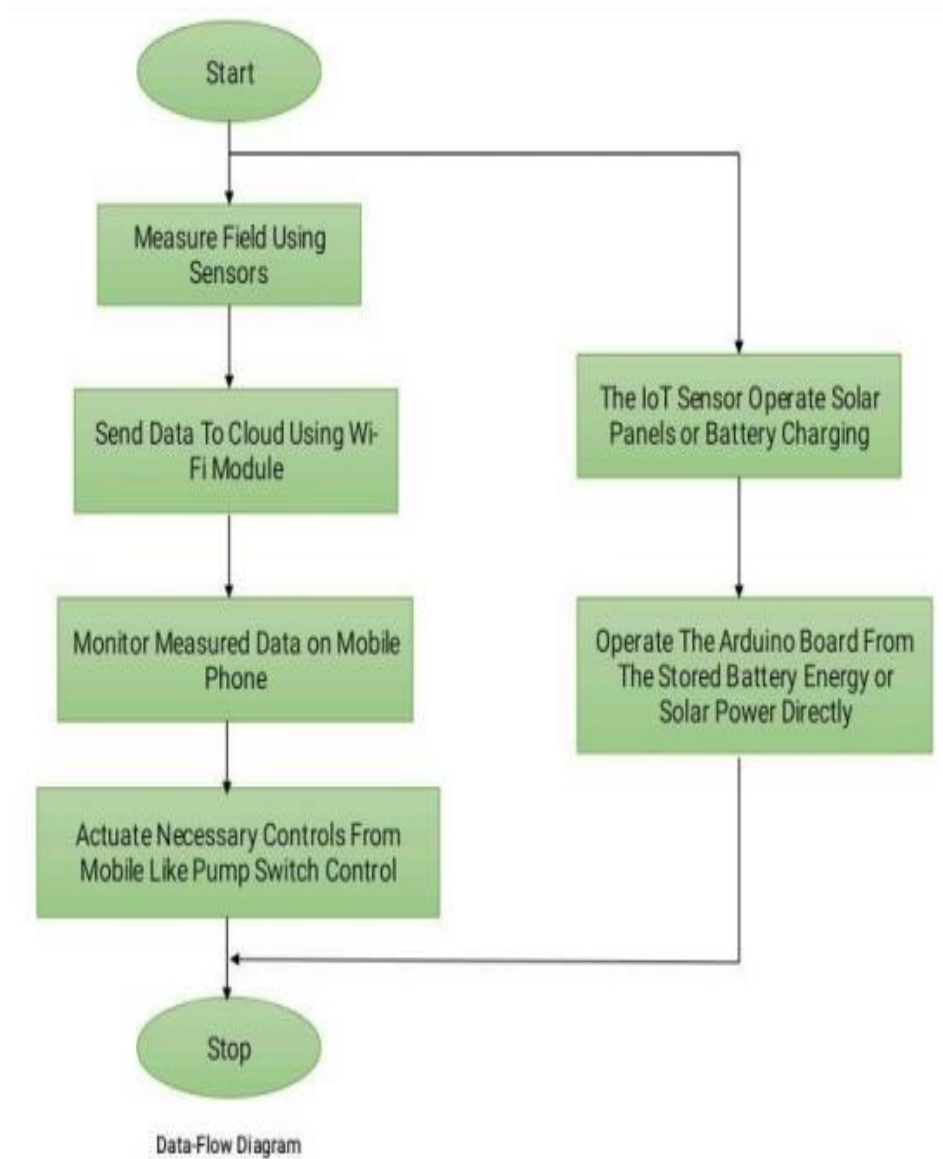
4.2 Non-Functional Requirements

| FRNo. | Non-FunctionalRequirement | Description |
|-------|---------------------------|---|
| NFR-1 | Usability | Use of fertilizers, Irrigation, and scheduled planting operation |
| NFR-2 | Security | Crops could be protected from these diseases use pesticides and biocontrol agents. |
| NFR-3 | Reliability | This project will help farmers with high production of crops and prevent the crops from abnormal growth. This will also help them in achieving better crop yields thus leading to their economic well being |
| NFR-4 | Performance | Agricultural productivity depends on the quality of the seeds with which farmers show their fields. |
| NFR-5 | Availability | Farming methods require growers' appropriate plant protection strategy and training |
| 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 |

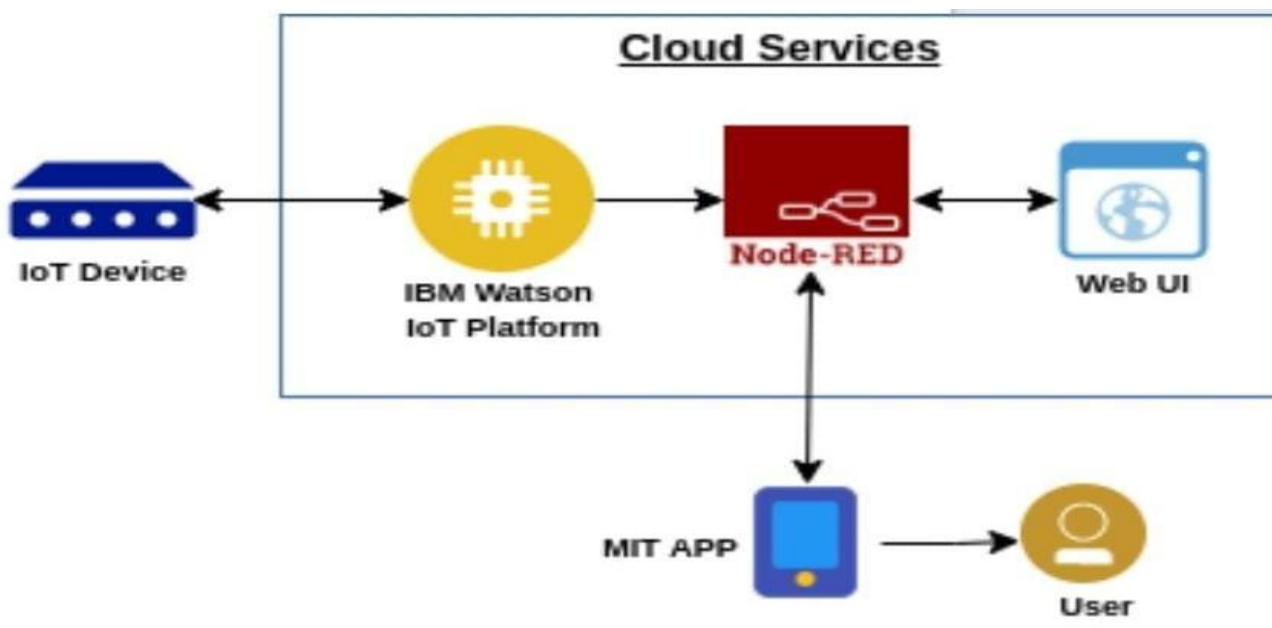
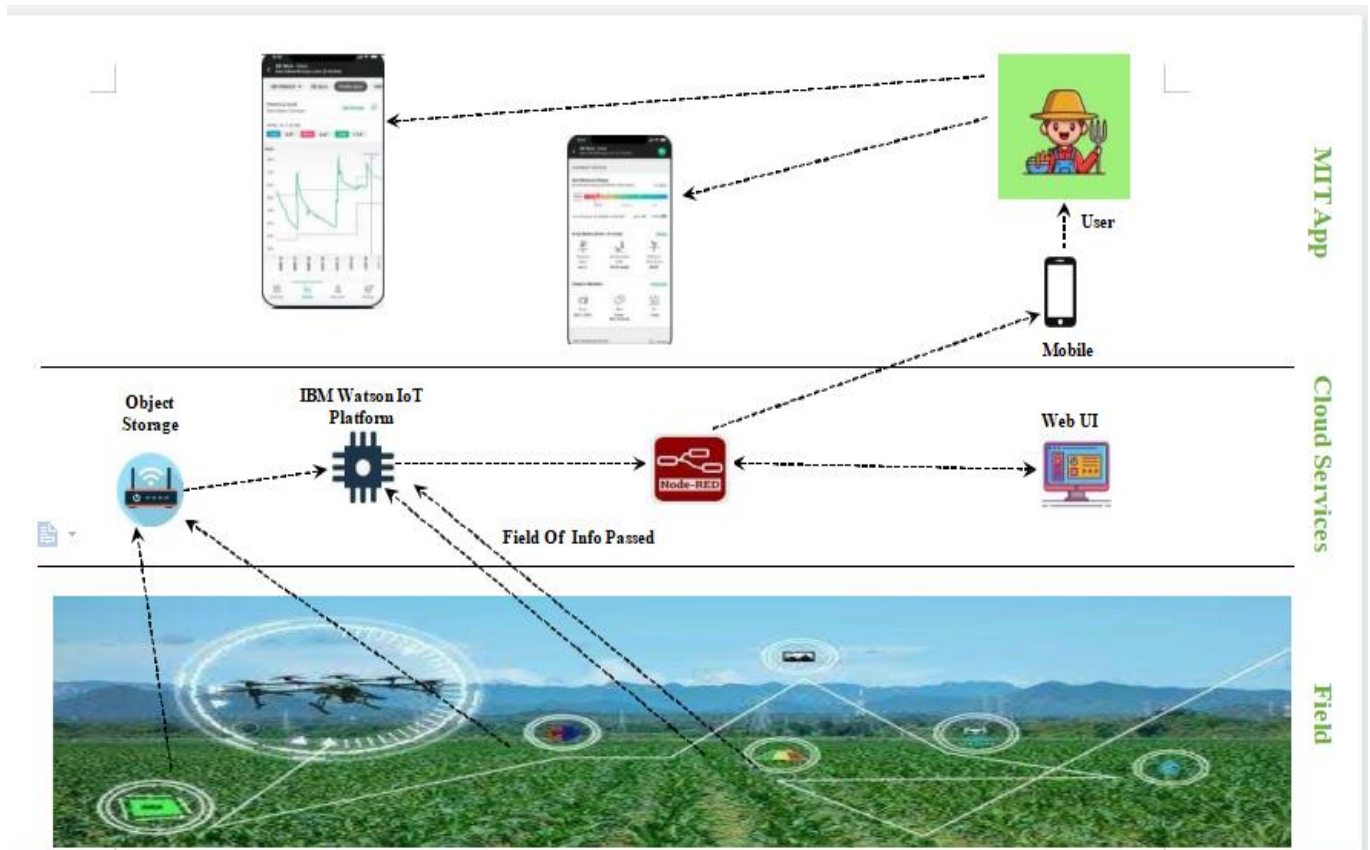
CHAPTER-5

PROJECT DESIGN

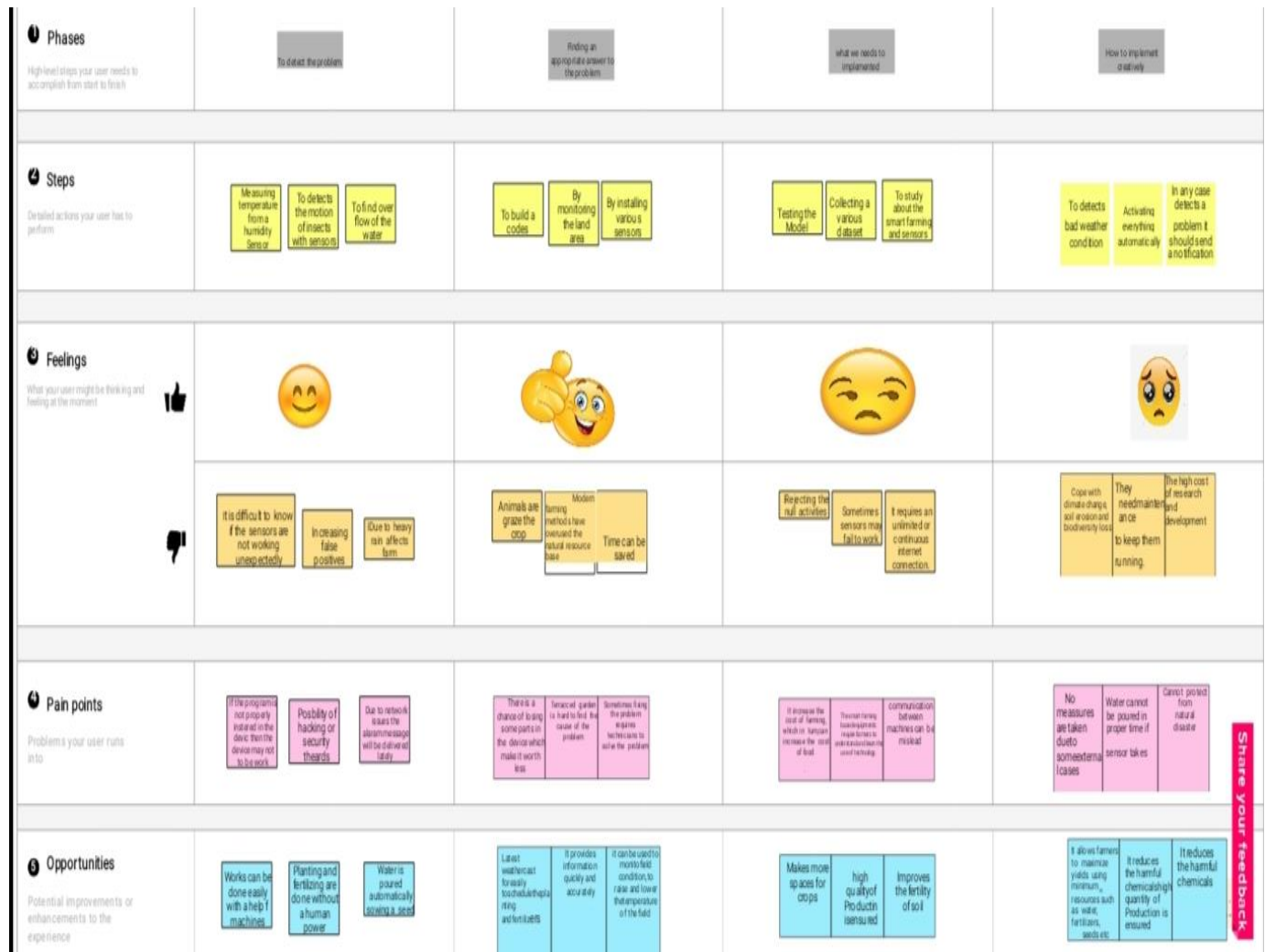
5.1 Data Flow Diagram



5.2 Solution & Technical Architecture



5.3 Customer Journey Map



CHAPTER-6

PROJECT PLANNING PHASE

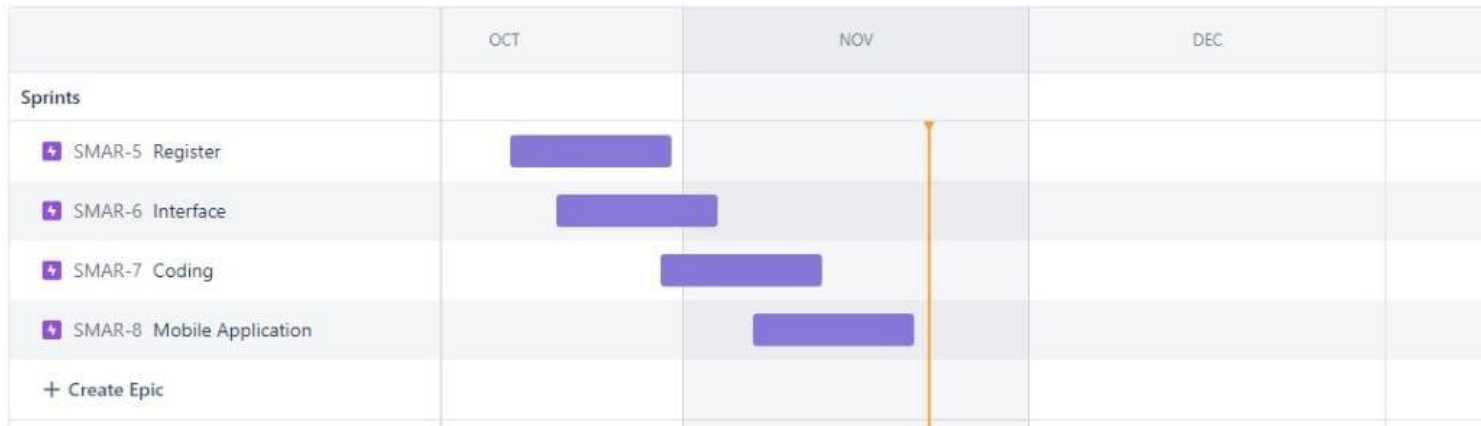
6.1 Sprint Planning, Schedule & Estimation

| 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 Username and password. | 2 | High | S Gunaseelan |
| Sprint-1 | User Confirmation | USN-2 | As a farmer, I will enter the Username and password. I have registered for the application. | 1 | Medium | M Surya Prakash |
| Sprint-1 | Login | USN-3 | As a farmer, I can log into the application by entering Username and password. | 2 | High | T Varun Pandiyan |
| 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 | R Rajadurai S Gunaseelan |
| 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 | T Varun Pandiyan S Gunaseelan M Surya Prakash R Rajadurai |
| Sprint-4 | Mobile Application | USN-1 | As a Farmer, I will show the current Information of the Field. | 1 | Medium | T Varun Pandiyan M Surya Prakash |

6.2 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 |

6.3 Reports From JIRA



CHAPTER-7

CODING & SOLUTION

7.1 Feature

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "puubdh"
deviceType = "raspberrypi"
deviceId = "demo123"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    else :
        print ("motor 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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting"
10 times
deviceCli.connect()

while True:
    #Get Sensor Data from DHT11

    temp=random.randint(0,100)
    hum=random.randint(0,100)
    moist=random.randint(0,100)

    data = { 'temp' : temp, 'hum': hum, 'moist' : moist }#print
    data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %" % hum, "Soil Moisture
= %s " % moist, "to IBM Watson")

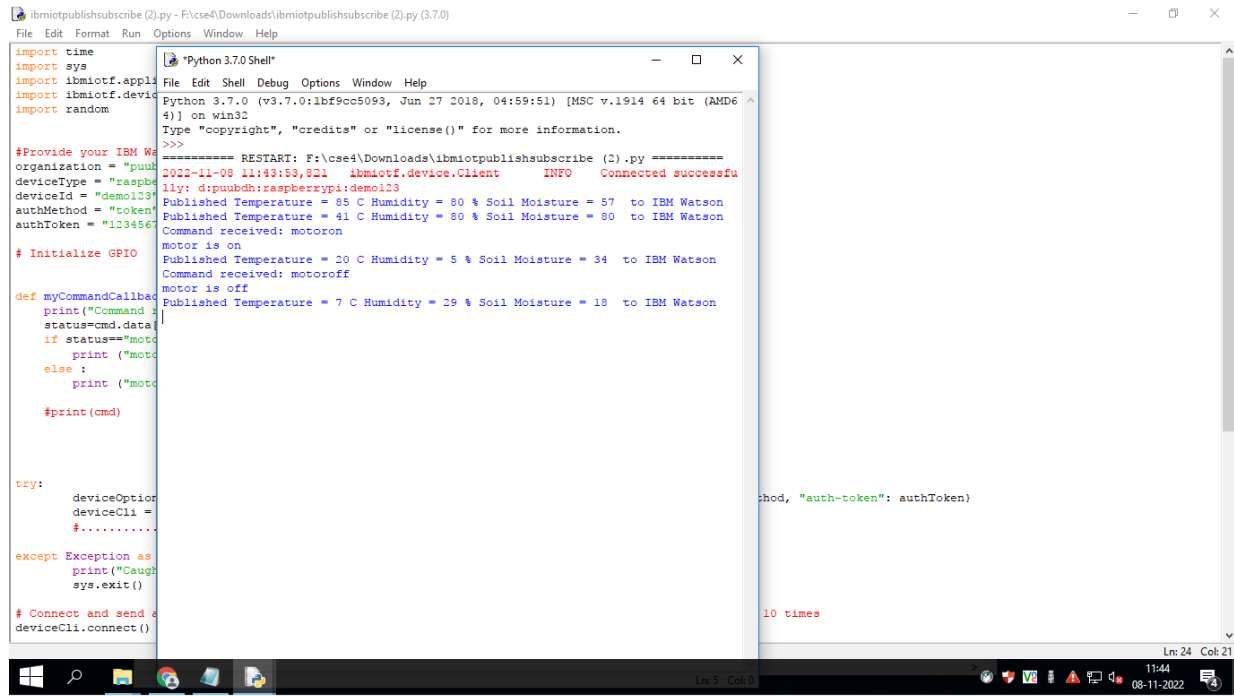
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(10)

    deviceCli.commandCallback = myCommandCallback

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

```

Output:



```
ibmiotpublishsubscribe (2).py - F:\cse4\Downloads\ibmiotpublishsubscribe (2).py (3.7.0)
File Edit Format Run Options Window Help

import time
import sys
import ibmiotf.appl
import ibmiotf.devi
import random

#Provide your IBM W
organization = "puub
deviceType = "raspb
deviceId = "demo123
authMethod = "token
authToken = "123456

# Initialize GPIO

def myCommandCallba
    print("Command
    status=cmd.data
    if status=="motc
        print ("motc
    else :
        print ("motc
    #print(cmd)

try:
    deviceOption
    deviceCli =
    #.....

except Exception as
    print("Caught
    sys.exit()

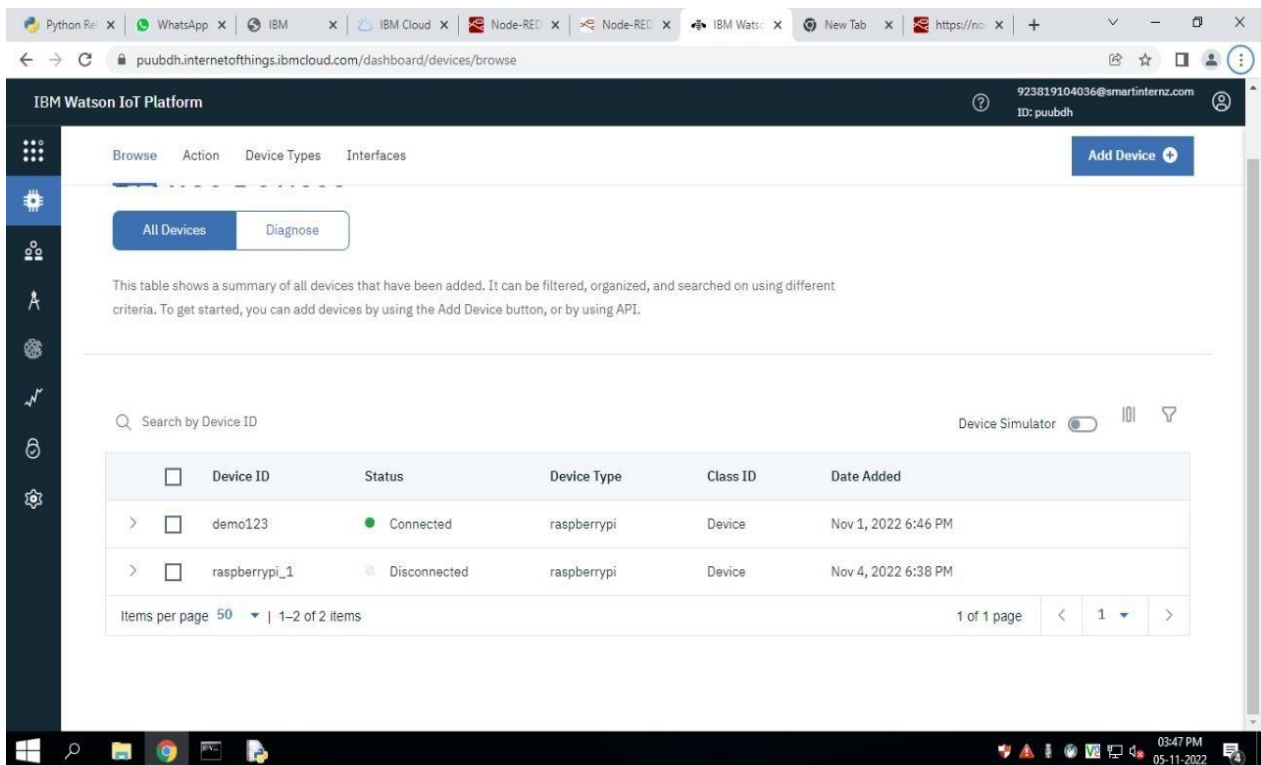
# Connect and send e
deviceCli.connect()

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (tags/v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: F:\cse4\Downloads\ibmiotpublishsubscribe (2).py =====
2022-11-08 11:43:53,821 ibmiotf.device.Client INFO Connected successfu
lly: d:puubdh:raspberrypi:demo123
Published Temperature = 85 C Humidity = 80 % Soil Moisture = 57 to IBM Watson
Published Temperature = 41 C Humidity = 80 % Soil Moisture = 80 to IBM Watson
Command received: motoron
motor is on
Published Temperature = 20 C Humidity = 5 % Soil Moisture = 34 to IBM Watson
Command received: motoroff
motor is off
Published Temperature = 7 C Humidity = 29 % Soil Moisture = 18 to IBM Watson

shod, "auth-token": authToken)

10 times
```

Device Details:



IBM Watson IoT Platform

923819104036@smartintenz.com
ID: puubdh

Browse Action Device Types Interfaces

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

| | Device ID | Status | Device Type | Class ID | Date Added |
|---|---------------|--------------|-------------|----------|---------------------|
| > | demo123 | Connected | raspberrypi | Device | Nov 1, 2022 6:46 PM |
| > | raspberrypi_1 | Disconnected | raspberrypi | Device | Nov 4, 2022 6:38 PM |

Items per page 50 | 1-2 of 2 items

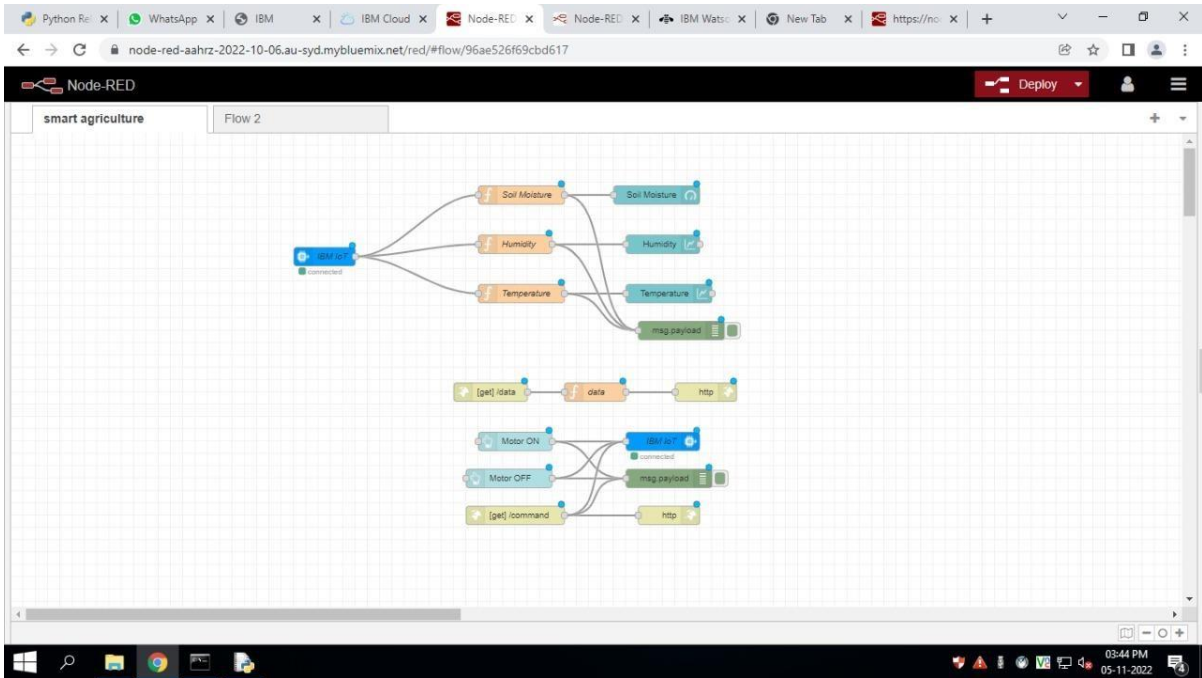
1 of 1 page

Recent Events:

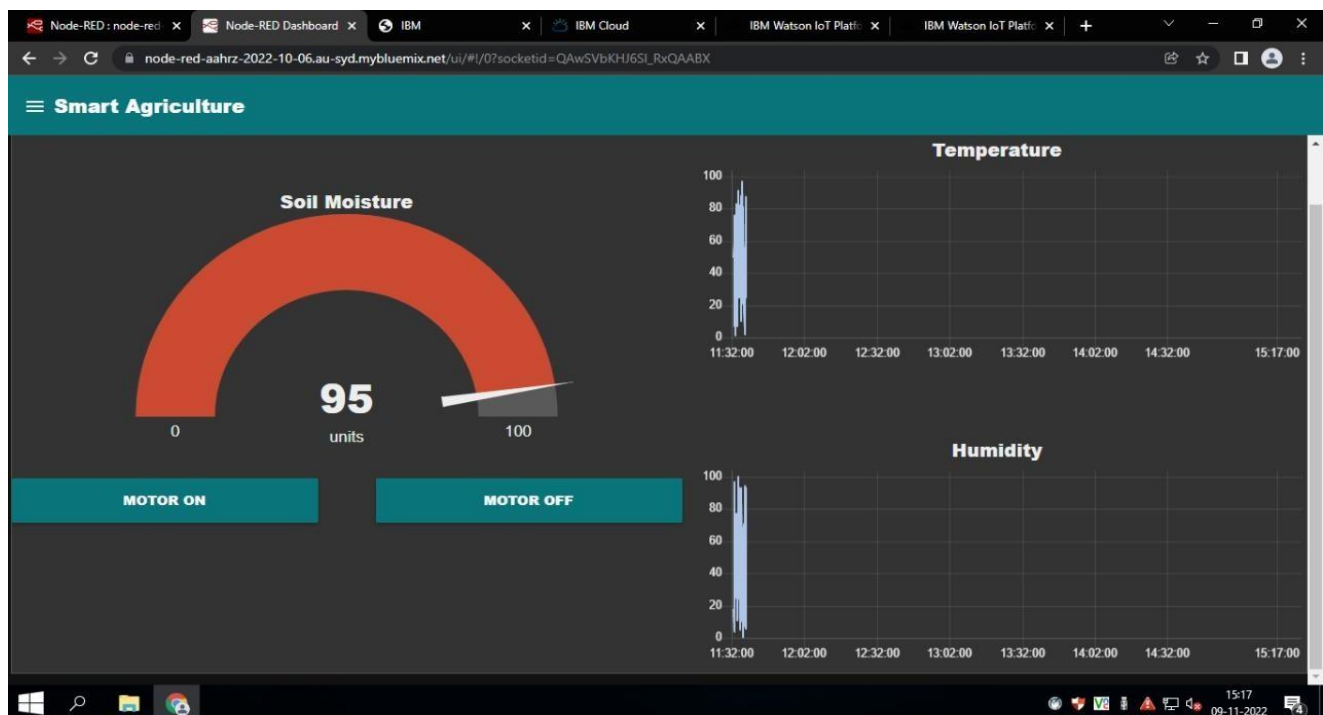
The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various functions. The main content area displays details for a device named 'demo123', which is 'Disconnected' and of type 'raspberrypi'. The 'Recent Events' tab is selected, showing a table of live data streams.

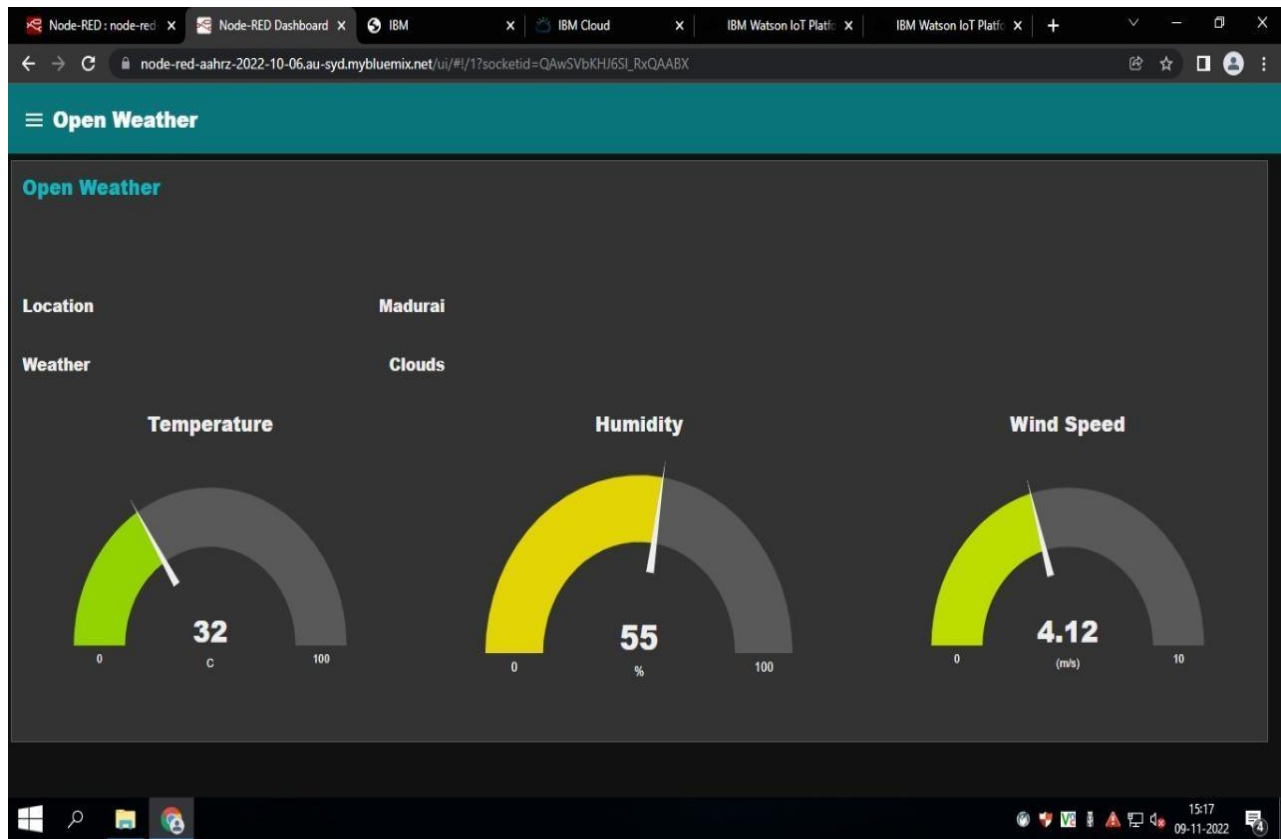
| Event | Value | Format | Last Received |
|-----------|---------------------------------|--------|-------------------|
| IoTSensor | {"temp":15,"hum":78,"moist":66} | json | a few seconds ago |
| IoTSensor | {"temp":41,"hum":90,"moist":15} | json | a few seconds ago |
| IoTSensor | {"temp":54,"hum":29,"moist":61} | json | a few seconds ago |
| IoTSensor | {"temp":46,"hum":38,"moist":77} | json | a few seconds ago |
| IoTSensor | {"temp":21,"hum":8,"moist":27} | json | a few seconds ago |

Node-Red Connection and Dashboard Design:



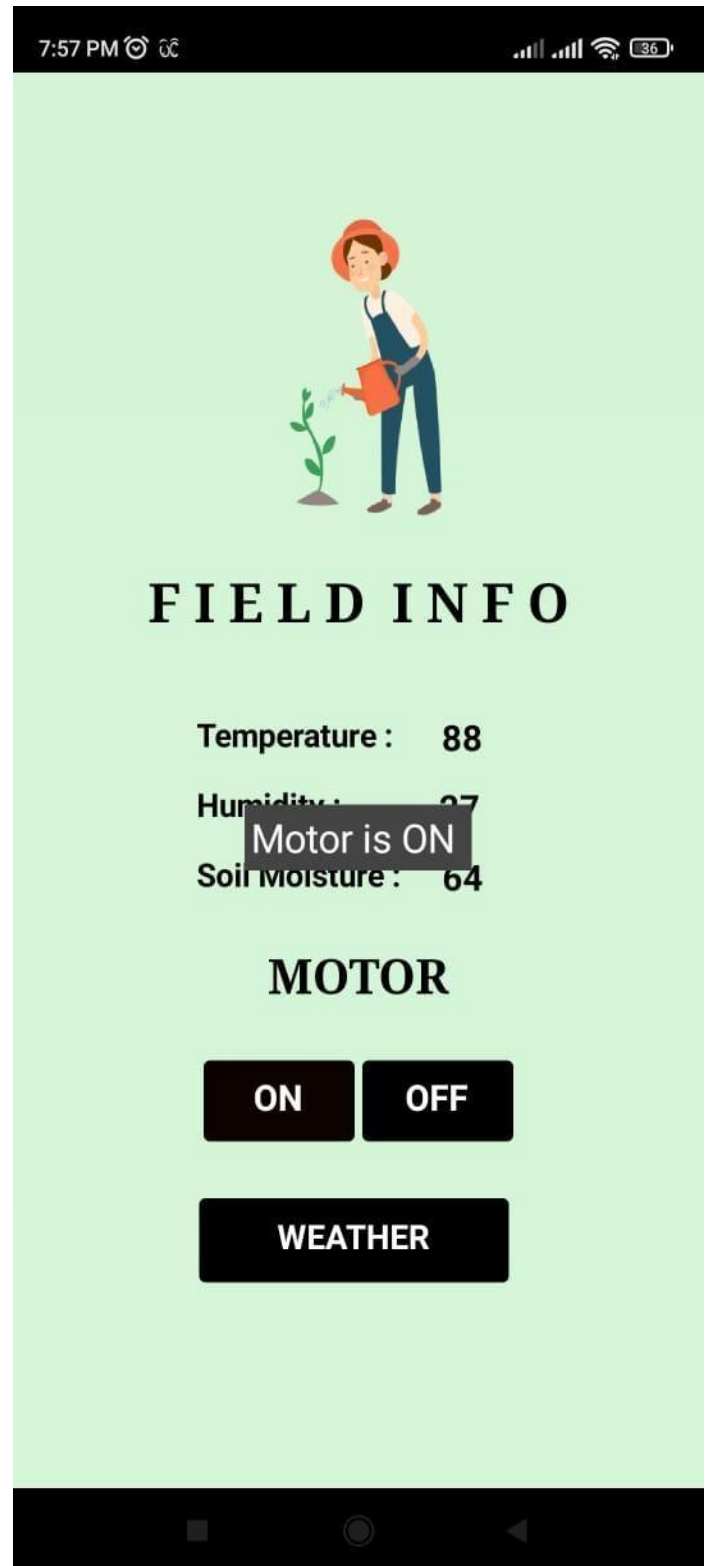
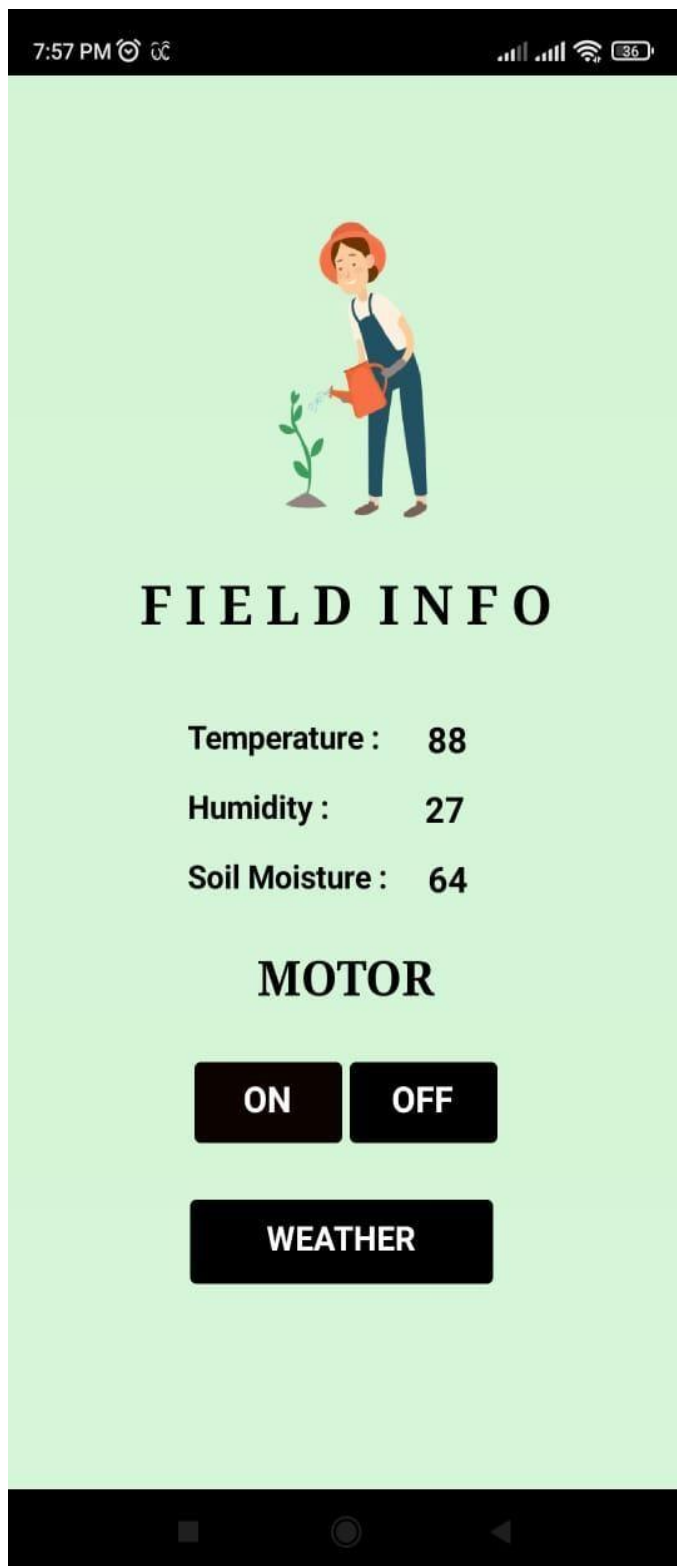
The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow titled "Open Weather". The flow starts with an "inject" node, followed by a "timestamp" node, then a "Madura" node. This is followed by a series of "function" nodes that output to "msg.payload" for "Location", "Weather", "Temperature", "Humidity", and "Wind Speed". A "get /details" node is also present, followed by a "function" node and an "http" node. The left sidebar shows the "common" and "function" node palettes. The right sidebar shows the "debug" console with a list of messages, including "Madura1" and "Haze". The bottom status bar shows the system time as 11:46 on 08-11-2022.

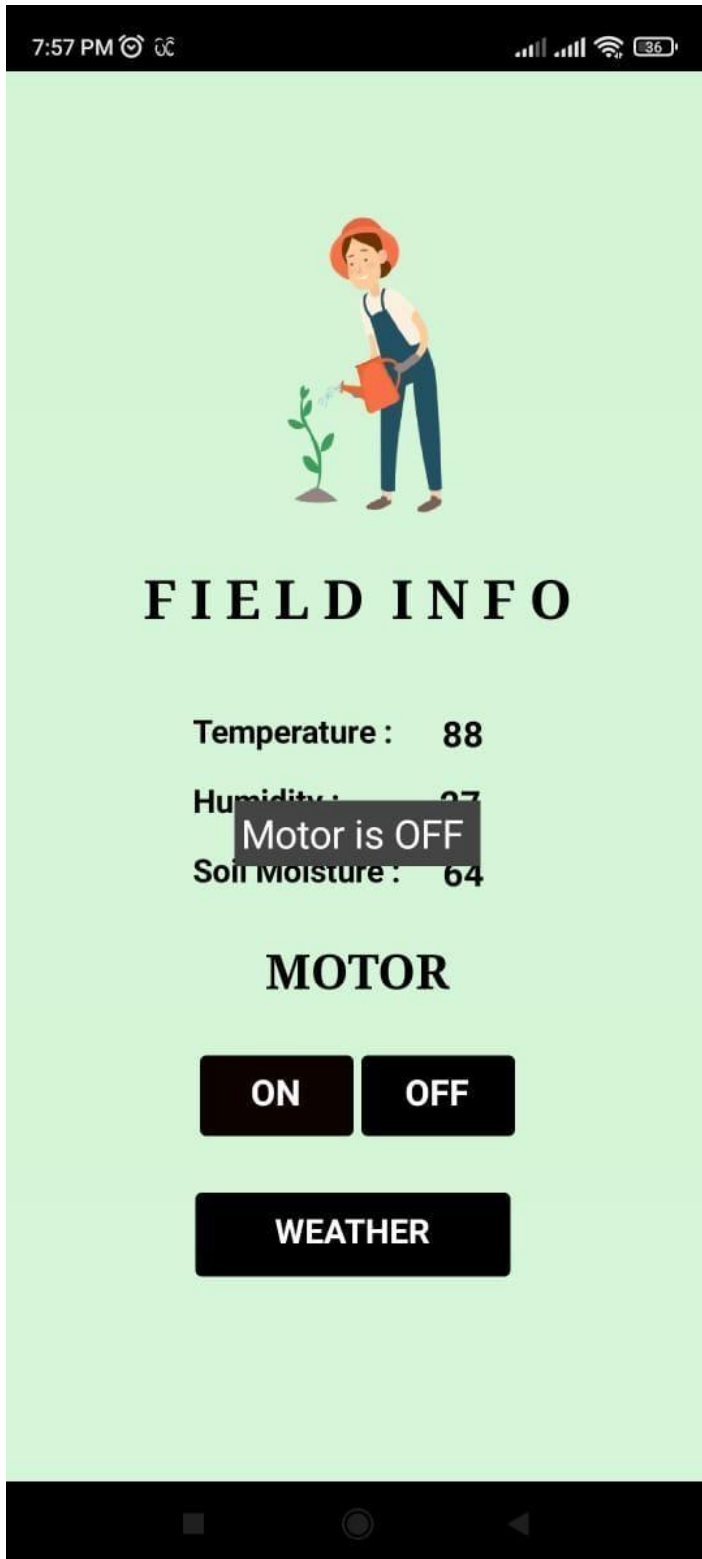




Mobile Application:







CHAPTER – 8

TESTING

8.1 Test cases

| Test Case | Test Scenario | Test Data | Status | Comments | Executed by |
|-----------|---|---|--------|------------------|-----------------|
| TC_001 | Create the IBM Cloud services which are being used in this project | https://cloud.ibm.com/login | Pass | Results verified | Rajadurai R |
| TC_002 | Configure the IBM Cloud services which are being used in completing this project. | https://cloud.ibm.com/login | Pass | Results verified | Gunaseelan S |
| TC_003 | IBM Watson IoT platform acts as the mediator to connect the web application to IoT devices, so create the IBM Watson IoT platform | https://puubdh.internetofthings.ibmcloud.com/dashboard/devices/browse | Pass | Results verified | VarunPandiyan T |
| TC_004 | 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 | VarunPandiyan T |
| TC_005 | 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=59c9d5bd-4d31-3611-897a-f94eea80dc9f&default | Pass | Results verified | Suryaprakash M |
| TC_006 | Create a Node-RED service. publish random sensor data such as temperature, humidity level, soil moisture to the IBM IoT platform | Values of sensors and button for Motor ON/OFF is displayed | Pass | Results verified | Suryaprakash M |
| TC_007 | | https://www.python.org/downloads/release/python-370/ | Pass | Results verified | Gunaseelan S |

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 [SmartFarmer IoT Enabled Smart Farming Application] 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
 - a. Functional Requirements
 - b.Non Functional Requirements
2. Implementation result
 - a.System Implementation results
 - b.Results of web application
 - c.Result of mobile application
3. Resource utilization results
 - a. Foreground activities results
 - b. Memory usage
 - c. Energy usage
4. Background activities result

CHAPTER-10

ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGE:

- ❖ Communicating the device at a larger distance through web application. It will play an important role in reducing the manpower and traveling expenses of a farmer.
- ❖ Monitoring parameters like temperature, humidity etc. will play an important role in improving the growth of the plant.
- ❖ Integrating the weather station to the web browser will provide the details of status of the cloud, wind speed etc. It will allow the farmer to protect their plants from natural calamities.

10.2 DISADVANTAGE:

- ❖ Since the real time sensor will be connected to the controller, the controller requires continuous supply of the internet to transfer the data.
- ❖ Non availability of weather prediction for a long period of time. Since the long weather prediction requires additional payment to open weather.

CHAPTER-11

CONCLUSION

IoT based SMART AGRICULTURE SYSTEM for Live Monitoring of Temperature and Soil Moisture and to control motor and light remotely have been proposed using Node Red and IBM Cloud Platform. The System has high efficiency and accuracy in fetching the live data of temperature and soil moisture. The IoT based smartfarming System being proposed via this project will assist farmers in increasing the agriculture yield and take efficient care of food production as the System will always provide a helping hand to farmers for getting accurate live feed of environmental temperature and soil moisture with more than 99% accurate results. Therefore, the project proposes a thought of consolidating the most recent innovation into the agrarian field to turn the customary techniques for water systems to current strategies in this way making simple profitable and temperate trimming.

CHAPTER-12

FUTURE SCOPE

The project has vast scope in developing the system and making it more user friendly and the additional features of the system like By installing a webcam in the system, photos of the crops can be captured and the data can be sent to a database. Speech based options can be implemented in the system for the people who are less literate. GPS (Global Positioning System) can be integrated to provide specific location of the farmer and more accurate weather reports of agriculture fields and gardens. Regional language features can be implemented to make it easy for the farmers who are aware of only their regional language..

CHAPTER-13

APPENDIX

Github : <http://bitly.ws/wYdM>

Demo Link : <http://bitly.ws/wYb5>