PROJECT REPORT

**Department Of Electronics And Communication Engineering**

# 

# IBM-NALAIYA THIRAN

# 

# **PROJECT NAME:**

# SMART FORMER-IOT ENABLED SMART

# 

# FARMING APPLICATION

# **TEAM ID:**

# 

# PNT2022TMID38164

# 

# **TEAM MEMBERS:**

# 

# BAVADHARANI.K - 411819106002

# 

# SATHISH.M - 411819106005

# 

# ARUN KUMAR.A - 411819106001

# 

# ESAKKIRAJAN.M - 411819106305

# INTRODUCTION

* 1. Project Overview
  2. Purpose

# LITERATURE SURVEY

* 1. Existing problem
  2. References
  3. Problem Statement Definition

# IDEATION & PROPOSED SOLUTION

* 1. Empathy Map Canvas
  2. Ideation & Brainstorming
  3. Proposed Solution
  4. Problem Solution fit

# REQUIREMENT ANALYSIS

* 1. Functional requirement
  2. Non-Functional requirements

# PROJECT DESIGN

* 1. Data Flow Diagrams
  2. Solution & Technical Architecture
  3. User Stories

# PROJECT PLANNING & SCHEDULING

* 1. Sprint Planning & Estimation
  2. Sprint Delivery Schedule
  3. Reports from JIRA

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

* 1. Feature 1
  2. Feature 2
  3. Database Schema (if Applicable)

# TESTING

* 1. Test Cases
  2. User Acceptance Testing

# RESULTS

* 1. Performance Metrics

# ADVANTAGES & DISADVANTAGES

1. **CONCLUSION**

# FUTURE SCOPE

1. **APPENDIX**

Source Code

GitHub & Project Demo Link

# INTRODUCTION

* 1. **Project Overview**

Agriculture plays a important role in country’s economy and provides a large scale employment to the people. However, agriculture is highly dependent upon weather and climate. For example, changes in temperature, soil moisture, carbon dioxide may result in low yield of crops. It is Significant to monitor environmental parameters in order to manage crop growth and increase the agricultural production yield. The sensed information is not only important for decision making but also for evaluating impacts of agricultural practices on environment. Nowadays, it is more necessary than ever to increase the crop yields food grain production. Cloud connected, wireless system aid in this crop yield maximization, which automates day- to-day agricultural tasks and real time monitoring for smart decision-making.

# Purpose

* + - Need for technology to monitor important parameters like soil moisture, temperature, Humidity etc. to improve the cultivation process.
    - Need for technology to monitor weather of particular area with reliable

source to save the crops at the time of natural calamities like flood, cyclone etc.

* + - Development of certain techniques to reduce the workforce, energy and time for cultivation.
    - Development of a feasible method to control the electrical equipment in the farm from any part of the world.

# LITERATURE SURVEY

* 1. **EXISTING PROBLEM**

1. Controlling the device from longer distance from web application.
2. Getting the weather data from weather station.
3. Transfer of node data to the gateway at faster rate.
4. Unavailability of data’s such as PH level, potassium, Nitrogen etc related to the soil.

# PROPOSED SOLUTION

1. To control a device from longer distance from web application.
2. To get the weather details like wind speed, temperature, humidity from weather station through weather API.
3. To display the data in the web application.

# References

1. **IoT Enabled Smart Farming and Irrigation System:**

**Authors:** M. Rohith, R Sainivedhana, Dr. N. Sabiyath Fatima

**Published:** IEEE 2021

**Description:** In this paper, authors have demonstrated a IoT enabled smart farming and irrigation system to automate the process of watering to plants. This system helps to measure the values of various parameters such as humidity, moisture and temperature of plants and water them accordingly. This system consists of three sensors which will sense the values of humidity, moisture and temperature of plants. If any of the sensor values decreases the motor automatically turns on the water for plants. The ultimate significance of the paper is that most of the manual work is reduced and watering process is automated with the help of IoT enabled devices as a result of which healthy plants can be grown.

# A Systematic Review of IoT Solutions for Smart Farming:

**Authors:** Emerson Navarro, Nuno Costa, and António Pereira

**Published:** MDPI 2020

**Description:** In this work, authors have presented a systematic review of the state-of-the-art of IoT adoption in smart agriculture and identified the main components and applicability of IoT solutions. In this particular work it was observed that the use of artificial intelligence and image processing techniques has become more common to improve the management of smart farming. From the identified applications of IoT for smart farming it was observed that the most common application is the monitoring of crops. Here, authors showed that different network protocols may be simultaneously used in IoT solutions for smart farming.

# A Multi-collective, IoT-enabled, Adaptive Smart Farming Architecture:

**Authors:** G. Kakamoukas, P. Sariciannidis, G. Livanos, M. Zervakis, D. Ramnalis, V. Polychrnos, T. Karamitsou, A. Folinas, N. Tsitsiokas

**Published:** IEEE 2019

**Description:** In this paper, authors have proposed a precision architecture for Smart Farming in order to use precise and efficient approaches for monitoring and processing information from farms,

crops, forestry, and livestock aiming at more productive and sustainable rural development. This proposed architecture encloses wireless sensor networks, meteorological stations and unmanned aerial vehicles along with an information processing system that leverages machine learning and computing technologies. The innovation of the proposed architecture lies in the creation of an integrated monitoring and decision support system for efficient allocation of resources and protection of plant capital from the diseases.

# Internet of Things and LoRaWAN – Enabled Future Smart Farming

**Authors:** Bruno Citoni, Francesco Fioranelli, Muhammad A. Imran, Qammer H. Abbasi

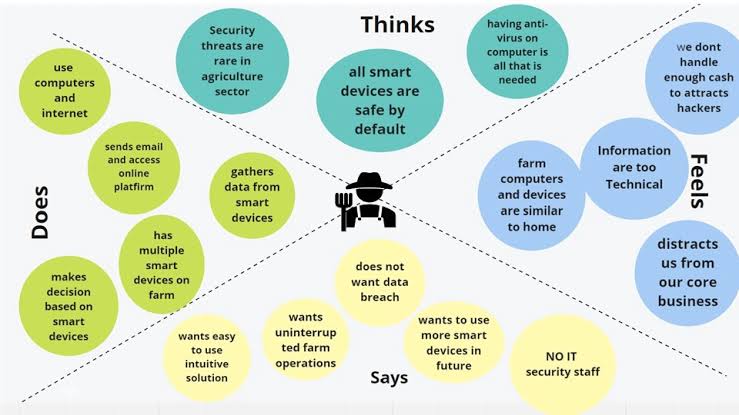
**Published:** IEEE 2019

**Description:** In this paper authors have explained about LoRaWAN which is been under the spotlight in recent years due to its suitability to be the standard communication protocol for IoT deployments. It provides long communication range and low energy consumption by drastically reducing the available data rate. They also explained about the development of LoRaWAN enabled smart agriculture test to improve the understanding about the impact of the limitations using experimental test data, and moving towards building predictive models and adaptive network management algorithms for smart farming using the data collected.

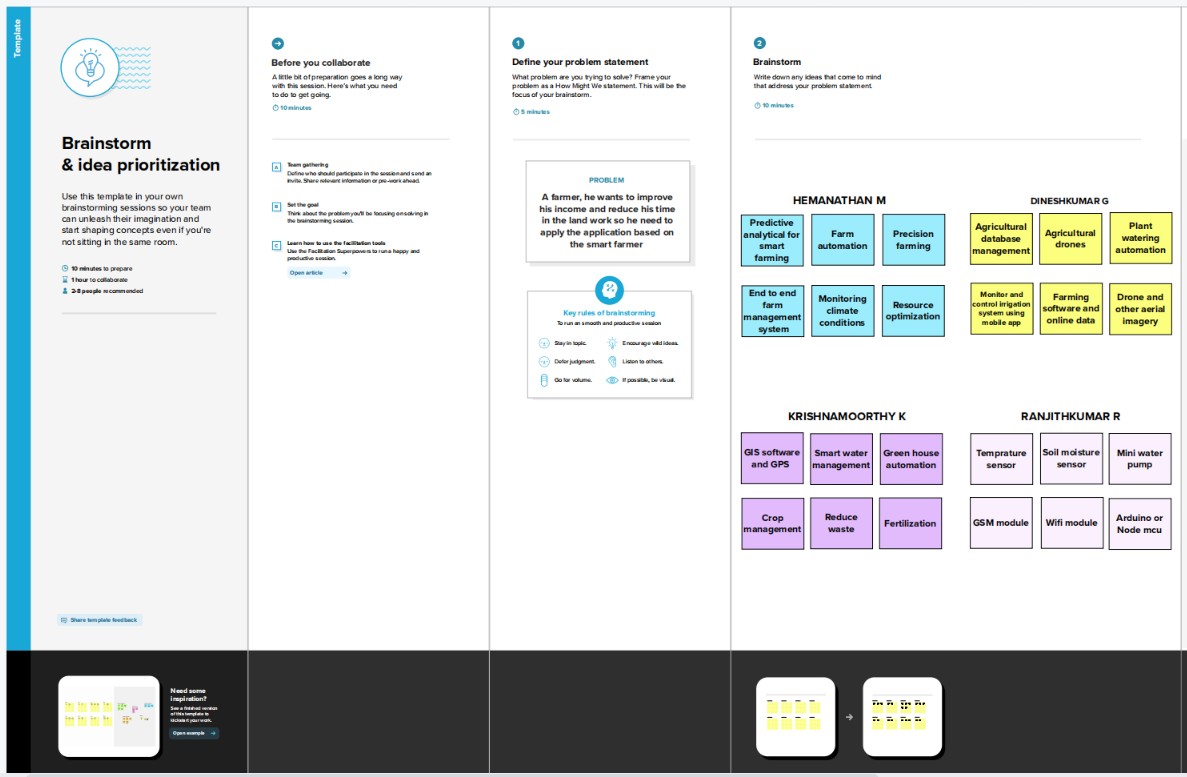
# Problem Statement Definition

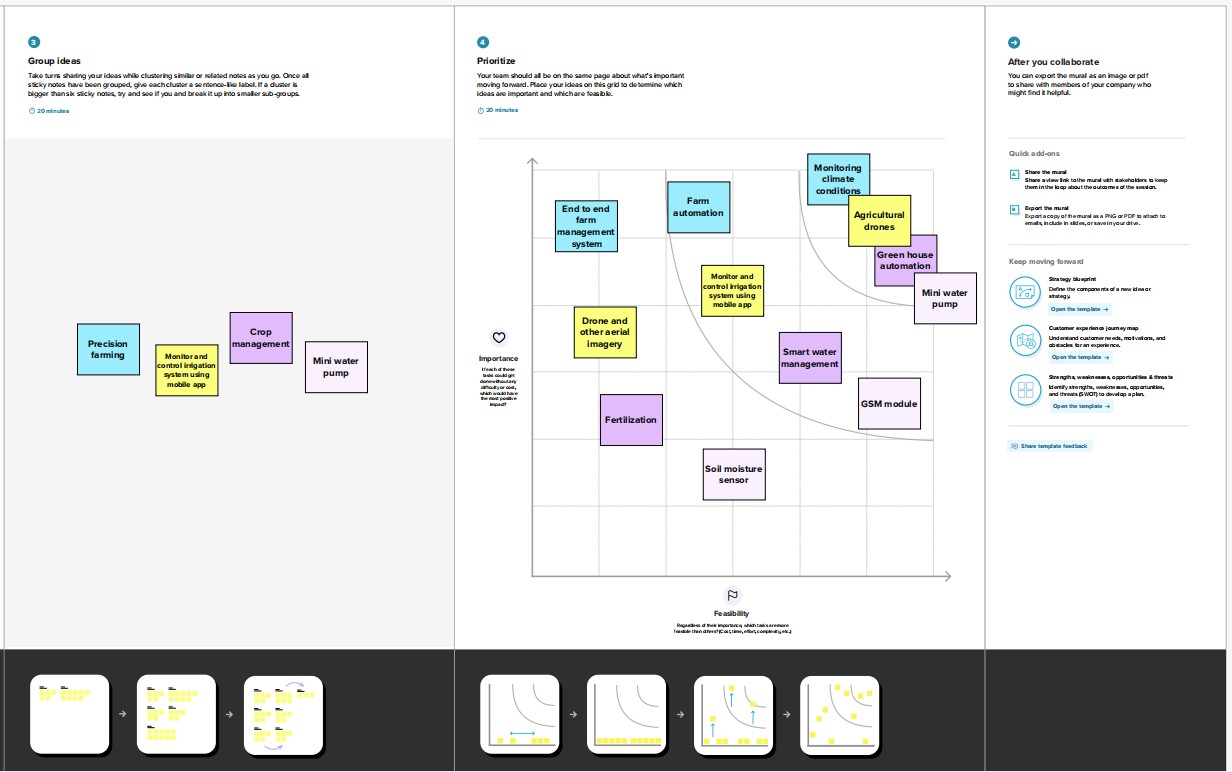
1. **IDEATION & PROPOSED SOLUTION**

# Empathy Map Canvas

****

* 1. **Ideation & Brainstorming**





# Proposed Solution

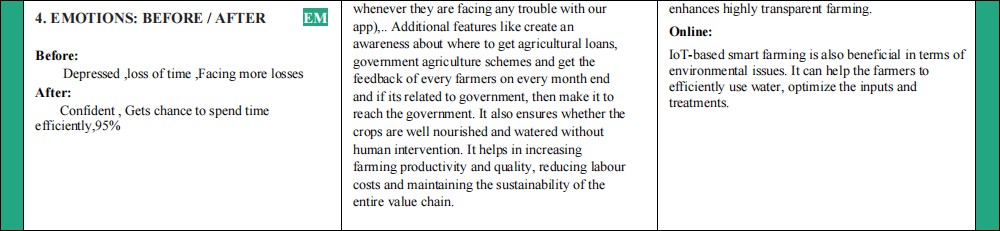
|  |  |  |
| --- | --- | --- |
| **S. No** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | * Watering the field is a difficult process, Farmers have to wait in the field until the water covers the whole farm field. * Power Supply is also one of the problems. In Village Side, the power supply may vary. * The Biggest Challenges Faced by IoT in the Agricultural Sector are Lack of Information, High Adoption, Cost and Security Concerns, etc |
| 2. | Idea / Solution description | * As is the case of precision Agriculture Smart Farming Technique Enables Farmers better to monitor the fields and maintain the humidity level accordingly. * The Data collected by sensors, In terms of humidity, temperature, moisture, and dew detections help in determining the weather pattern in Farms. So cultivation is done for suitable crops. |
| 3. | Novelty / Uniqueness | **ALERT MESSAGE –** IoT sensor nodes collect information from the farming environment, such as soil moisture, air humidity, temperature, nutrient ingredients of soil, pest images, and water quality, then transmit collected data to IoT backhaul devices.  **REMOTE ACCESS –** It helps the farmer to operate the motor from anywhere. |
| 4. | Social Impact / Customer Satisfaction | * Reduces the wages for labours who work in the agricultural field. * It saves a lot of time. * IoT can help improve customer relationships by enhancing the customer's overall experience. * Easily identify maintenance needs, build better products, send personalized communications, and more. * IoT can also help e-commerce businesses thrive and increase sales. * It make a wealthy society. |
| 5. | Business Model (Revenue Model) | No.Users(X) Vs Month(Y) |

100

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | | | | | | |  |
| 80  60  40  20  0 |  |  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 0 2 4 6 8 10 | | | | | | |
| 6. | Scalability of the Solution | Scalability in smart farming refers to the adaptability of a system to increase the capacity, for example, the number of technology devices such as sensors and actuators, while enabling timely analysis. | | | | | | | | |



# Problem Solution fit



1. **REQUIREMENT ANALYSIS**

# Functional requirement

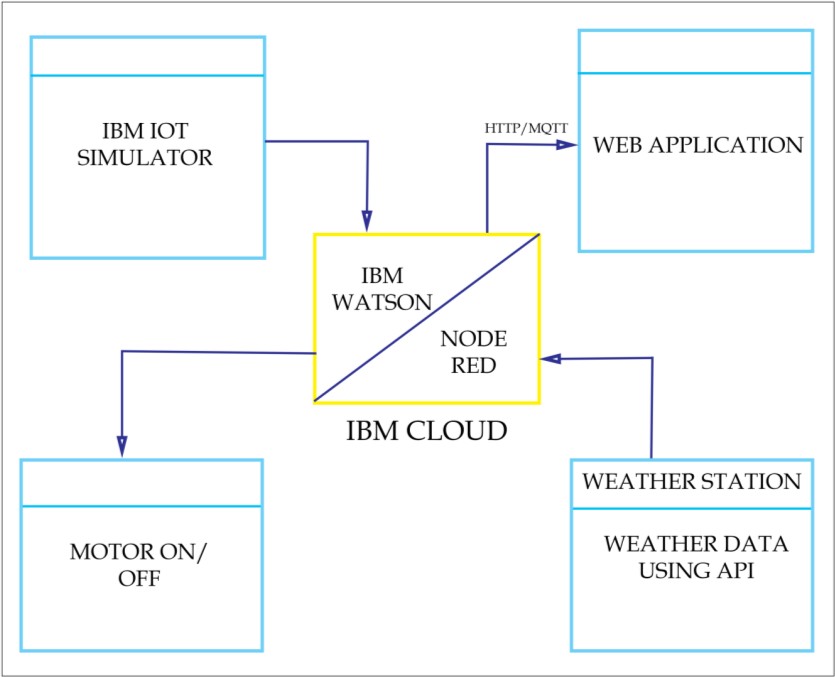
|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email Confirmation via OTP |
| FR-3 | Log in to system | Check Credentials Check Roles of Access. |
| FR-4 | Manage Modules | Manage sensor data Manage weather data Control System  Plant details |
| FR-5 | Log out | Exit |

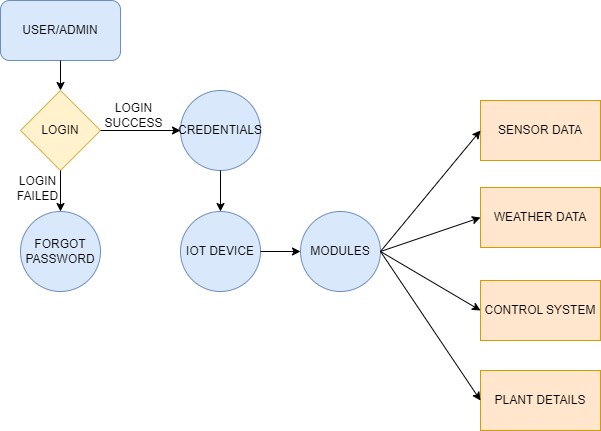
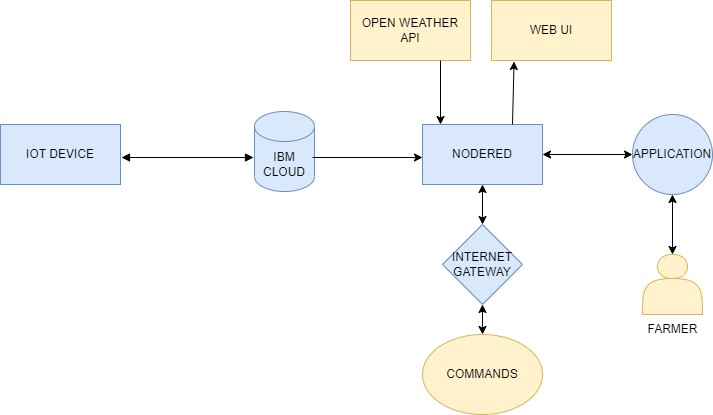
* 1. **Non-Functional requirements**

|  |  |  |
| --- | --- | --- |
| **FR**  **No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Usability includes easy learn ability, efficiency in use, remember ability, lack of errors in operation and subjective pleasure. |
| NFR-2 | **Security** | Sensitive and private data must be protected from their production until the decision-making and storage stages. |
| NFR-3 | **Reliability** | The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages. |
| NFR-4 | **Performance** | The idea of implementing integrated sensors with sensing soil and environmental or ambient parameters in farming will be more efficient for overall monitoring. |
| NFR-5 | **Availability** | Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust  temperature, humidity, etc. |
| NFR-6 | **Scalability** | Scalability is a major concern for IoT platforms. It has shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousand. |

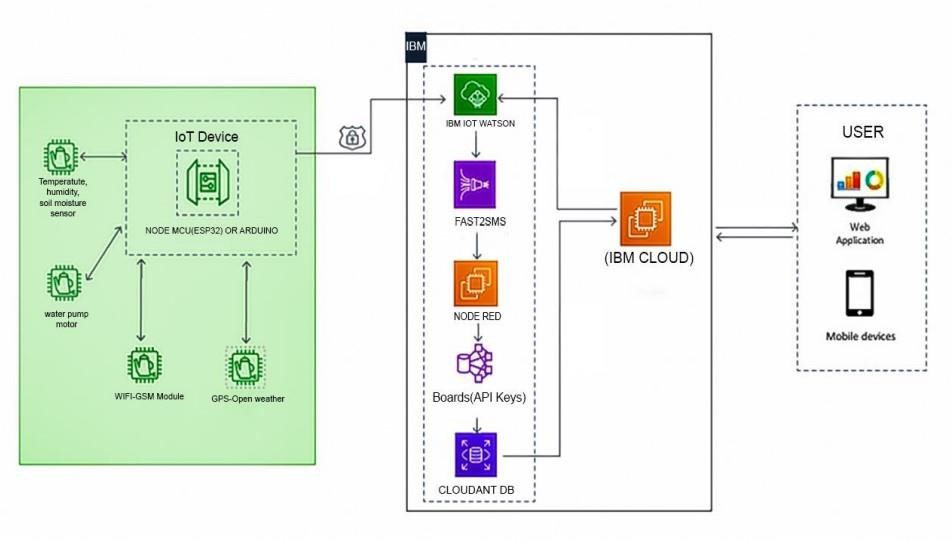
# PROJECT DESIGN

* 1. **Data Flow Diagrams**





# Solution & Technical Architecture



* 1. **User Stories**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Mobile user) | Registration | 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 | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered  for the application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  |  | USN-3 | As a user, I can register for the application through Gmail |  | Low | Sprint-1 |
|  | Login | USN-4 | As a user, I can log into the application by entering email &  password |  | Medium | Sprint-1 |
| Customer (Web user) | Dashboard | USN-5 | Main Menu Is Apprised And Manages Modules. | Can View The Main Menu  And Access The Mainframe. | High | Sprint-2 |
|  |  | USN-6 | User Can Access All Sort Of Sensor Details And Access From  All Over His Places. | Smart Farming Application. | High | Sprint-2 |
|  | Multiple User | USN -7 | User can switch the application different login credentials |  |  | Sprint-3 |
| Administrator |  |  | User Can Manage And Access The |  |  | Sprint-3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
|  |  |  | Resources And Manages Using It. |  |  |  |
|  |  |  |  |  |  |  |

# PROJECT PLANNING & SCHEDULING

* 1. **Sprint Planning & Estimation**

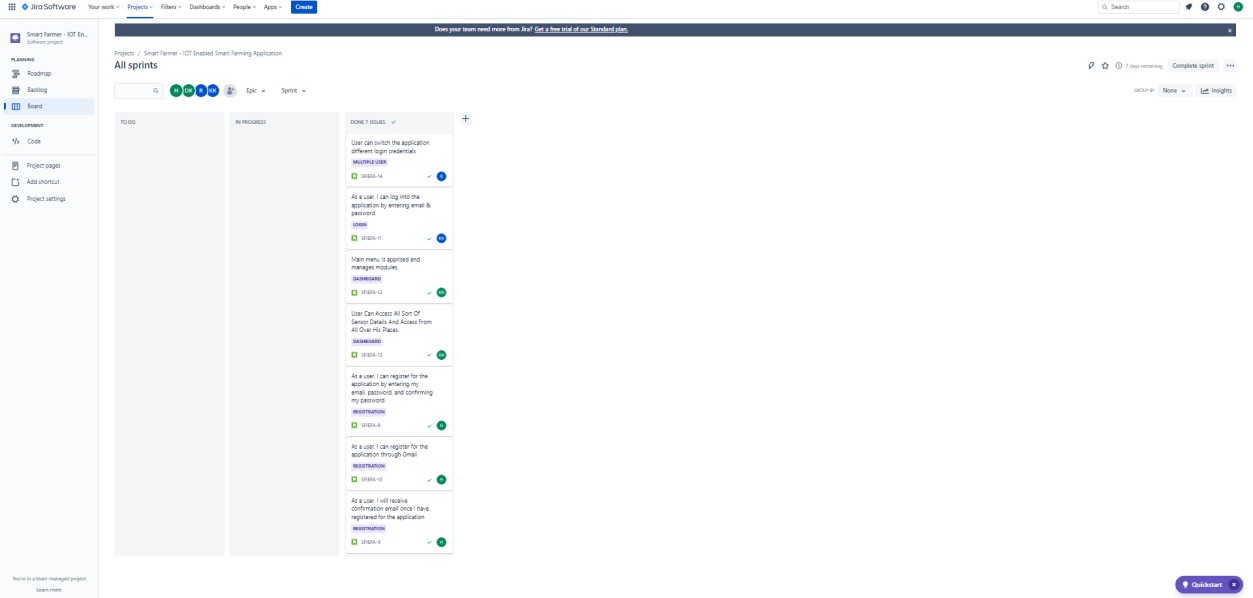
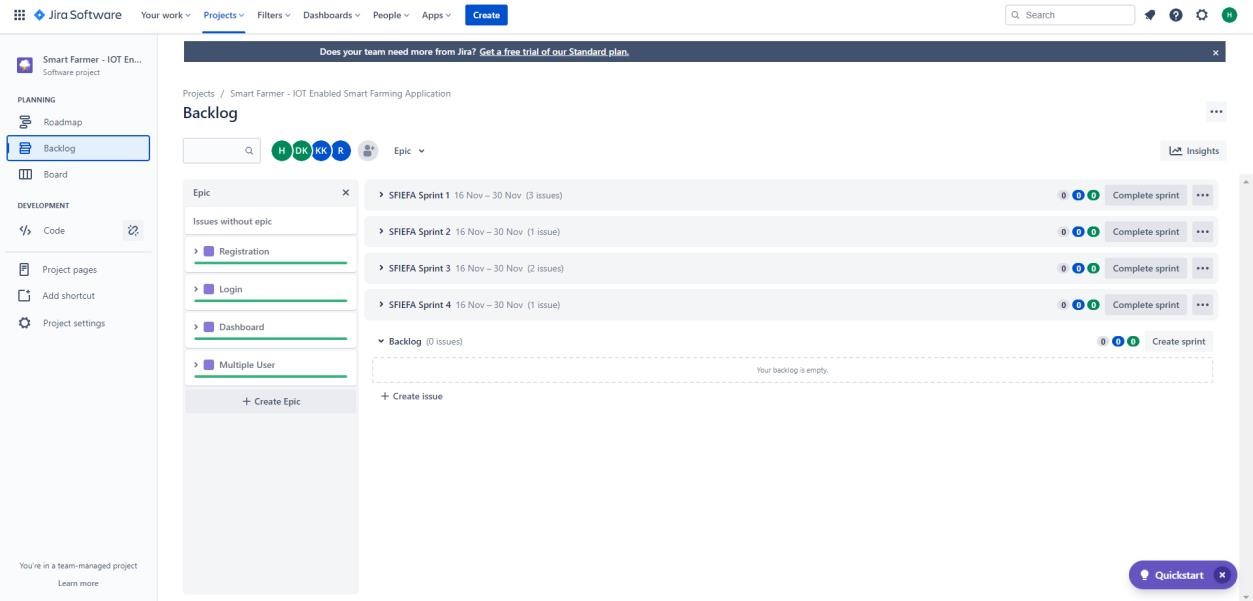
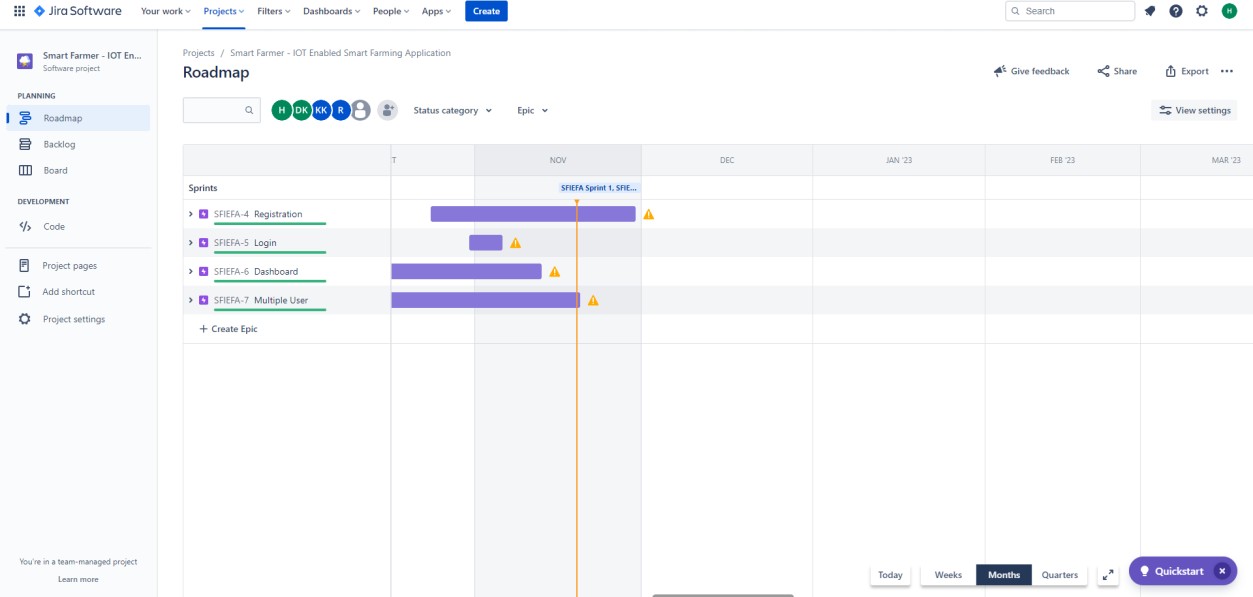
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Function al Require**  **ment (Epic)** | **User Story Numbe r** | **User Story / Task** | **Story Points** | **Prio rity** | **Team Members** |
| Customer (Mobile User) | Registrati on | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 6 | High | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
|  |  | USN-2 | As a user, I will receive confirmation email once I have registered for the  application | 8 | High | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
|  |  | USN-3 | As a user, I can register for the application through Gmail | 6 | Low | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
|  | Login | USN-4 | As a user, I can log into the application by entering email & password | 20 | Medi um | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
| Customer( Web User) | Dashboar d | USN-5 | Main menu Is apprised and manages modules. | 10 | High | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Function al Require ment (Epic)** | **User Story Numbe r** | **User Story / Task** | **Story Points** | **Prio rity** | **Team Members** |
|  |  | USN-6 | User Can Access All Sort Of Sensor DetailsAnd Access From All Over His Places. | 10 | High | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
|  | Multiple User | USN -7 | User can switch the application different login credentials | 20 | High | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
| Administr ator |  |  | User Can Manage And Access The Resources And Manages Using It. |  |  | BAVADHARANI.K  SATHISH.M  ARUN KUMAR.A  ESAKKIRAJAN.M |
|  |  |  |  |  |  |  |

# 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 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

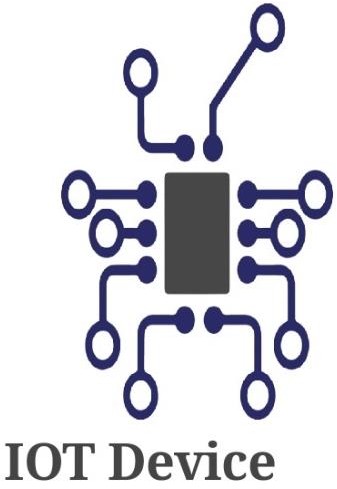
* 1. **Reports from JIRA**

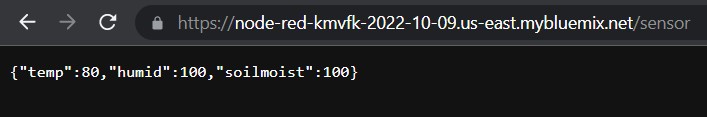
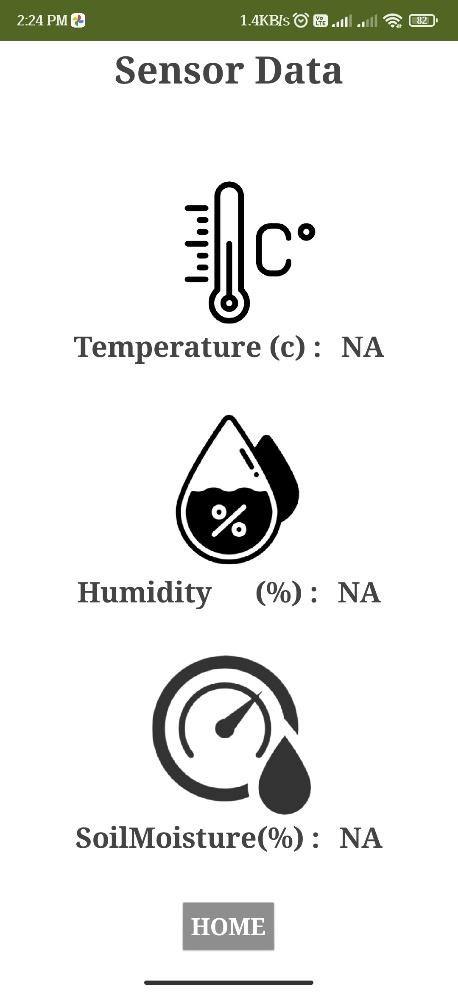


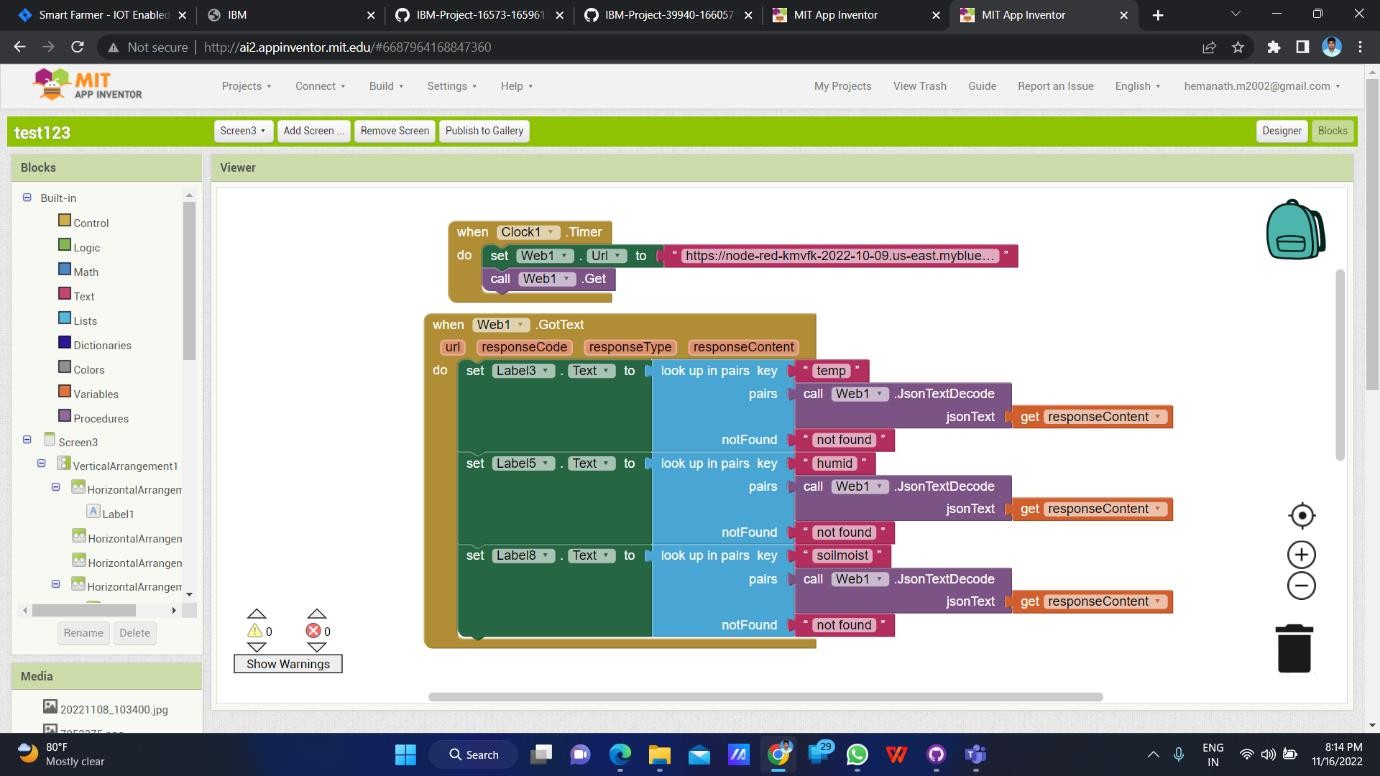
# CODING & SOLUTIONING

* 1. **Feature 1**

Monitor the Realtime sensor data

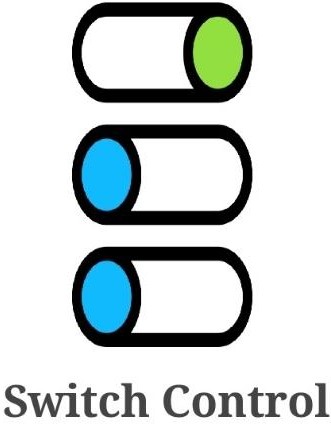


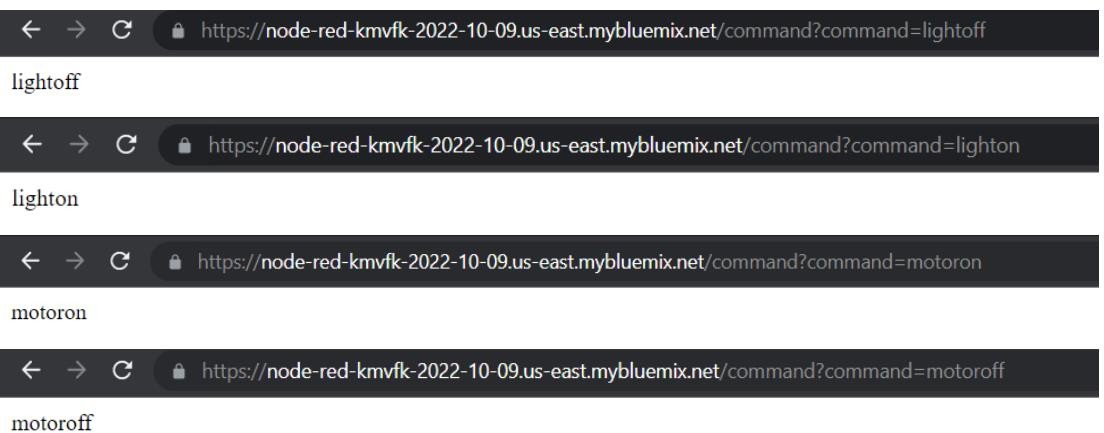
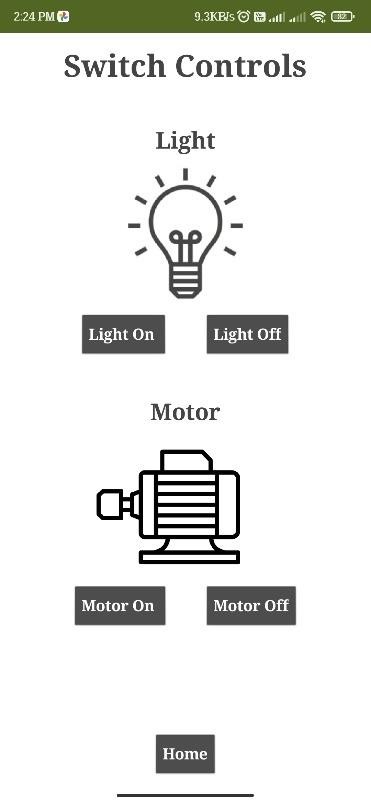


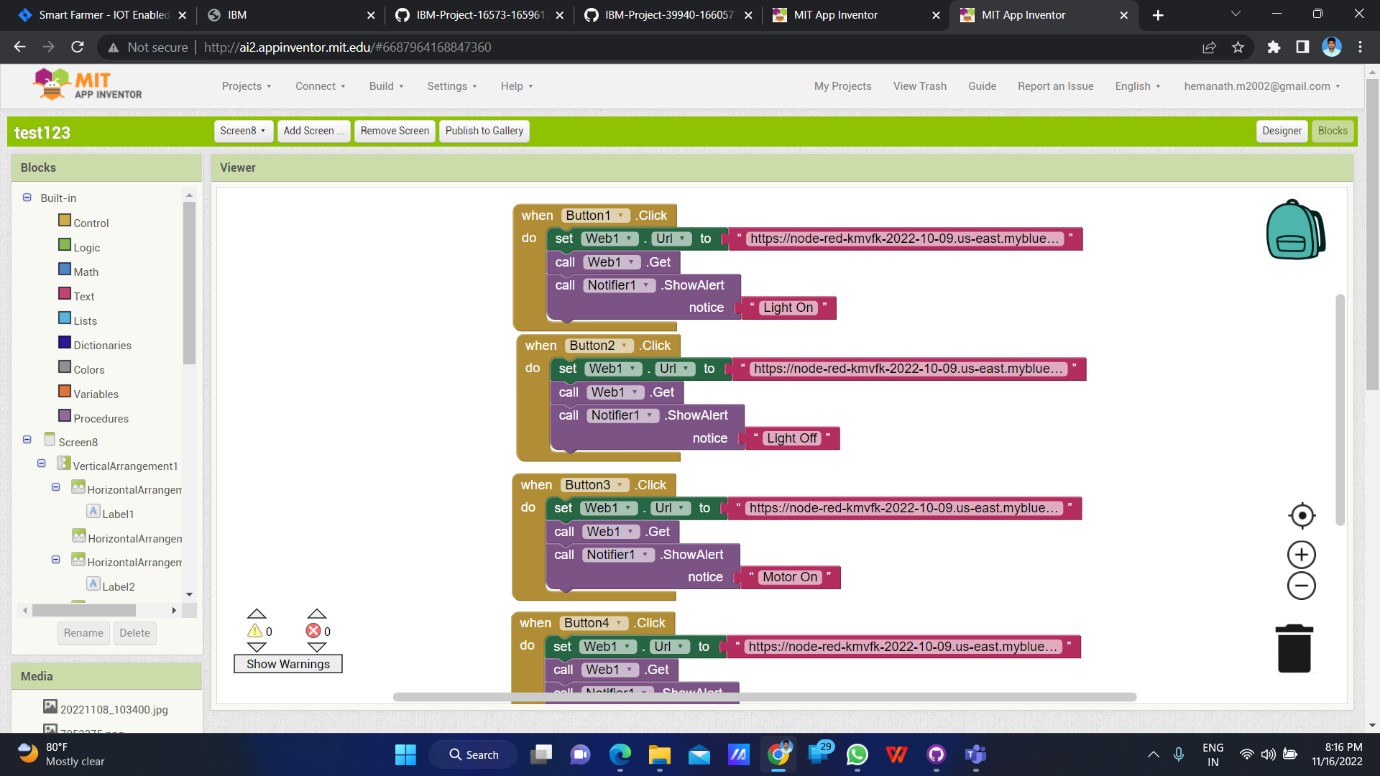


# Feature 2

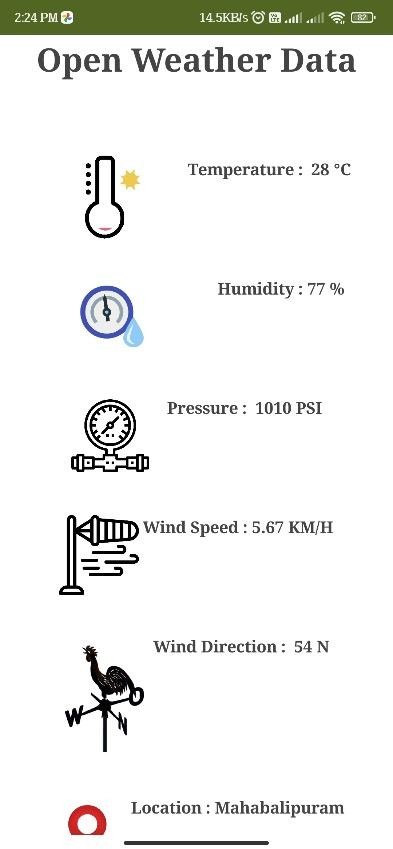
Control the switch remotely

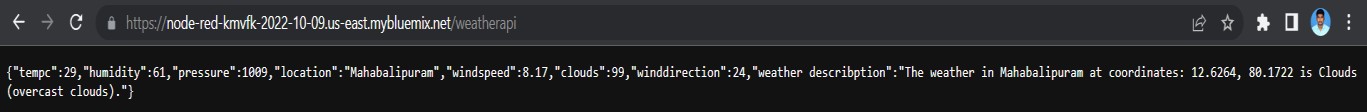


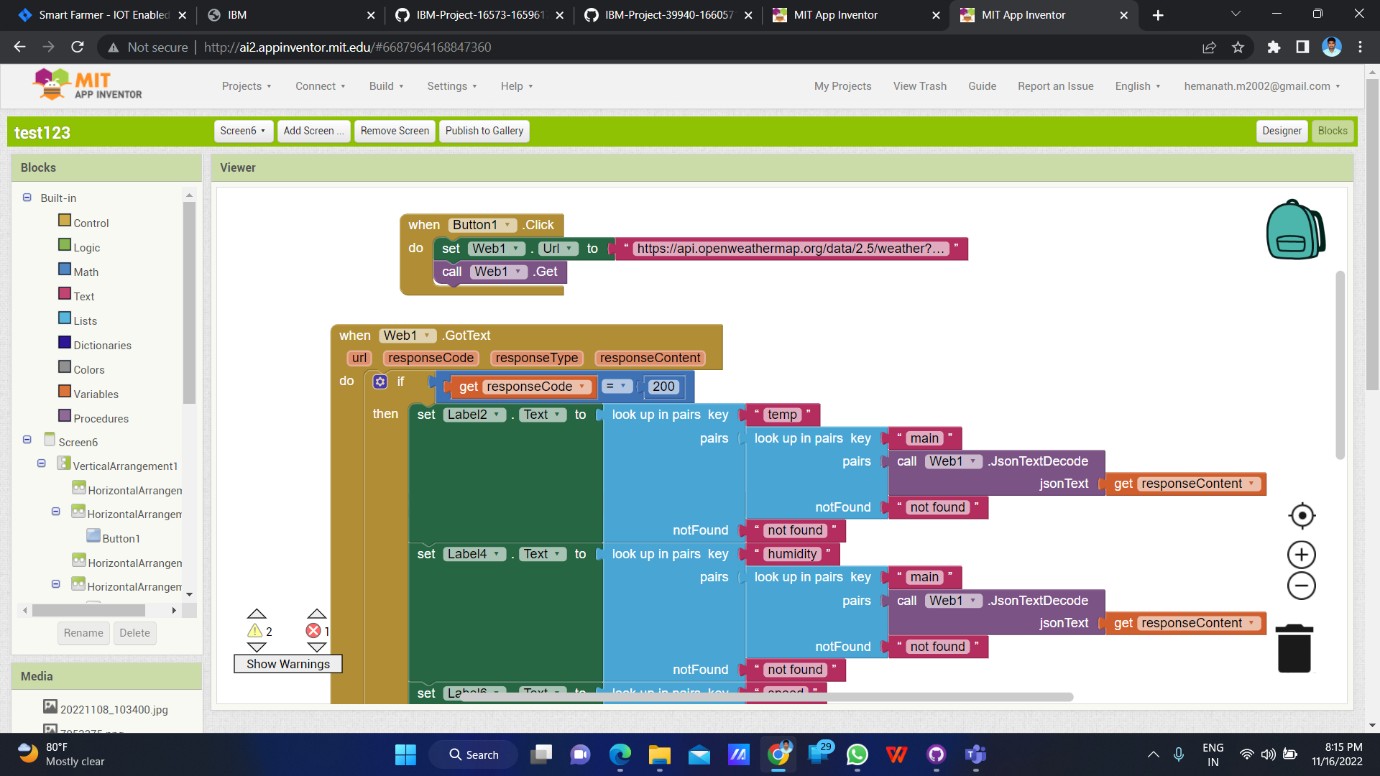




# Feature 3

Get the Realtime weather data





# TESTING

* 1. **Test Cases**
     1. Verify user is able to see the Login/Signup popup when user clicked on My account button.
     2. Verify the UI elements in Login/Signup popup.
     3. Verify user is able to log into application with Valid credentials.
     4. Verify user is able to log into application with InValid credentials.

# 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).

# 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 | 10 | 4 | 2 | 3 | 20 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 11 | 2 | 4 | 20 | 37 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 24 | 14 | 13 | 26 | 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 | 7 | 0 | 0 | 7 |
| Client Application | 51 | 0 | 0 | 51 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 3 | 0 | 0 | 3 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |
| Version Control | 2 | 0 | 0 | 2 |

# RESULTS

**9.1 Performance Metrics**

Hence a helpful and useful system is built for farmers to assist them in farming

and also prevent them from natural calamities. It also saves farmers time to maintain all these things as this is working on cloud he can turn on/off motor from anywhere so basically it helps farmers and make them relived thus helping our economy to grow.

# ADVANTAGES & DISADVANTAGES ADVANTAGE

* 1. Communicating the device at larger distance through web application. It will play an important role in reducing the man power and travelling expenses of a farmer.
  2. Monitoring the parameter like temperature, humidity etc will play an important role in improving the growth of the plant.
  3. 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 prevent their plants from natural calamities.

# DISADVANTAGE

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

# CONCLUSION

The various parameters like temperature, humidity etc were monitored using web application. The data from weather station like wind speed, temperature, humidity etc were displayed in the web browser. The device like motor, light etc can also controlled by the web application.

# FUTURE SCOPE

1. The various data’s of soil nutrients is not added in the web browser, that can be added to the web application.
2. Long range forecast is not available in the web application, it can also be added to provide accurate information about weather.
3. Controlling the device through mobile application and voice will play important role in enhancing this project.
4. Providing the GPS and GIS information will also improve productivity of the farmer.

# 12. APPENDIX

**12.1 Source Code 1)Python Code**

import time import sys

import ibmiotf.application import ibmiotf.device import random

organization = "22r9m3" deviceType = "123"

deviceId = "1234567" authMethod = "token" authToken = "12345678"

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command']) status=cmd.data['command']

if status=="lighton": print ("led is on")

elif status == "lightoff": print("led is off")

elif status == "motoron": print("motor is on")

elif status == "motoroff": print("motor is off")

else :

print ("please send proper command")

#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()

deviceCli.connect() while True:

temp=random.randint(0,100) humid=random.randint(0,100) soilmoist=random.randint(0,100)

data = { 'temp' : temp, 'humid': humid, 'soilmoist': soilmoist } def myOnPublishCallback():

print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %

humid,"Soilmoisture = %s %%" % soilmoist, "to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connected to IoTF") time.sleep(1)

deviceCli.commandCallback = myCommandCallback deviceCli.disconnect()

# 2)Wokwi Simulator

#include <WiFi.h> #include <PubSubClient.h> #include "DHT.h"

#define DHTPIN 15 #define DHTTYPE DHT22 #define LED 2

#define MOTOR 4

DHT dht (DHTPIN, DHTTYPE);

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength);

#define ORG "22r9m3"//IBM ORGANITION ID

#define DEVICE\_TYPE "123"//Device type mentioned in ibm watson IOT Platform #define DEVICE\_ID "1234567"//Device ID mentioned in ibm watson IOT Platform #define TOKEN "12345678" //Token

String data3; float h, t;

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name

char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event perform and format in which data to be send

char subscribetopic[] = "iot-2/cmd/command/fmt/String";// cmd REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING

char authMethod[] = "use-token-auth";// authentication method char token[] = TOKEN;

char clientId[] = "d:" ORG ":" DEVICE\_TYPE ":" DEVICE\_ID;//client id

WiFiClient wifiClient;

PubSubClient client(server, 1883, callback ,wifiClient);

void setup()

{

Serial.begin(115200); dht.begin(); pinMode(LED,OUTPUT); pinMode(MOTOR,OUTPUT);

delay(10); Serial.println(); wificonnect(); mqttconnect();

}

void loop()

{

h = dht.readHumidity();

t = dht.readTemperature(); Serial.print("temp:"); Serial.println(t); Serial.print("humid:"); Serial.println(h);

PublishData(t, h); delay(1000);

if (!client.loop()) { mqttconnect();

}

}

void PublishData(float temp, float humid) { mqttconnect();

String payload = "{\"temp\":"; payload += temp;

payload += "," "\"humid\":"; payload += humid;

payload += "," "\"soilmoist\":"; payload += humid;

payload += "}";

Serial.print("Sending payload: "); Serial.println(payload);

if (client.publish(publishTopic, (char\*) payload.c\_str())) { Serial.println("Publish ok");

} else {

Serial.println("Publish failed");

}

}

void mqttconnect() {

if (!client.connected()) { Serial.print("Reconnecting client to "); Serial.println(server);

while (!!!client.connect(clientId, authMethod, token)) { Serial.print(".");

delay(500);

}

initManagedDevice(); Serial.println();

}

}

void wificonnect()

{

Serial.println(); Serial.print("Connecting to ");

WiFi.begin("Wokwi-GUEST", "", 6);

while (WiFi.status() != WL\_CONNECTED) { delay(500);

Serial.print(".");

}

Serial.println(""); Serial.println("WiFi connected"); Serial.println("IP address: "); Serial.println(WiFi.localIP());

}

void initManagedDevice() {

if (client.subscribe(subscribetopic)) { Serial.println((subscribetopic)); Serial.println("subscribe to cmd OK");

} else {

Serial.println("subscribe to cmd FAILED");

}

}

void callback(char\* subscribetopic, byte\* payload, unsigned int payloadLength)

{

Serial.print("callback invoked for topic: "); Serial.println(subscribetopic);

for (int i = 0; i < payloadLength; i++) { data3 += (char)payload[i];

}

Serial.println("data: "+ data3); if(data3=="lighton")

{

Serial.println(data3); digitalWrite(LED,HIGH);

}

else if(data3=="motoron")

{

Serial.println(data3); digitalWrite(MOTOR,HIGH);

}

else if(data3=="motoroff")

{

Serial.println(data3); digitalWrite(MOTOR,LOW);

}

else

{

Serial.println(data3); digitalWrite(LED,LOW);

}

data3="";

}

# GitHub & Project Demo Link

GitHub **: https://github.com/IBM-EPBL/IBM-Project-53971-1661585887**

Project Demo Link **:** [**https://youtu.be/saJbdO5B1sw**](https://youtu.be/saJbdO5B1sw)