PROJECT REPORT

1.INTRODUCTION

1.1. Project Overview

This smart agriculture using IOT system is powered by Arduino, it consists of Temperature sensor,Moisture sensor, water level sensor, DC motor and GPRS module. When the IOT based agriculture monitoring system starts it checks the water level, humidity and moisture level. It sends SMS alert on the phone about the levels. Sensors sense the level of water if it goes down, it automatically starts the water pump. If the temperature goes above the level, fan starts. This all is displayed on the LCD display module. This all is also seen in IOT where it shows information of Humidity, Moisture and water level with date and time, based on per minute. Temperature can be set on a particular level, it is based on the type crops water cultivated. If we want to close the water forcefully on IOT there is button given from where pump can be forcefully stopped

1.2. Purpose

Smart farming based on IoT technologies enables growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the number of journeys the farm vehicles have made, and enabling efficient utilization of resources such as water, electricity, etc. IoT smart farming solutions is a system that is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, crop health, etc.) and automating the irrigation system. The farmers can monitor the field conditions from anywhere.

2.LITERATURE SURVEY

2.1. Existing Problem

IoT Based Crop-field monitoring an irrigation automation system describes how to monitor a crop field. A system is developed by using sensors and according to the decision based on sensed data, the irrigation system is automated. Through wireless transmission the sensed data is forwarded to web server database. If the irrigation is automated then the moisture and temperature fields are decreased below the potential range. The user can monitor and control the system remotely with the help of application which provides a web interface to user. The system focuses on developing devices and tool to manage, display and alert the users using the advantages of a wireless sensor network system. It aims at making agriculture smart using automation and IoT technologies.

The cloud computing devices are used at the end of the system that can create a whole computing system from sensors to tools that observe data from agriculture field. Here one can access and control the agriculture system in laptop, cell phone or a computer.

2.2. References

1. T.Rajesh, Y.Thrinayana and D.Srinivasulu "IoT based smart agriculture monitoring system", International Research Journal of Engineering and Technology, Vol.07.

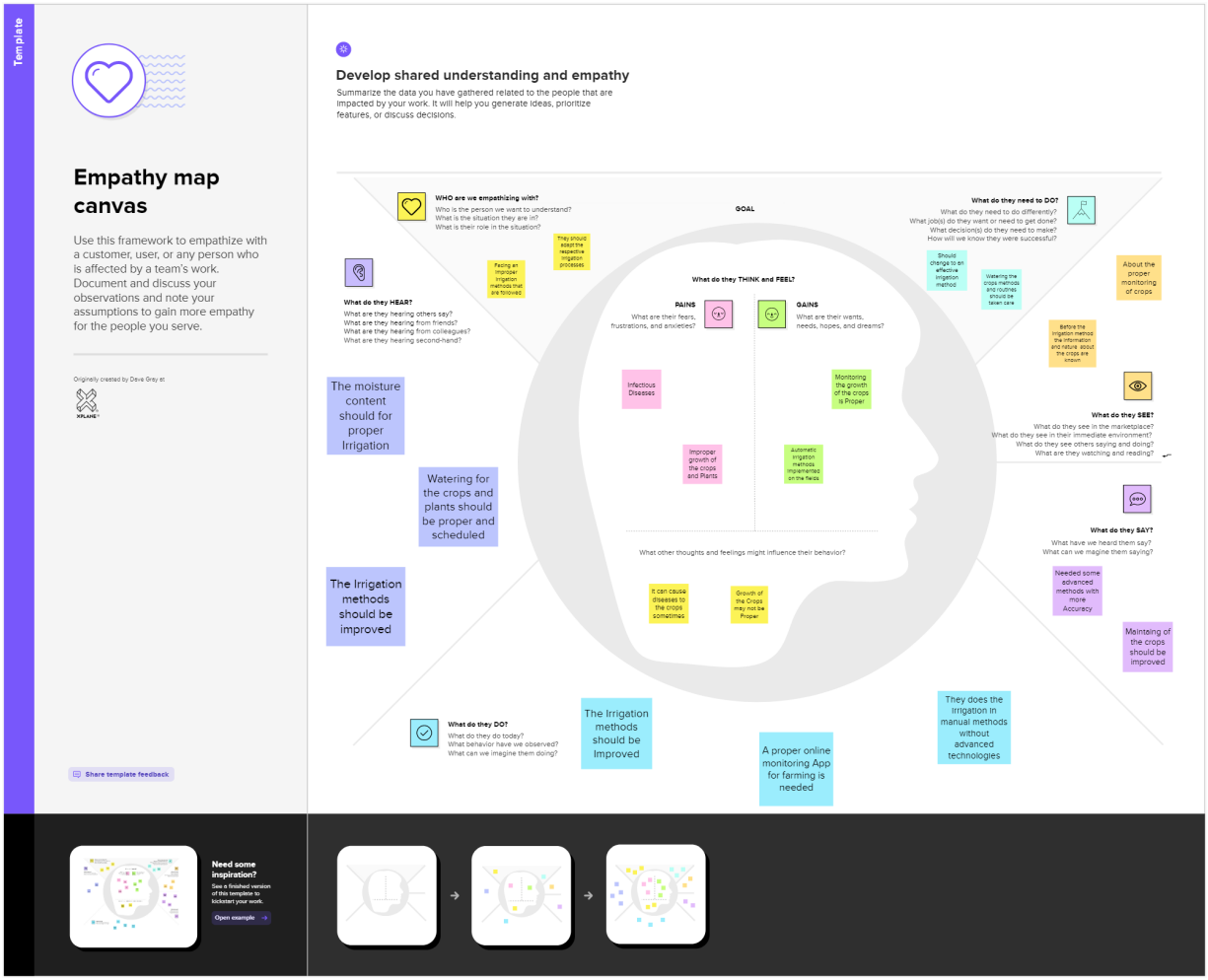
2. Rajalakshmi.P and S.Devi Mahalakshmi, "IoT Based Crop Monitoring and Irrigation Automation", 10th International conference on Intelligent systems and control (ISCO), 2016.

2.3. Problem Statement Definition

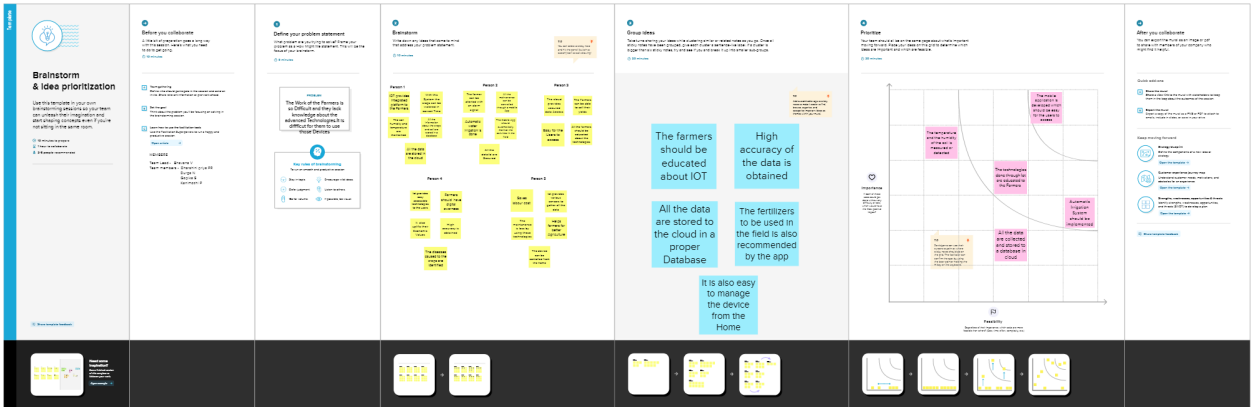
Mr.Rajesh is a Farmer. His Son is an Engineer. Though his son is an Engineer he is also interested in Farming So he joined with his Father and helped him. When he started working with his Father he noticed that they have do every work manually and that required a lot of Hard Work. Then he decided that they have to think of alternative ways which will reduce their work and increase their productivity. They need some advanced technologies where all their work can be done easily. So the son has to actively research on the crop planting ideas and techniques and have to come up with a better solution for more yield.

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas



3.2. Ideation & Brainstorming



3.3 Proposed Solution

1.Problem Statement (Problem to be solved):

Ideally, each field should get just the right amount of water at just the right time.

Under-watering causes crop stress and yield reduction.

Overwatering can also cause yield reduction and consumes more water and fuel than necessary and leads to soil erosion and fertilizer, herbicide, and pesticide runoff.

2. Idea / Solution description:

Smart Farming systems uses modern technology to increase the quantity and quality of agricultural products.

This enables the farmers better to monitor the fields and maintain the humidity level accordingly.

3. Novelty / Uniqueness:

The development of lightweight and powerful hyperspectral snapshot cameras that can be used to calculate biomass development and fertilization status of crops.

Moreover, decision-tree models are available now that allow farmers to differentiate between plant diseases based on optical information

Virtual fence technologies allow cattle herd management based on remote-sensing signals and sensors or actuators attached to the livestock.

4. Social Impact / Customer Satisfaction:

Iot helps in improving customer relationships by enhancing customers overall performance

It also saves a lot of Time

This technology reduces the works to be done by the farmers

5. Business Model (Revenue Model):

The revenue model represents a gradual rise in both no of users and the income

6.Scalability of the Solution:

Scalability in smart farming refers to the adaptability of a system to increase the capacity.

3.4 Problem Solution fit

Project Title:SmartFarmer-IoT enabled Smartfarming Application **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID30750

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	▯ Registration through Form ▯ Registration through Gmail ▯ Registration through LinkedIN
FR-2	User Confirmation	▯ Confirmation via Email ▯ Confirmation via OTP
FR-3	Sensor function for framing system	▯ These equipment equipped with sensors that provide information about soil temperature, air temperature, rainfall, leaf wetness, chlorophyll, wind direction, solar radiation, relative humidity, atmospheric pressure etc.
FR-4	Check weather Details	▯ Temperature Details ▯ Humidity Details

4.2 Non-Functional requirements

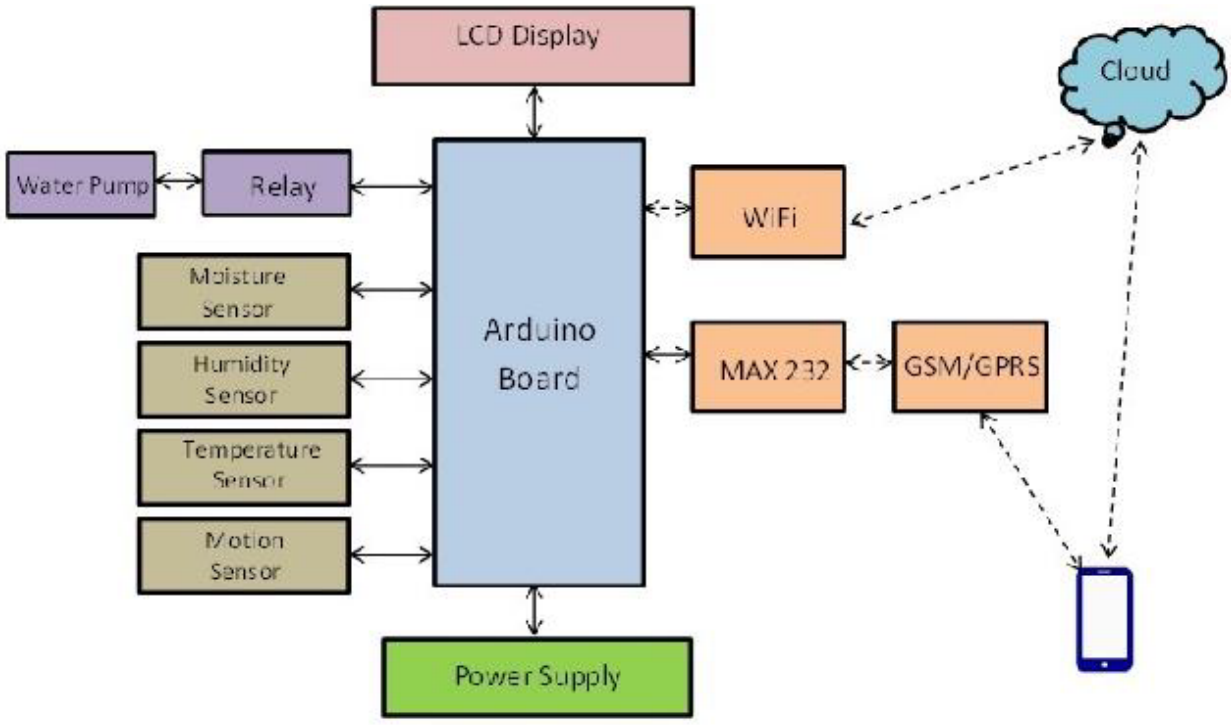
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	▯ This type of non functional requirement is concerned with characteristics such as aesthetics and consistency of the user interface
NFR-2	Security	▯ Security is a non-functional requirement assuring all data inside the system or its part will be protected against malware attacks or unauthorized access
NFR-3	Reliability	▯ Reliability specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions.

NFR-4	**Performance**	**The areas of performance, availability, reliability, usability, flexibility, configurability, integration, maintainability, portability, and testability.**

5. PROJECT DESIGN

5.1. Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2. Solution & Technical Architecture

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.

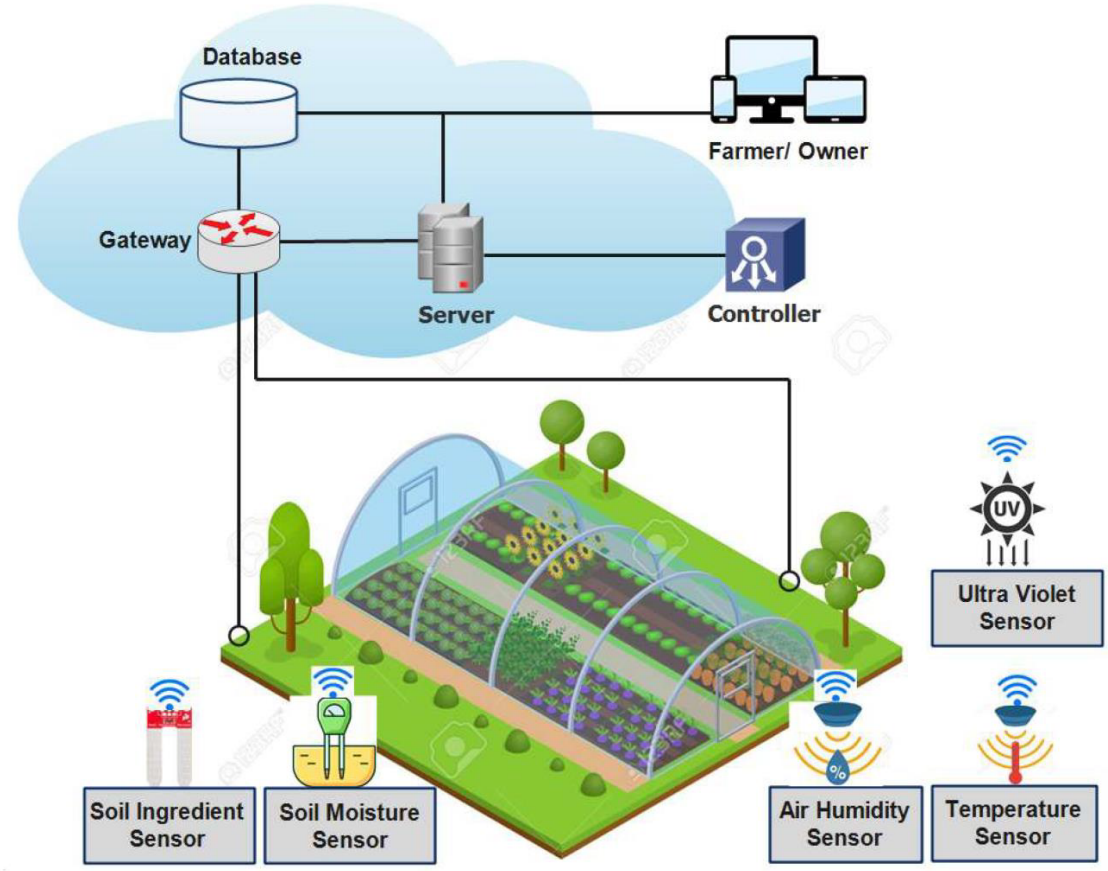
The different soil parameters are sensed using different sensors, and the obtained value is stored in the IBM cloud.

Arduino UNO is used as a processing that processes the data obtained from sensors and weather data from weather API.

Node red is used as a programming tool to wire the hardware, software, and APIs.

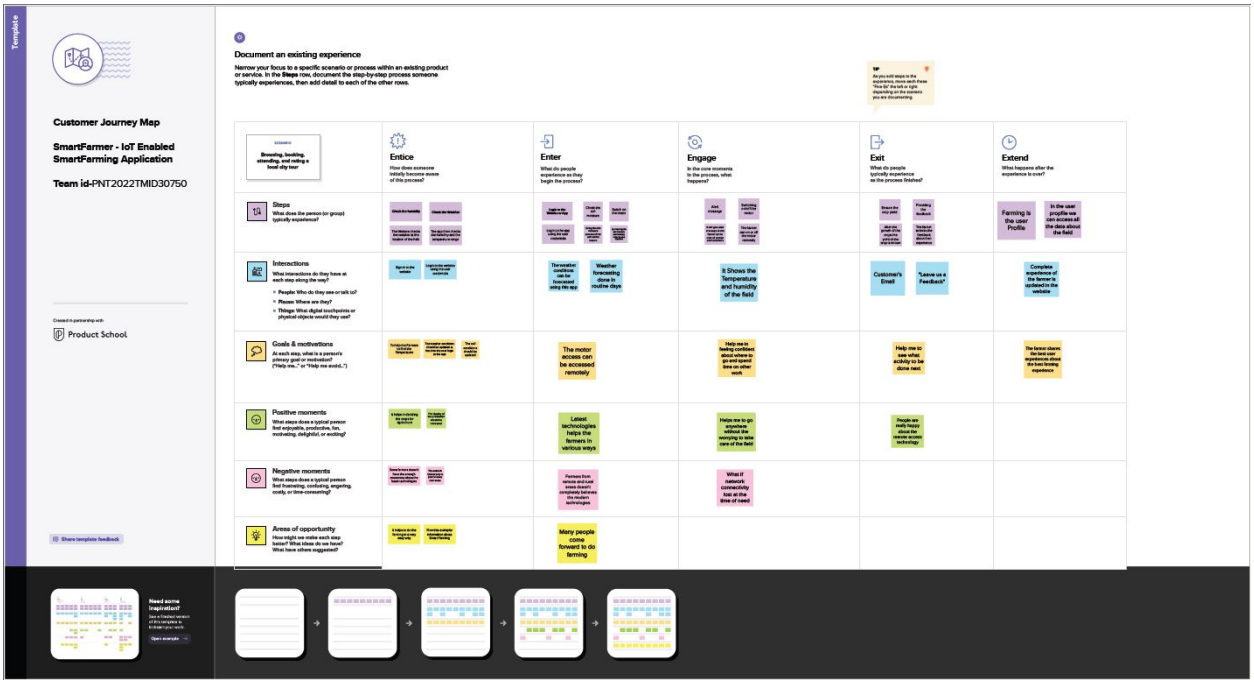
All the collected data are provided the user through a mobile application that was developed using the MIT app inventor.

The user could make decision through an app, whether to water the crop or not depending upon the sensor values. By using this app they can remotely access the motor switch.



Architecture for IoT Enabled smart farming application

5.3 User Stories



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

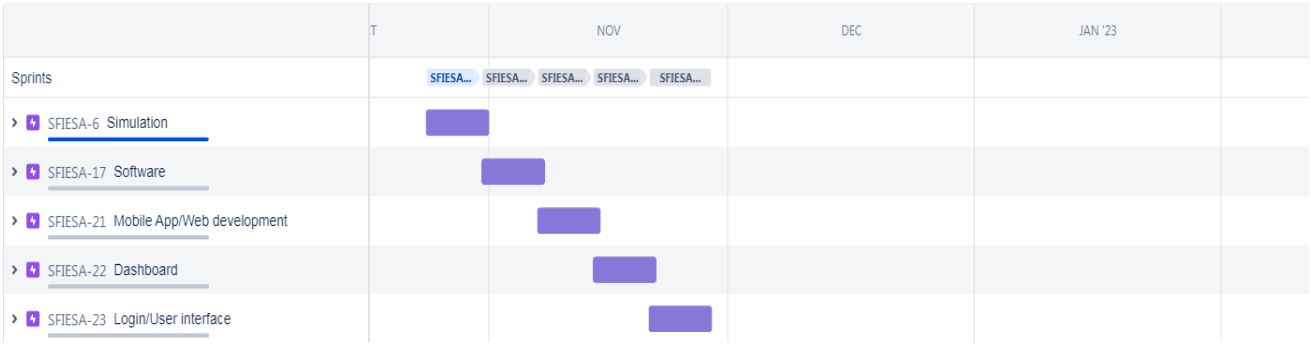
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation	USN-1	Create the Simulation by connecting the sensors by using the Arduino and connect with the code	2	High	Bhavana V Dharshini Priya PR Gopika S
Sprint-2	Software	USN-2	Create the device on IBM cloud platform and the node red platform to set the iot device workflow	2	High	Durga N Kanimozhi P Gopika S

Sprint - 3	**Mobile App/Web Application**	**USN-3**	**Develop the Application for Smartfarmer Iot enabled smart farming Application project using MIT App Inventor.**	**2**	**High**	**Dharshini Priya PR Bhavana V Kanimozhi P**
Sprint-4	**Dashboard**	**USN-4**	**Design all the modules and create all the features of the App and test the application.**	**2**	**High**	**Kanimozhi P Dharshini Priya PR Durga N**
Sprint - 4	**Login/User Interface**	**USN-5**	**Using the login make connections with the end users and make them interact with the software**	**2**	**High**	**Gopika S Bhavana V Durga N**

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**	**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**	**18 Nov 2022**
Sprint-4	**20**	**6 Days**	**14 Nov 2022**	**19 Nov 2022**	**20**	**18 Nov 2022**

6.3 Reports from JIRA



7. CODING & SOLUTIONING

7.1 Feature 1

-> Connecting the python code to the IBM Watson Cloud Platform

->Setting the organization id and token to make the connection with cloud

#Providing the IBM Watson Device Credentials

organization = " o58zsl "

deviceType = " abcd "

deviceId = "1234"

authMethod = "token"

authToken = "12345678"

7.2. Feature 2

-> Importing all the necessary libraries and Developing Python code

import time

import sys

import ibmiotf.application

import ibmiotf.device

```
import random
# Initializing  GPIO

def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['command'])
   status=cmd.data['command']
   if status=="motoron":
      print ("motor is on")
   if status=="motoroff" :
      print ("motor is off")
   if status=="manual" :
      print ("Motor Control is in Manual Mode")
   if status=="automatic" :
      print ("Motor control is in Automatic Mode")
      if soilmoisture > 600:
         print ("motor is on")


   #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:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    soilmoisture=random.randint(0,1023)
    Phlevel=random.randint(0,14)
    y=soilmoisture


    data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture ,'Phlevel' : Phlevel }
    #print data
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Soil Moisture is %s %%" % soilmoisture,"PH level is %s" %Phlevel ,"to IBM Watson")


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


    deviceCli.commandCallback = myCommandCallback
```

 # Disconnecting the device and application from the cloud

deviceCli.disconnect()

8. TESTING

8.1 Test Cases

Testcases	Test Scenarios
Testcase 1	Verify user is able to see the Login/Signup popup when user logged on to the app.
Testcase 2	Verify the UI elements in Signup popup
Testcase 3	Verify user is able to log into application with Valid credentials
Testcase 4	Verify user is able to log into application with InValid credentials
Testcase 5	Verify user is able to log into application with InValid credentials

8.2 User Acceptance Testing

Test Case Id	Component	Expected Result	Actual Result	Status
Testcase 1	Home Page_1	Login/Signup popup should display	Working as expected	Pass
Testcase 2	Home Page_2	Application should show below UI elements: a.user text box	Working as expected	Pass

		b.password text box c.submit button with red colour		
Testcase 3	Login Page_3	User should navigate to the next page	Working as expected	Pass
Testcase 4	Login Page_4	Application should show 'Incorrect Credentials ' message	Working as expected	Pass
Testcase 5	Login Page_4	Application should show 'Incorrect Credentials ' message	Working as expected	Pass

9. RESULTS

9.1 Performance Metrics

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Risk Score
1	Smart Farmer - IoT Enabled Smart Farming Application	Increase quality and quantity of agriculture.	Low	Temperature sensor, soil moisture sensor	Node-red, MIT inventor	Orange
2.	Smart Farmer - IoT Enabled Smart Farming Application	Help measuring the soil moisture and enabling methods to turn on or turn off the motors.	High	Smart Phones	Analyze temperature ,humidity,soil moisture	Orange

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- **Water irrigation is automatically done**
- **Protection of agriculture field**
- **Continuous monitoring control of field Smart work**
- **I will intimate the status off the land automatically through internet.**

DISADVANTAGES OF EXISTING SYSTEM

- **It is not a secure system.**
- **There is no motion detection for protection of agriculture field.**
- **Automation is not available.**

11. CONCLUSION

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted dependent on the local environmental variables. This technology also detects animal invasions, which are a major cause of crop loss. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source. It helps in deciding the farmer to whether to do irrigation or not to do.Continuousinternet connectivity isrequired for continuous monitoring of data from sensors.This also can be overcomed by using GSM unit as an alternative of mobile app. By GSM, SMScan be sent to farmers phone.

12. FUTURE SCOPE

In the current project we have implemented the project that can protect and maintain the the crop. In this project the farmer monitor and control the field remotely. In future we can add or update few more things to this project.

• We can create few more models of the same project ,so that the farmer 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. It reduces 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 farmer to get information if there is a internet issues

• We can add camera feature so that the farmer can monitor his field in real time. This helps in avoiding thefts.

13. APPENDIX

Source Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Providing the  IBM Watson Device Credentials
organization = " o58zsl "
```

```
deviceType = " abcd "
deviceId = "1234"
authMethod = "token"
authToken = "12345678"
global y

# Initializing  GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    if status=="motoroff" :
        print ("motor is off")
    if status=="manual" :
        print ("Motor Control is in Manual Mode")
    if status=="automatic" :
        print ("Motor control is in Automatic Mode")
        if soilmoisture > 600:
            print ("motor is on")

    #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:
    #Get Sensor Data from DHT11
    temp=random.randint(0,100)
    Humid=random.randint(0,100)
    soilmoisture=random.randint(0,1023)
    Phlevel=random.randint(0,14)
    y=soilmoisture

    data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture ,'Phlevel' :
Phlevel }
    #print data
    def myOnPublishCallback():
      print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid,"Soil Moisture is %s %%" % soilmoisture,"PH level is %s" %Phlevel ,"to IBM
Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
```

```
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)

        deviceCli.commandCallback = myCommandCallback

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

GitHub & Project Demo Link:

Github link : https://github.com/IBM-EPBL/IBM-Project-39940-1660571002

Project Demo Link :https://photos.app.goo.gl/JPMqJGT6f6JZHz3X7