<u>PROJECT REPORT</u>

Team ID	***PNT2022TMID01185***
Project Name	***Smart Farmer - IoT Enabled Smart FarmingApplication***

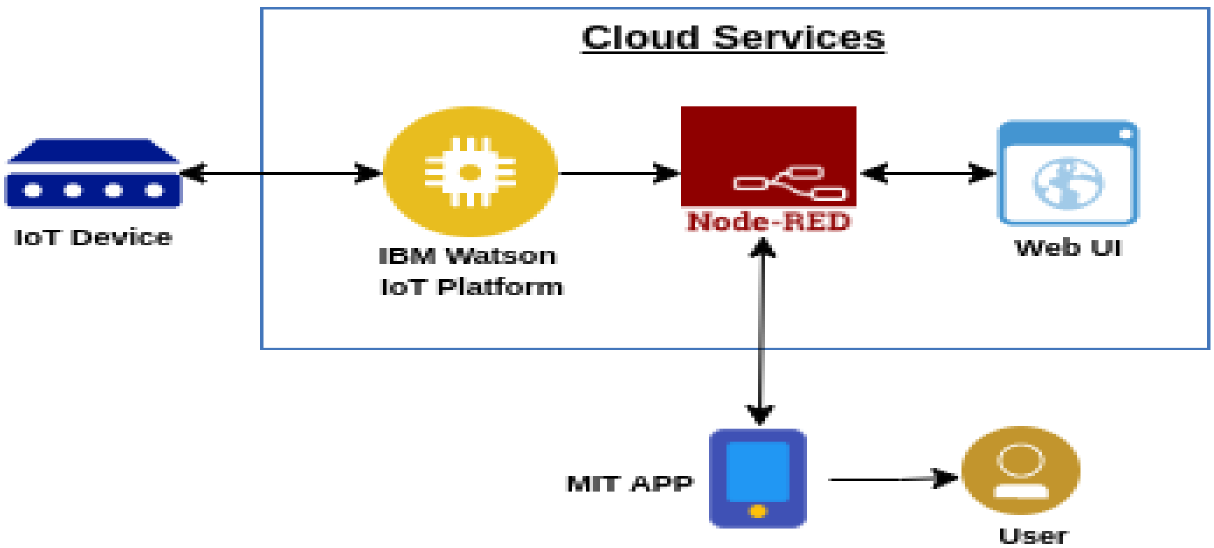
Team members:

- THANOJ K.R– TEAM LEAD
- S.VIVEKANAND
- G.V.VISWA BHARATH
- J.SHAHUL HAMEED

1. ***INTRODUCTION***

1.1 Project Overview:

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, 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:

The smart agriculture model main aim **to avoid water wastage in the irrigation process**. It is low cost and efficient system Is shown below. It includes Node MCU, Arduino Nano, sensors like soil moisture and Dht11, solenoid valves, relays.

2. LITERATURE SURVEY

2.2 Existing problem

The challenges of a smart agriculture system include the integration of these sensors and tying the sensor data to the analytics driving automation and response activities. When integrated, the use of data analytics can reduce the overall cost of agriculture and contribute to higher production from the same amount of area through precise control of water, fertilizer and light. Smart methods allow for farming on smaller and more distributed lands through remote monitoring, whether indoor or outdoor.

To successfully deploy a smart agriculture system, consider setting up a communications network that can integrate a limited number of sensors across a large area of farmland. This will require third-party network provisioning or setting up a private network consisting of access points and uplinks to a private backhaul network, which channels all the data traffic to centralized monitoring software or an analytics head-end system

- It is not a secure system.

- There is no motion detection for protection of agriculture field.

- Automation is not available.

2.2 References

[1] ISSN No:-2456-2165 Volume 4, Issue 2 Feb – 2019: "Solars' Energy: - A safe and reliable, eco-friendly and sustainable Clean Energy Option for Future India: - A Review."

[2] Universal Paper of advanced science and science and exploration technology. [2] GRD Journals- Global Research and Development Journal for Engineering | Volume 4 | Issue 3 | February (2019) ISSN: 2455-5703 “Design and Implementation of an Advanced Security System for Farm Protection from Wild Animals”.

[3] International Journal of Innovations in Engineering and Science, Impact Factor Value 4.046 e-ISSN: 2456-3463 Vol.4, No. 5, 2019 “Solar Powered Smart Fencing System for Agriculture Protection using GSM & Wireless Camera”.

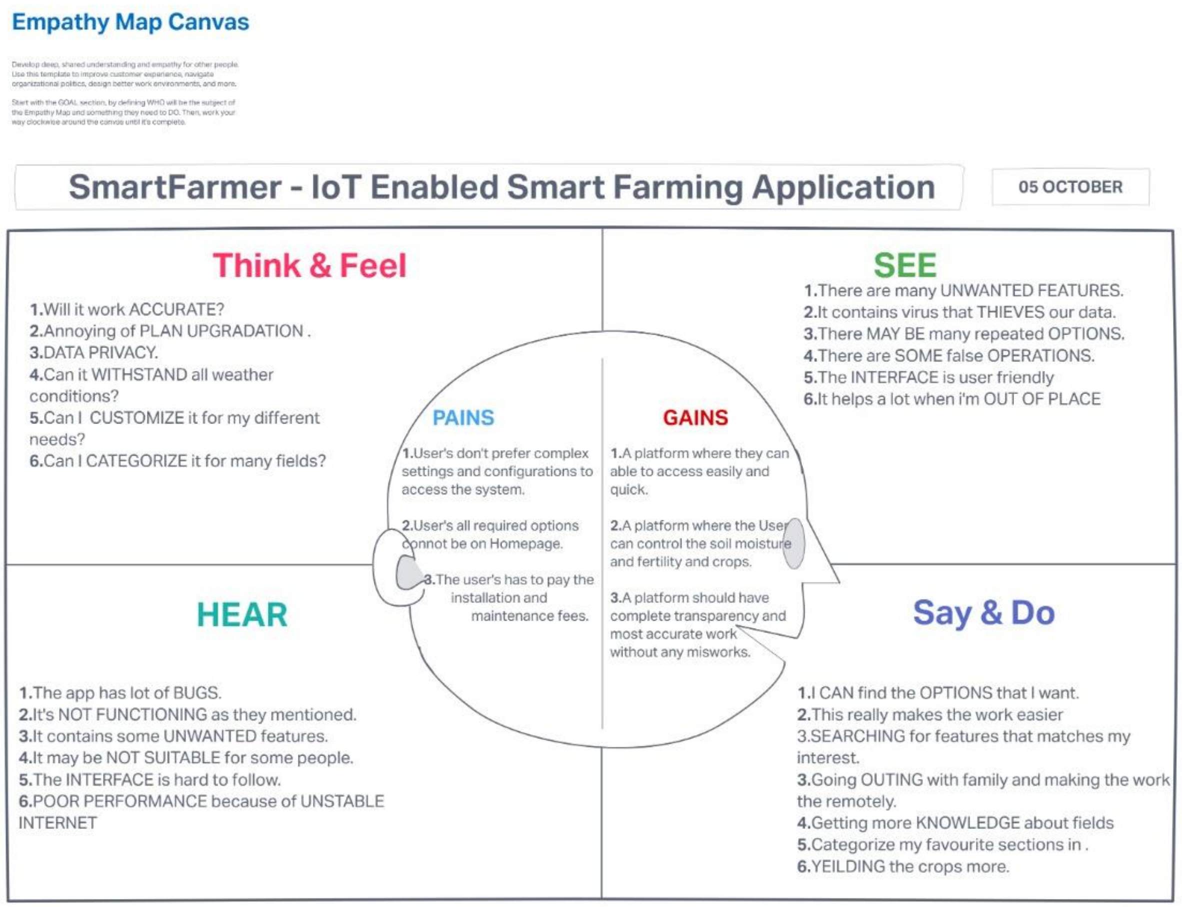
[4] International Journal of Management, Technology And Engineering ISSN NO : 2249-7455 Volume 8, Issue VII, JULY/2018”Protecting Crops From Birds, Using Sound Technology In Agriculture” [5] American Journal of Engineering Research (AJER)2018 eISSN: 2320-0847 p-ISSN : 2320- 0936 Volume-7, Issue-7, pp-326-330 “Moisture Sensing Automatic Plant Watering System Using Arduino Uno”.

2.3 Problem Statement Definition

The soil moisture sensor measures wetness content in the soil. The Arduino UNO microcontroller used to receive input from a various sensor and it can be controlled automatically. When soil moisture sensor goes low the water pump will be on and it exceeds defined levels of the water motor will turn off automatically. We can constantly monitor the growth of a crop using ultrasonic sensor. PIR sensor detects the motion or unusual movement in the agricultural land. This device his very helpful to the former to monitor and control environmental parameters at their field. The farmers did not go to their field, they can remotely monitor and control using cloud.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2Ideation and Brainstorming

Introduction on Internet of Things (IoT), application of IoT in agricultural field to improve the yield and quality by reducing the cost is provided. The sensors which are used in the architecture are discussed briefly and the process of transmission of data from the agriculture field to the central system is explained. The proposed system advantages are included. In addition, open research issues, challenges, and future of IoT in agricultural field are highlighted. The concept is basically developed on an idea, where there are numerous things or objects - such as Arduino, sensors, GSM models, LCD display, etc., that are connected with the Internet. Each of the objects has a different address and is able to interact with other items. The things or objects co-operate with each other to reach a common goal.

We are going to construct a smart agricultural monitoring system which can collect crucial agricultural data and send it to an IoT platform called Thingspeak in real time where the data can be logged and analyzed. The logged data on Thingspeak is in graphical format, a botanist or a reasonably knowledged farmer can analyze the data (from anywhere in the world) to make sensible changes in the supplied resources (to crops) to obtain high quality yield.

Smart agriculture monitoring system or simply smart farming is an emerging technology concept where data from several agricultural fields ranging from small to large scale and its surrounding are collected using smart electronic sensors. The collected data are analyzed by experts and local farmers to draw short term and long-term conclusion on weather pattern, soil fertility, current quality of crops, amount of water that will be required for next week to a month etc.

We can take smart farming a step further by automating several parts of farming, for example smart irrigation and water management. We can apply predictive algorithms on microcontrollers or SoC to calculate the amount of water that will be required today for a particular agriculture field. Say, if there was rain yesterday and the quantity of water required today is going to be less. Similarly, if humidity was high the evaporation of water at upper ground level is going to be less, so water required will be less than normal, thus reducing water usage.

3.3 Proposed Solution

Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	*Many problems are faced by the farmers, some of them are,* *1.Soil erosion and soil nutrient loss,* *2.product satisfaction customers ,* *3.adaptation to climate,* *4.usage of harmful fertilizer and manure and pesticides,* *5.taking agriculture to the next generation,* *6.only fresh water can be used for farming.* *Farmers are more affected and annoyed by the above factors.*
2.	Idea / Solution description	*By providing an IoT integrated platform where all the agricultural devices and agricultural oriented servers are connected , where they can find solutions for the above problems like with climate and loss of soil nutrient by providing adequate knowledge by news feeds and for approximate climate prediction by performing analysis on the day to day weather.*
3.	Novelty / Uniqueness	Building a community of farmers around the globe will definitely do great, where they can find lots and lots of information about crops and plantations directly from the other farmers and the agricultural experts around the globe.

4.	Social Impact / Customer Satisfaction	The good thing is that IoT instructs or alerts them to do the amount of work at the right time. So that the yielding will be more and good, it also reduces the attention and time required to the field, which makes the Customer Satisfaction. On the other hand the main disadvantage is that IoT reduces the number of laborers and their wages, as a result many people may lose their work.
5.	Business Model (Revenue Model)	Many agricultural products like fertilizers, pesticides,manure, and field equipment can also be promoted in the form of ads. A small amount of subscription fee can also be collected from the farmers.
6.	Scalability of the Solution	The scalability of the above proposed solution is not limited. Here a lot of sensors and analysing tools and algorithms can be integrated to provide the best experience .

3.4 Problem solution fit

1.Customer segments:- Types of Customers who are going to this project are • Large Scale Farmers • Remote Farmers	6.**Customer constrains**:- The customer needs a solution which will solve the problems in farming when he is in a remote location and that solution should fulfil the following needs. • Cost efficient • Low power consumption • Time efficient	5. **Available solutions** :- We can give solutions to this problem by using the Smart Farming Application which collects the Moisture level data from the field and operate in the basis of that moisture level.
2. **Jobs to be done** :- The Customers want to automate the irrigation process, reduce cost of manual workers and minimize the power consumption	9. **Problem route cause:-** The route cause for Smart farming Applica	7. **Behavior:-** The customer needs to make a revolutionary change in farming by means of modern technologies.
3. **Triggers:-** Farmers are facing many problems while farming in traditional manner. This triggers the Smart Farming Applications. 4. **Emotions:-** Farmers feel very relaxed and feel stressless while working in field.	10. **Solution:-** Our solution for this project is to give environment sustainable Product for the farming in modern era with reduced cost and with best efficiency	8.**Channels of behaviour**:- The channels of behavior recombines the ration of the following • Online • Offline

4.1 Functional Requirement:

Functional Requirements:

Following are the functional requirements of the proposed solution.

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 System Admins Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details Humidity details
FR-6	Log out	Exit

4. PRODUCT DESIGN

4.1 Data flow diagrams

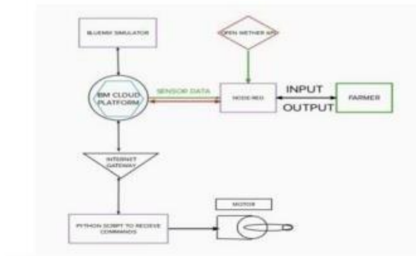
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

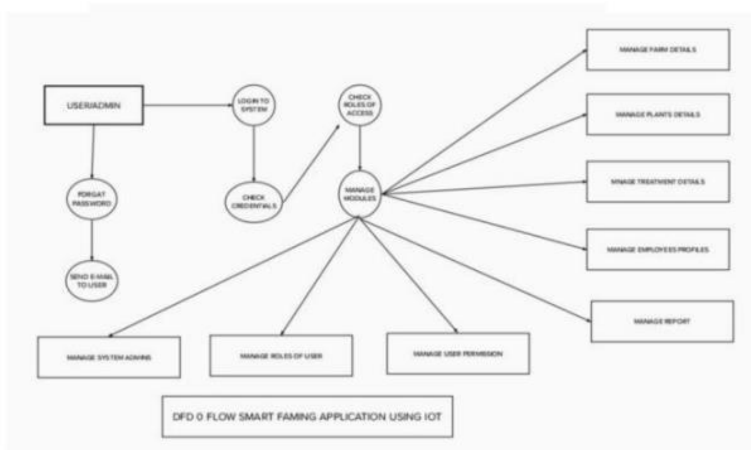
• The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.

• Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.

• NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.

• All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could plan through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.



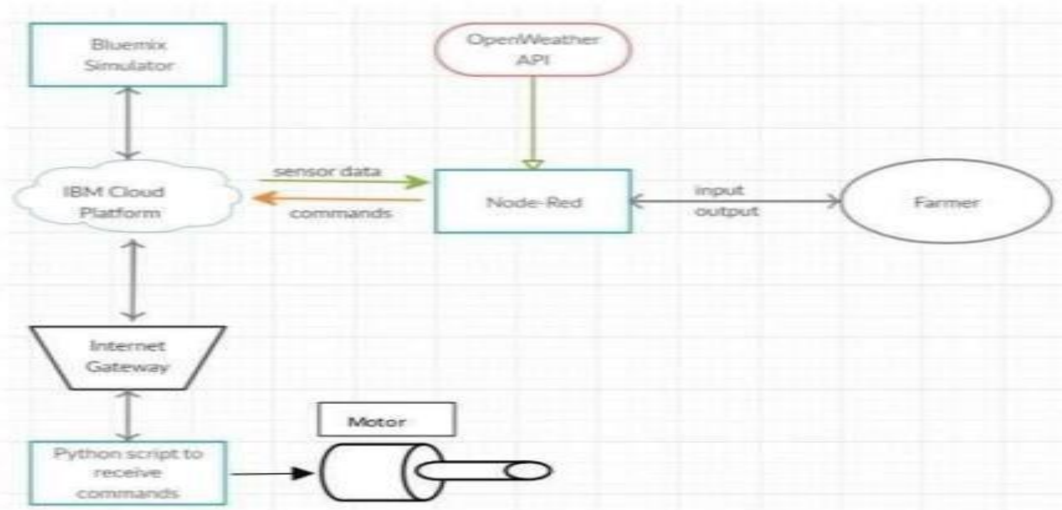


4.2 <u>Solution and Technical Architecture:</u>

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2 Guidelines:

- The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.
- Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.
- NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.
- All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could decide through an app, weather to

water the crop or not depending upon the sensor values. By using the app, they can remotely operate the motor switch.



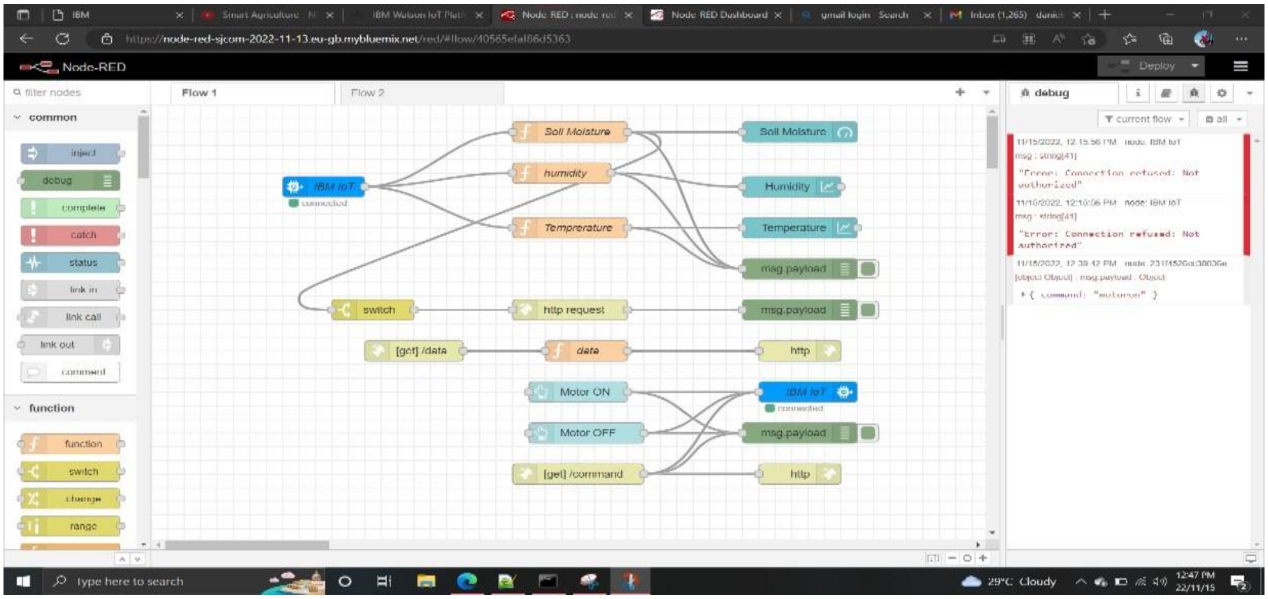
5. PROJECT PLANNING AND SCHEDULING

	Functional Requirement (Epic)	User Story Number		Points		Team Members
Sprint-1	Simulation creation	USN-1	Connect Sensors and Arduino with python code	2	High	Abishek, Kumaran, DanielRemijes, Adhvaidh
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High	Abishek, Kumaran, DanielRemijes, Adhvaidh
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using MIT App Inventor	2	High	Abishek, Kumaran, DanielRemijes, Adhvaidh
Sprint			**User Story / Task**	**Story**	**Priority**	
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Abishek, Kumaran, DanielRemijes, Adhvaidh
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Abishek, Kumaran, DanielRemijes, Adhvaidh

6. CODING AND SOLUTIONING

6.1 Feature

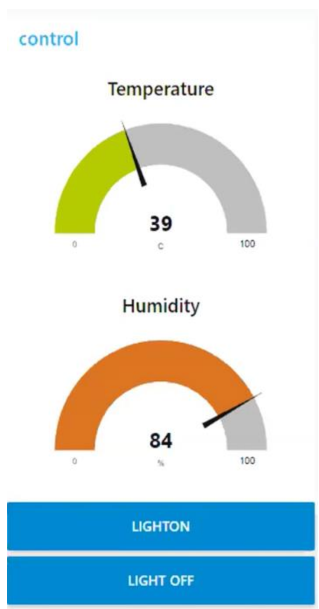
```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig ={
   "identity": {
        "orgId": "0hzydu",
        "typeId": "NodeMCU",
        "deviceId": "12345"
   },
   "auth": {
      "token": "12345678"
   }
}
client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect ()
def myCommandCallback (cmd) :
   print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
   m=cmd.data['command']
   if (m=="motoron"):
      print("Motor is switchedon")
   elif (m=="motoroff"):
      print ("Motor is switchedOFF")
   print (" ")
while True:
   moist =random.randint (0,100)
   temp=random.randint (-20, 125)
   hum=random.randint (0, 100)
   myData={'moisture':moist,'temperature':temp,'humidity':hum}
   client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0 ,
onPublish=None)
   print ("Published data Successfully: %s",myData)
   time.sleep (2)
   client.commandCallback =myCommandCallback
client.disconnect ()
```

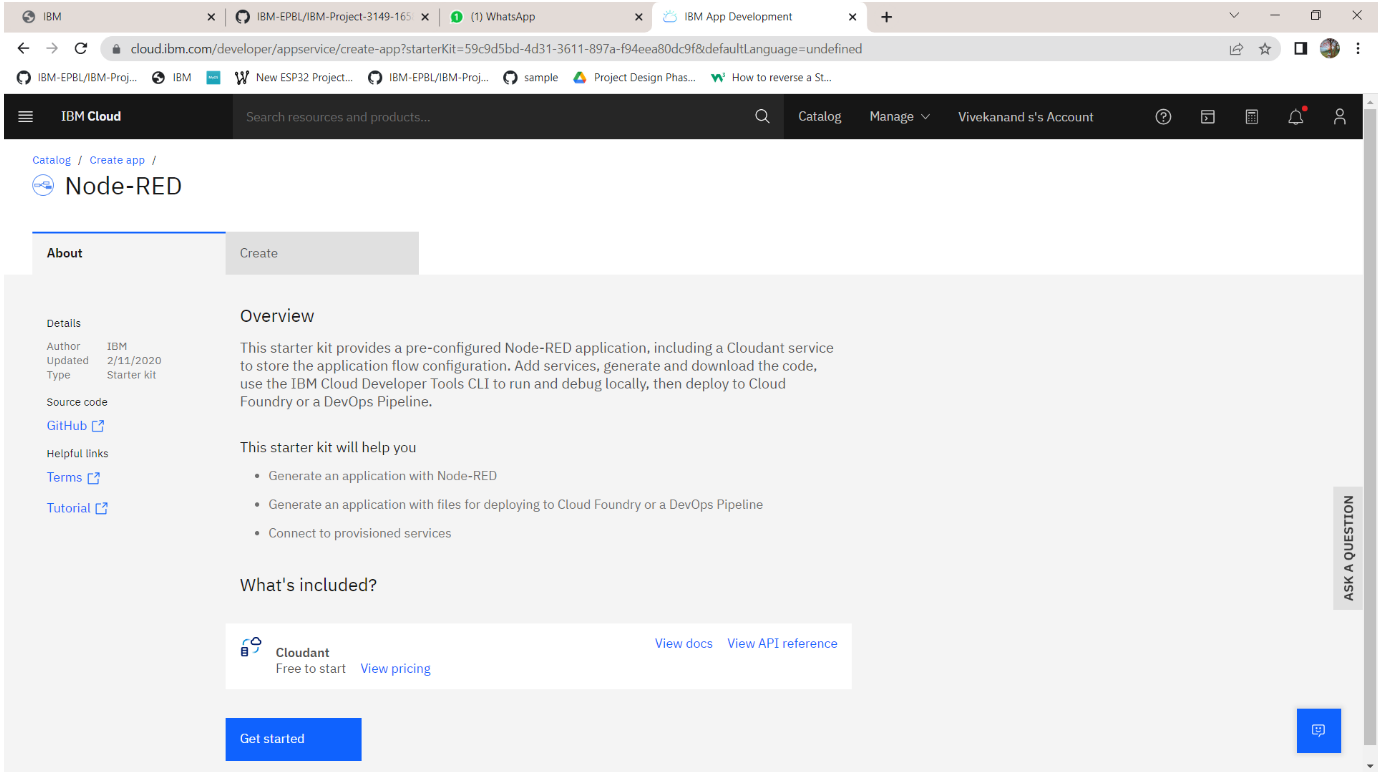


7. TESTING

7.1 Test case:

Web application using Node Red





today classs.py - C:\Users\VIVEK ANAND\Desktop\ibm\today classs.py (3.7.9)

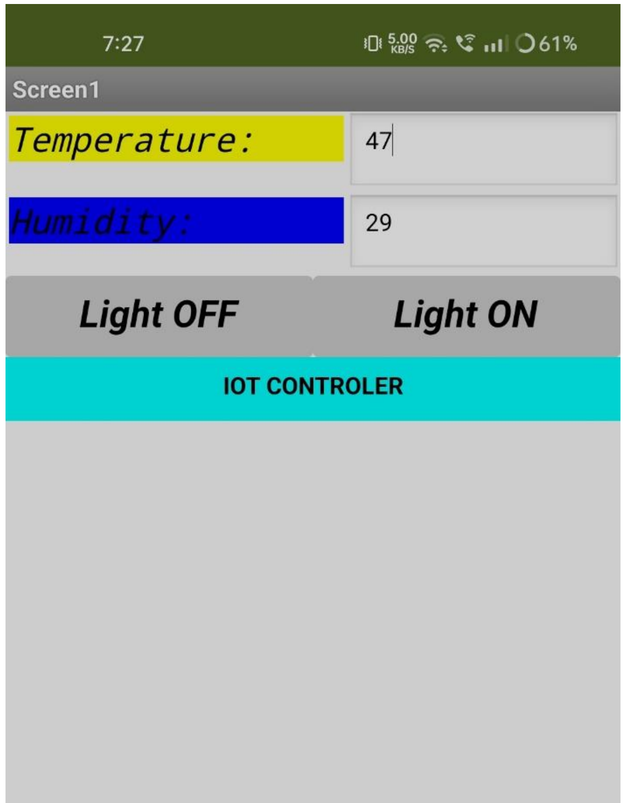
File Edit Format Run Options Window Help

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


#Provide your IBM Watson Device Credentials
organization = "br1jua"
deviceType = "harish123"
deviceId = "123"
authMethod = "token"
authToken = "harish123"

# Initialize GPIO

temp=60
pulse=70
oxygen= 30
lat =  17
lon = 18


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['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()

# 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

        data = {"d":{ 'temp' : temp, 'pulse': pulse ,'oxygen': oxygen,"lat":lat,"lon":lon}}
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % pulse, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
```

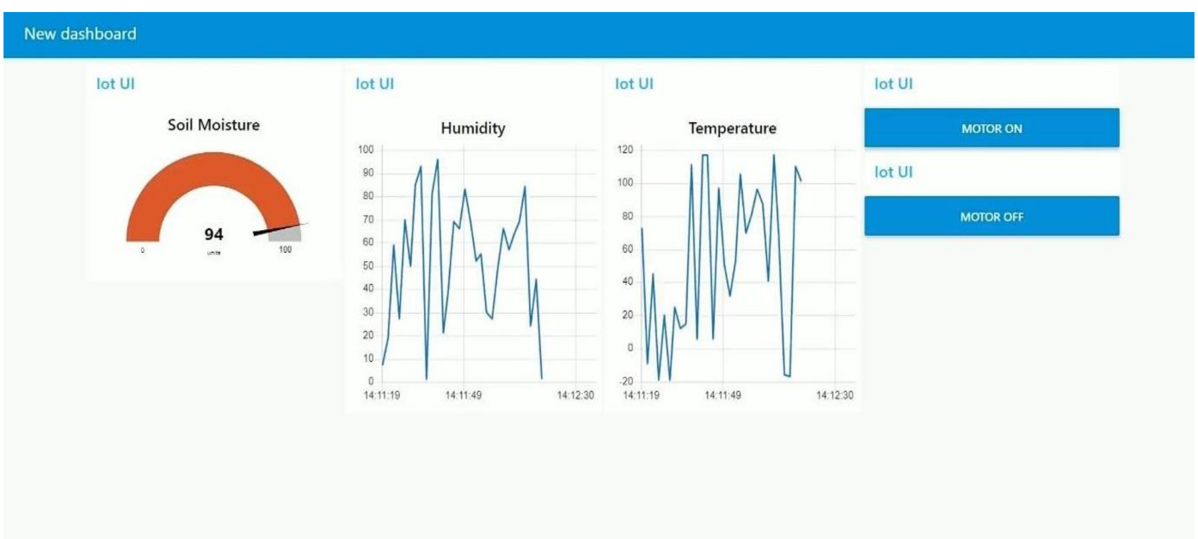
Ln: 1 Col: 0

7.2 User Acceptance Testing

8. RESULTS



8.1 Performance Metrics



9. Advantages and disadvantages

Advantages:

- A remote-control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.
- For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.
- Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.
- Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.

Disadvantages:

- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover, internet connection is slower.
- The smart farming-based equipment require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

10. 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. Continuous internet connectivity is required for continuous monitoring of data from sensors. This also can be overcome by using GSM unit as an alternative of mobile app. By GSM, SMS can be sent to farmers phone.

11. Future scope

In the current project we have implemented the project that can protect and maintain the the crop. In this project the farmer monitors 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 an entire.

• We can update 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 thisproject.

• 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.

12. Appendix

Source Code

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


#Provide your IBM Watson Device Credentials
organization = "br1jua"
deviceType = "viv1"
deviceId = "vivk"
authMethod = "token"
authToken = "vivek300"

def myCommandCallback(cmd):
   print("Command received: %s" % cmd.data['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()

# 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=rand.randint(0,100)
        humid=rand.randint(0,100)

      data = { 'temp' : temp, ‘humid’:’humidity’}
      #print data
      def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % humidity, "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

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

Github link:

https://github.com/IBM-EPBL/IBM-Project-3149-1658503112

Project Demo link:

https://drive.google.com/drive/folders/11iIi1ehtzKjj3JLjIcgnMhBwUTIiFpcM