PROJECT REPORT

done by

SUDHARSAN M

SMART IRRIGATION SYSTEM BASED ON IOT

as a part of

RSIP-2020 (July 15th,2020 - August 12th,2020)

conducted by

1. INTRODUCTION

1.1 Overview:

Agriculture is the backbone of our country. Many primitive methods are still in use across many parts of the country. As demand for agriculture increases, at the same time use of upcoming technologies also increasing. This increasing demand on both sides is lead to develop of systems that are really helpful to the farmers in agricultutal lands.

1.2 Purpose:

There are various purpose involved in making Smart systems. As far as this project is concern, **Smart Irrigation system is deployed in order to make things automatic at remote access.** One cannot sit adn moniter 24 hrs in the field as we are in 21st Century, we involve in many other working fileds too. And also this will make the users to control various lands to access from one single source. Overall **Smart Irrigation system will play a major role in managing and monitoring of agricultural fields.**

2. LITERATURE SURVEY

2.1 Existing problem:

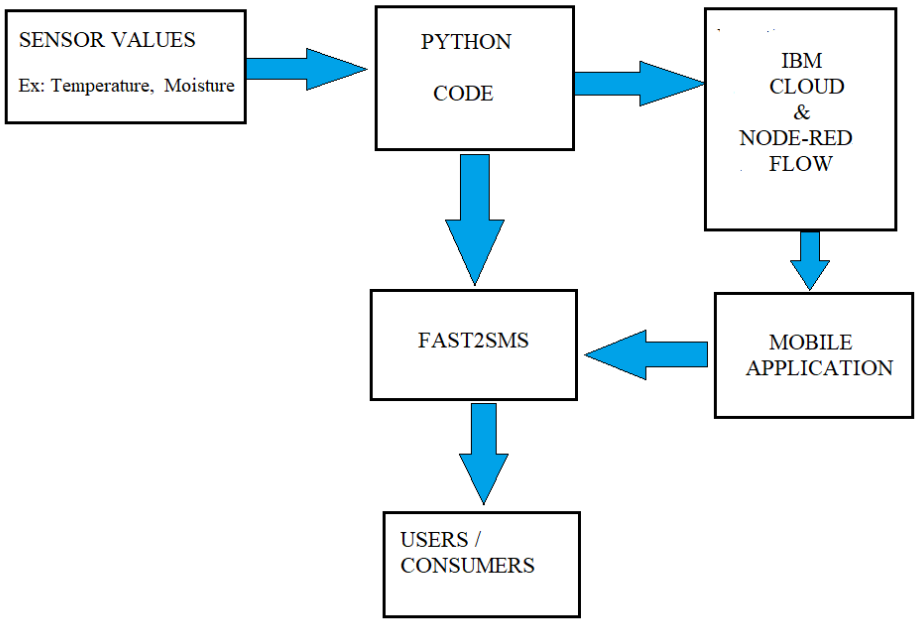
There are many problems associated with the present system. Many rural areas are still deployed with only Manual Irrigation system. Even though many large lands are associated with Smart Irrigation, there also problems still existing. Impacts like **reduced river flow, over usage of water, no proper monitor of crops** etc. And because of these impacts, it even lead to loss of money, more paying of electricity bills etc are also raising. This makes the farmers becoming less in wealth as well as our country's economy on Agriculture will also reduce.

2.2 Proposed Solution:

In this Smart Irrigation system, **Internet of Things (IoT)** which is one the emerging technology is deployed in irrigation of agriculture crops. Using data from OpenWeather, we integrate IBM Node-Red Platform with the MIT App Inventor and Fast2Sms to send alerts to the user about the usage of water in a justified way.

3. THEORITICAL ANALYSIS

3.1 Block Diagram:



3.2 Hardware / Software Designing :

IBM : Acts as a platform to create Node-Red flow. IBM Cloud is used to retrive obtained data from OpenWeather website through Python Code. Over all this acts as an integrated platform for all process involved

NODE-RED: Platform which is availabe within the IBM Platform. It is used to design the flow process by installing and deploying various nodes such as IBM_IoT node, function nodes etc. Dashboard nodes are also availabe to design UI dashboard to view the real time values.

PYTHON(IDLE): Platform in writing the code to get data from OpenWeather website and to send SMS to the user through Fast2Sms website.

FAST2SMS: It helps in sending alert messages to multiple users at one instance of time.

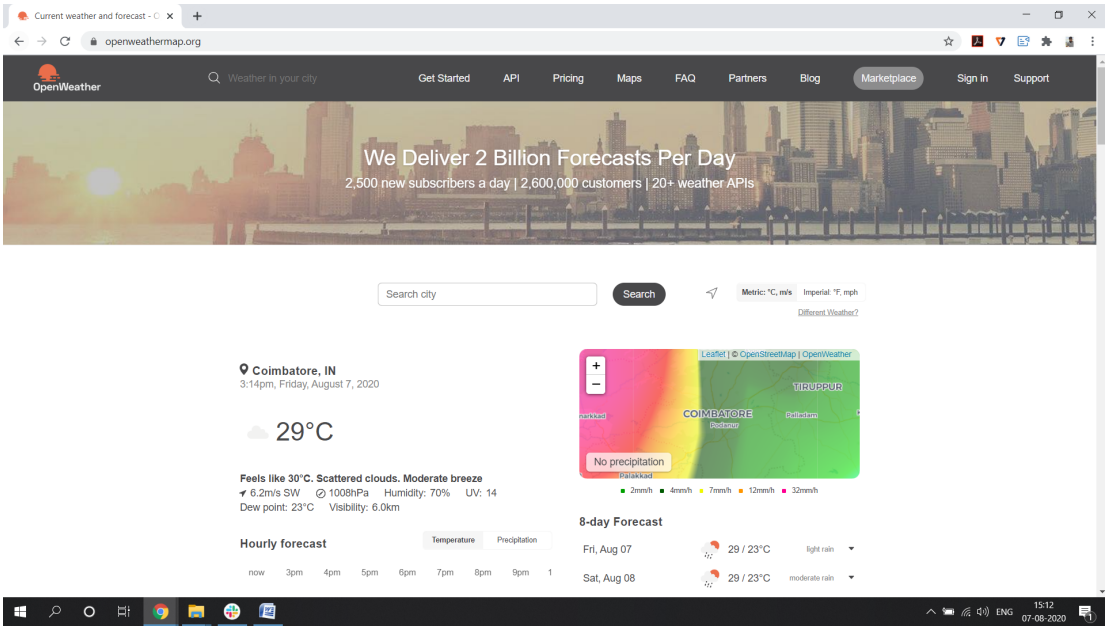
OPEN WEATHER PLATFORM: This platform has the information about weather and in-built API's to use and access the data within it.

MIT APP INVENTOR: This platform is used to build a model APK file using various blocks. The created APK file can be used to get data from IBM Cloud to display in the created Mobile App.

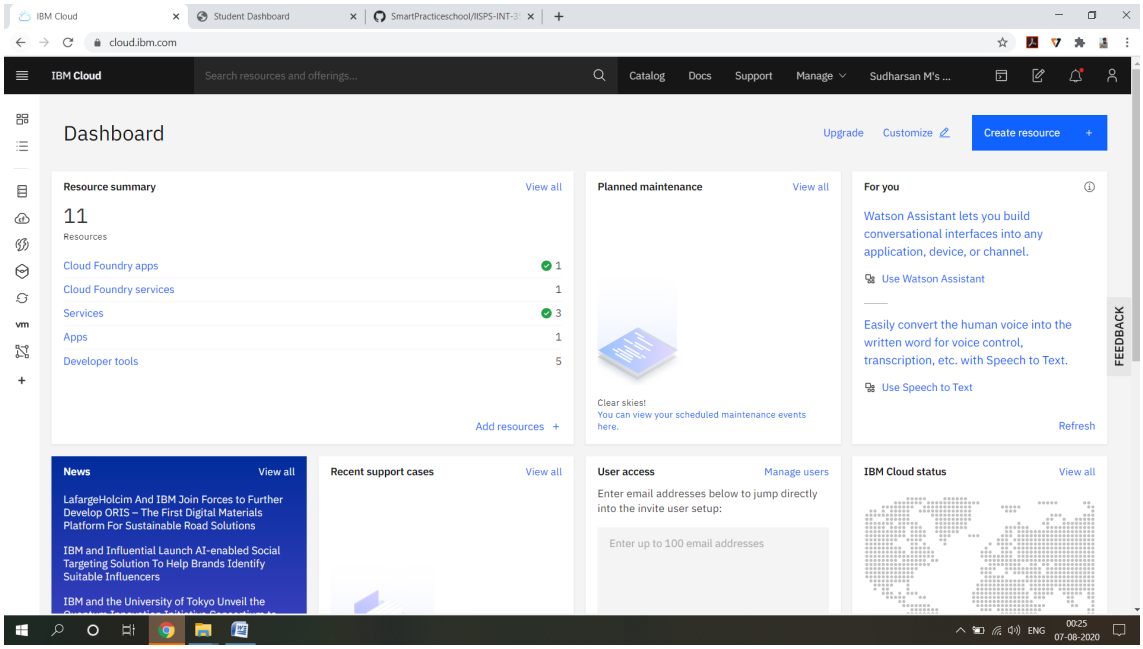
4. EXPERIMENTAL INVESTIGATIONS

IBM IoT platform is used to create several services and appliations. It helps to store the data and retrive it whenever required. Open Weather website will send the real time values of a particualre location to IBM cloud through python code written in IDLE shell. Using Node-Red platform in IBM, various nodes are created. Dashboard nodes are created to design UI interface. IBM IoT nodes are deployed so as to connect to the virtual device. The data created in Node-Red platform will be of .json format. Then model APK file is created in MIT App Inventor. In that using https requests nodes in Node-Red, the date from IBM cloud is retrived and displayed in the created Mobile Application. If the value of moisture content decreases, the Fast2Sms platform will alert the user to turn ON the Motor for irrigation of the crops. The motor can be turned ON using the Mobile application created in MIT App inventor.

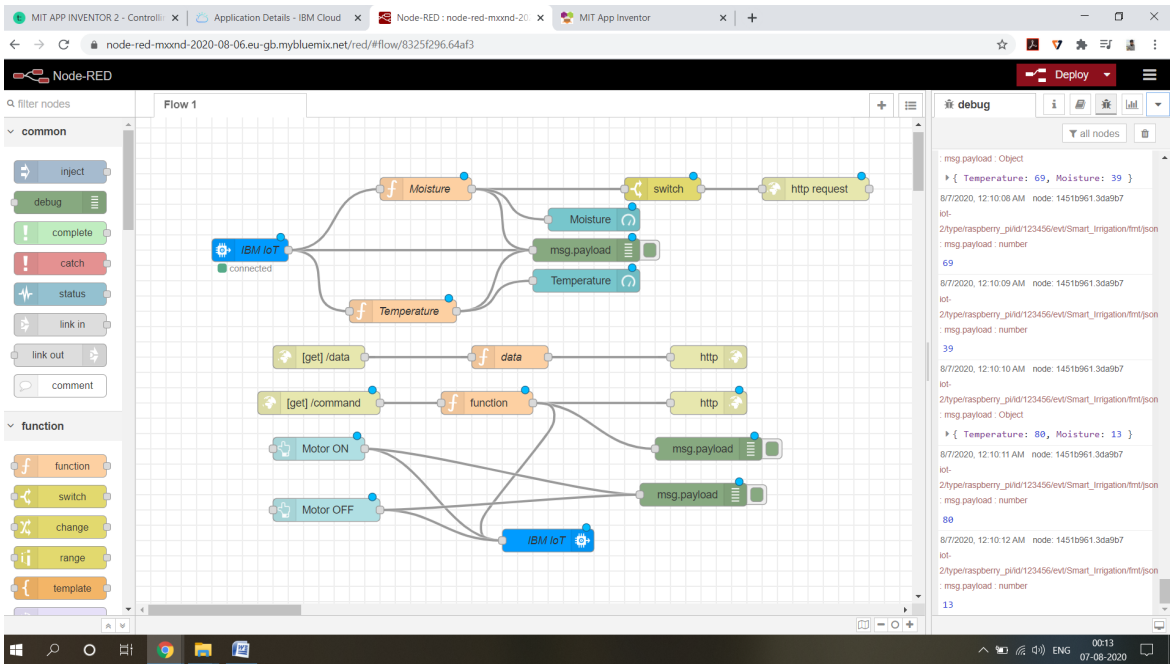
4.1 OPEN WEATHER PLATFORM:



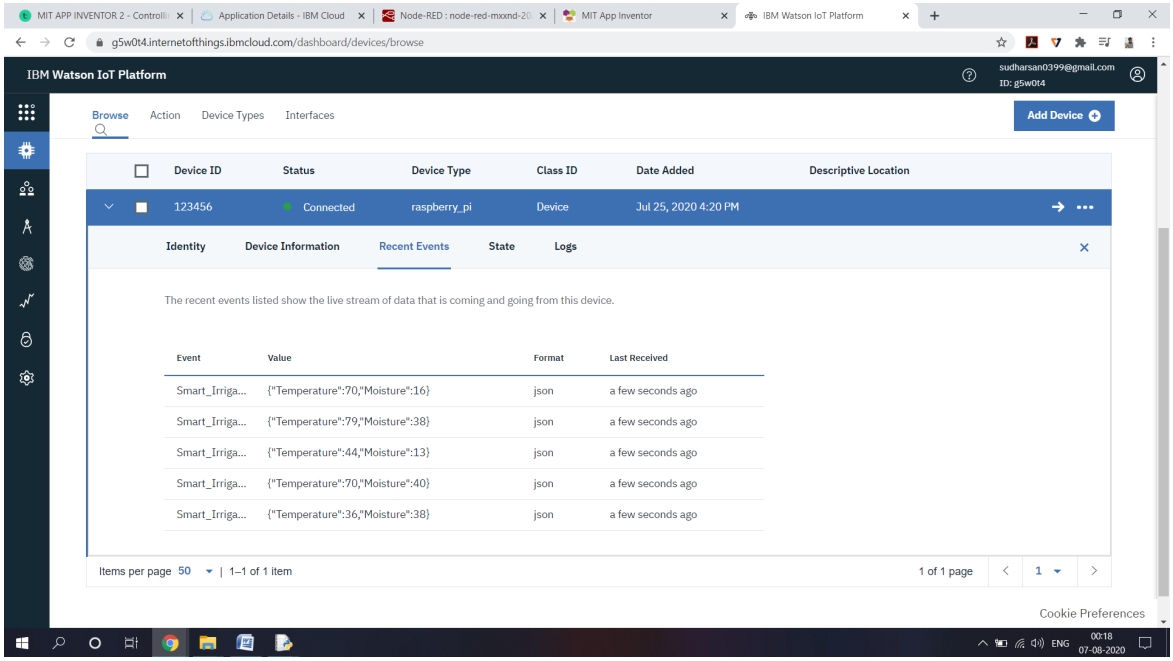
4.2 IBM CLOUD DASHBOARD



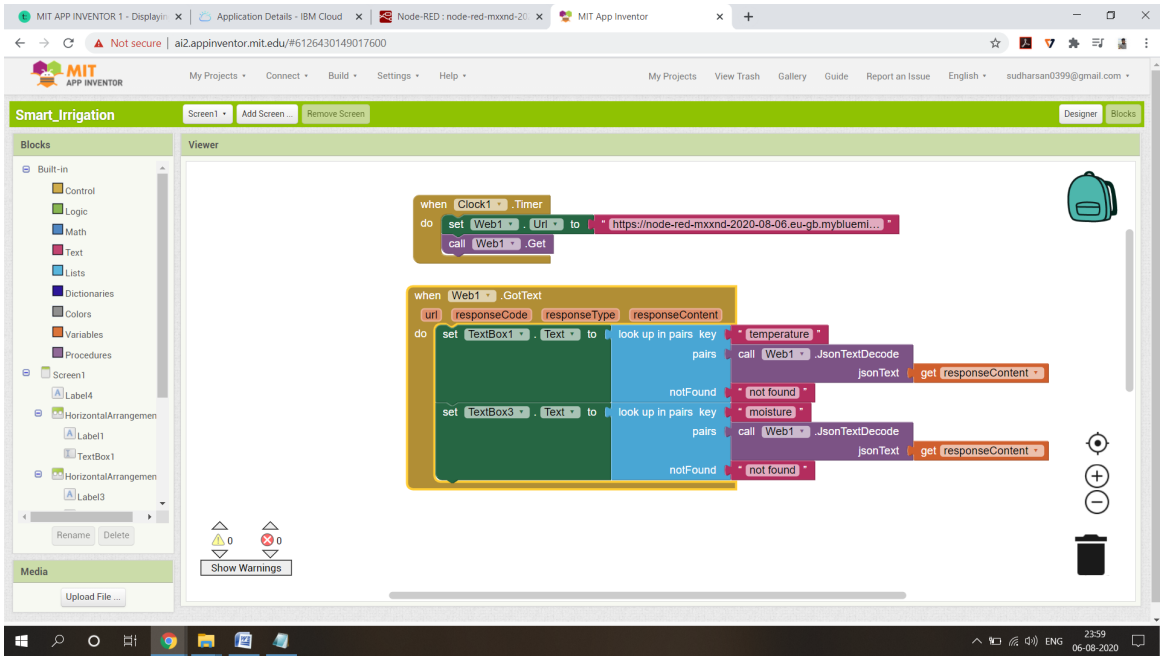
4.3 NODE-RED FLOW:



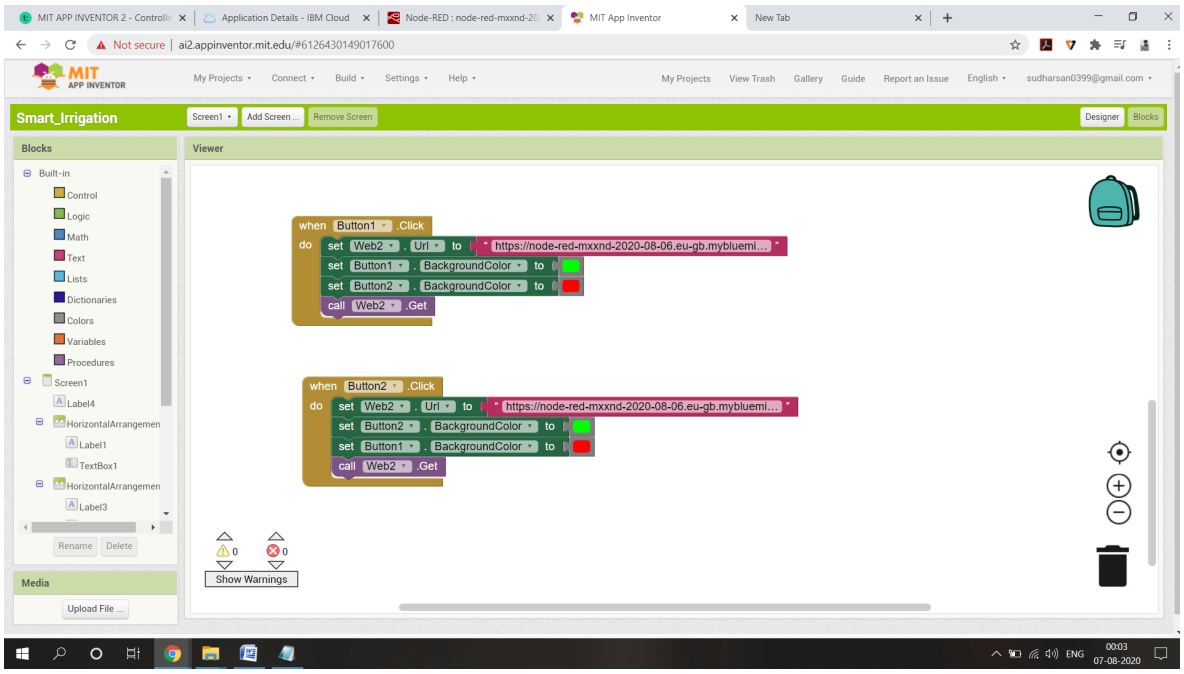
4.4 IBM IOT PLATFORM:



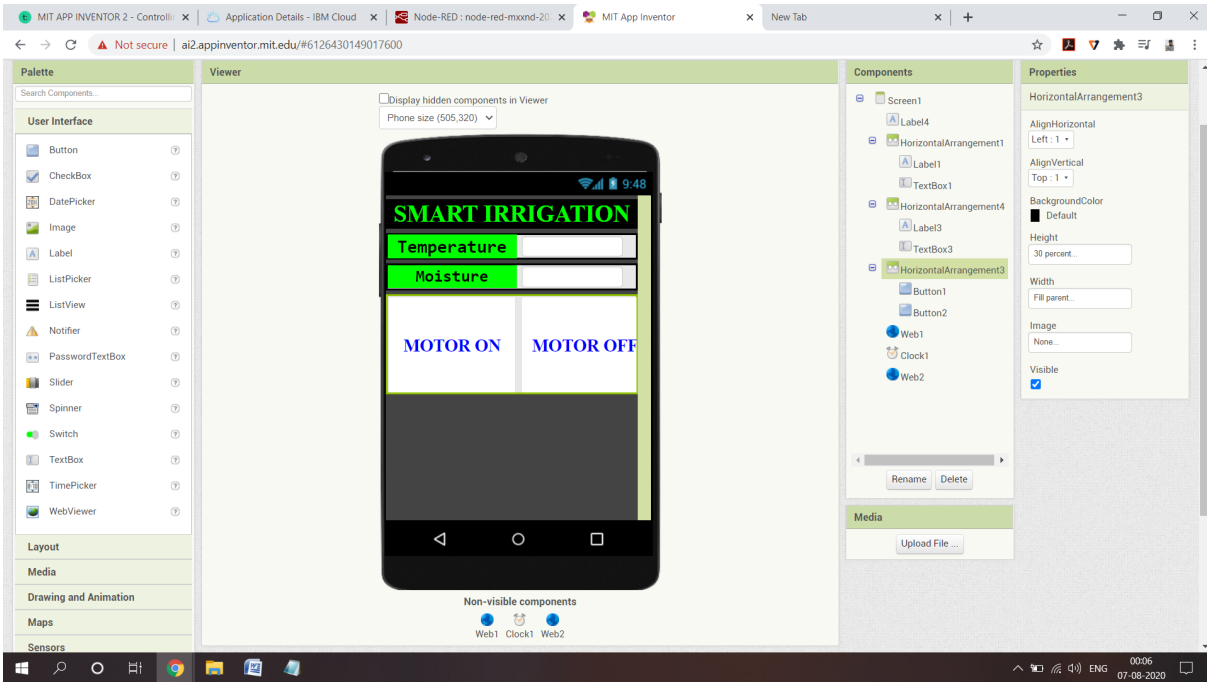
4.5 MIT APP INVENTOR- BLOCKS-1:



4.6 MIT-APP INVENTOR BLOCKS-2:

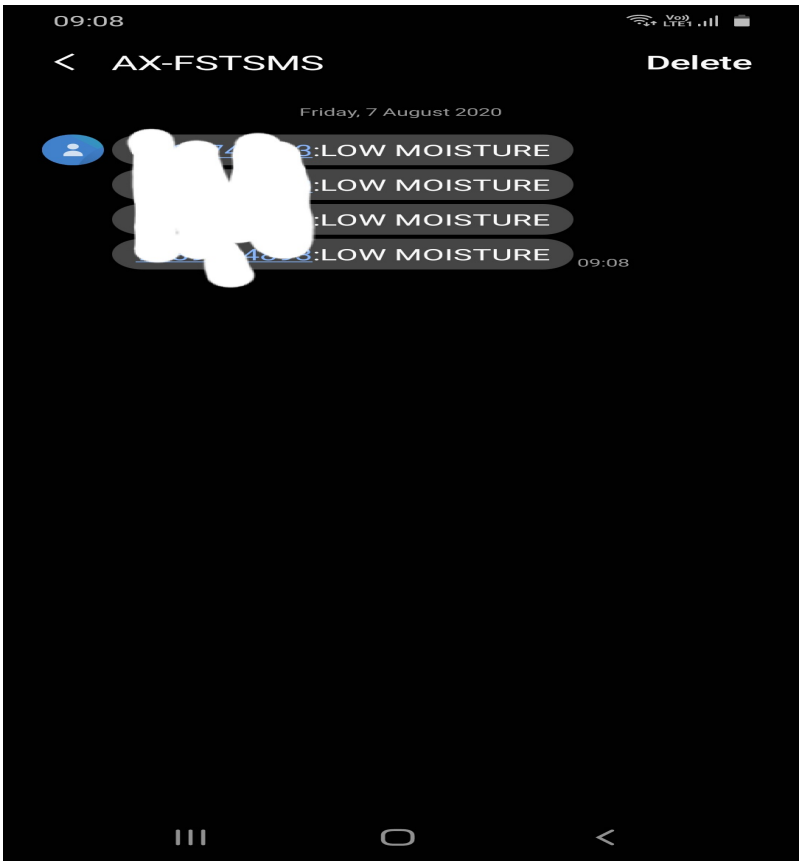


4.7 MIT APP INVENTOR DESIGNER PAGE:

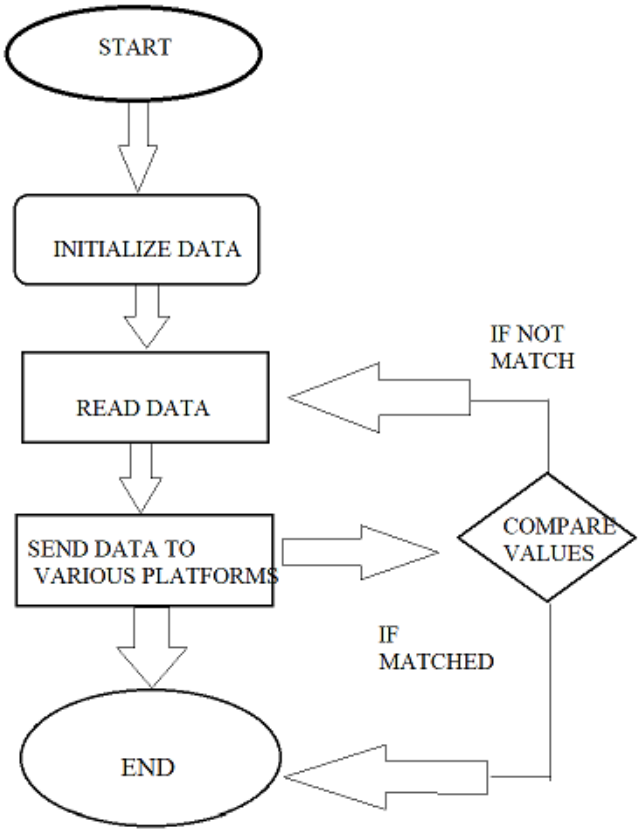


4.8 SMART IRRIGATION APP:

4.9 FAST2SMS ALERT MESSAGE:



5. FLOW CHART



6. RESULT

Hence, using the obtained value from OpenWeather platform, the parameters are assessed and the alerts are sent to the user using the Fast2Sms service to the registered mobile. Thus it will lead to proper irrigation of crops and also save water and used only when needed.

7.1 ADVANTAGES:

- Smart Irrigation system prevents unncessary use of water
- Useful in places where water scarcity is problem
- Users can monitor the weather conditions and increase in economy of production

7.2 DISADVANTAGES:

- The system requires availability of Internet
- Small errors in data/sensor values will influence is production

8. APPLICATIONS:

- Residential Applications such as gardening
- Smart Greenhouse
- Agricultural Fields
- Golfs, turfs.

9. CONCLUSION:

A Smart Irrigation System prevents unnecessary loss of water and provides the information about the current weather conditions. This project alerts the End User whenever it is required and thereby creating a chance for better irrigational facilities and a chance for having better agricultural produce.

10. FUTURE SCOPE:

Better is the Irrigation - better will be the Agriculture. As we are already suffering with severe global warming and scarcity of water, we have a great responsibility of providing a better amount of water resources and irrigational facilities to our younger generations. Smart Irrigation System directly or indirectly plays a prominent role in doing so.

11. BIBILOGRAPHY:

- https://openweathermap.org/
- https://www.fast2sms.com/dashboard/sms/bulk
- https://www.ibm.com/in-en/cloud

12. APPENDIX:

A. Source Code:

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

import requests

organization = "g5w0t4"

deviceType = "raspberry_pi"

deviceId = "123456"

authMethod = "token"

authToken = "12345678"

def myCommandCallback(cmd):

        print("Command received: %s" % cmd.data)

        print(type(cmd.data))

        i=cmd.data['command']

        if i=='motorON':

                print("Motor is on")

        elif i=='motorOFF':

                print("Motor is off")

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:
        moist=random.randint(10,40)
        temp =random.randint(30,80)
        data = { 'Temperature' : temp, 'Moisture': moist }
        if moist<=50:
r=requests.get('https://www.fast2sms.com/dev/bulk?authorization=PGmSoj
zFqKwhN45ti0D1dpBUuAnr3xRMb7CQagce6f2XYJsZWVXU1SZuCABYlaoNE7x46sczHPgp
Gb0O&sender_id=FSTSMS&message=MOTOT%20ON&language=english&route=p&numb
ers=9789744893')
                print(r.status_code)
        def myOnPublishCallback():
            print ("Published Temperature Value = %s C and " % temp,
"Moisture Content = %s %%" % moist, "to IBM Watson")
        success = deviceCli.publishEvent("Smart_Irrigation", "json",
data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(2)
        deviceCli.commandCallback = myCommandCallback
deviceCli.disconnect()
```

B. UI Output Screenshot