REPORT

ON

SMART IRRIGATION SYSTEM BASED ON IOT

BY- J BHAVANA

PRACHIKA CHAUDHARY

INDEX

INTRODUCTION:

1.1 OVERVIEW

Agriculture plays a crucial in country's development and any country's stand in the world can be determined by this sector. Agriculture plays a vital role in India's economy. Over 58% of the rural area depends on agriculture as their main source of livelihood. Agriculture export constitutes 10% of the country's export. Yet farmers are using traditional techniques for agriculture and most of the farmers still depend on rain for irrigation. The farmer's and even the nation's economy will be ruined if there are no proper yields due to lack of knowledge of the soil nature, timely unavailability of water. Thus the government should take strict steps for better met and profitable irrigation.

It is a smart irrigation system based on IoT(Internet of Things) technology which brought change to each and every field of common man's life by making everything smart and intelligent. Aim of this project is to propose a smart IoT based irrigation system assisting farmers in getting live data about the (weather condition ,temperature, humidity and pressure) for efficient monitoring of the water content in the field by controlling the motor that enables them to provide smart irrigation system and reduces the unnecessary wastage of water in the fields, increase the overall productivity of the crops as adequate amount of water is provided and increase the quality of crops. We will use IBM Watson IoT Platform to create devices and MIT app for IBM Node red. The live data of the environment of field is given by using Open Weather API Key.

1.2 PURPOSE

According to the World Wildlife Fund (WWF), 70% of our planet is covered by the water. However, only 3% of it is fresh water, and only one-third of that is available for consumption. Agriculture consumes more water than that of any other source. Much of the water used in agriculture is wasted due to inefficient irrigation systems. To alleviate this problem, many government impose restrictions on water usage, especially in drought-stricken areas.

To improve irrigation efficient and properly enforce water usage restriction we use Smart Irrigation system. This project enables smart monitoring that helps the government even a state, so that necessary precautionary steps can be taken to make such a lands fertile. Besides, the project is also very much useful for the farmers, organizations or individuals running plant nurseries to automatically the pumping Motor ON and OFF on sensing the need of water to the soil. The advantages of using this method is to reduce is to reduce human intervention and still ensure proper irrigation.

LITERATURE SURVEY:

2.1 EXISTING PROBLEMS

India's economy is highly dependent on agriculture secure especially in rural areas, almost 58% of rural population is dependent on agriculture as their primary source of income. Due to the increase in population they have to increase their crop production rate but with traditional method that face so many problems to meet the needs of the country.
One of the major factor which causes decrease in agriculture yield is climate change.
Agriculture is not only sensitive to climate change but also one of the major drivers for climate change. Understanding the weather changes over a period and adjusting the management practices towards achieving better harvest are challenges to the growth of agriculture sector as a whole.
Other problem's faced by the farmers are:

- Many trips have to be taken in order to check the soil humidity on a regular basis manually.
- It can be difficult to know the exact amount of water to give plants, thus causing stress for the crops by over or under watering.
- Overwatering leads to soil erosion.
- It is sometimes difficult to know the optimal time to plant without data.
- Manually measuring key data points about crops is often difficult , time-consuming, and more0likely to be inaccurate.
- Overwatering crops could lead to higher water costs than what is really needed.

2.2 SOLUTIONS TO THE PROBLEM:

To overcome the problem of traditional farming method, we proposed a solution by building a smart irrigation system that works with IoT (Internet of Things) and cloud computing.

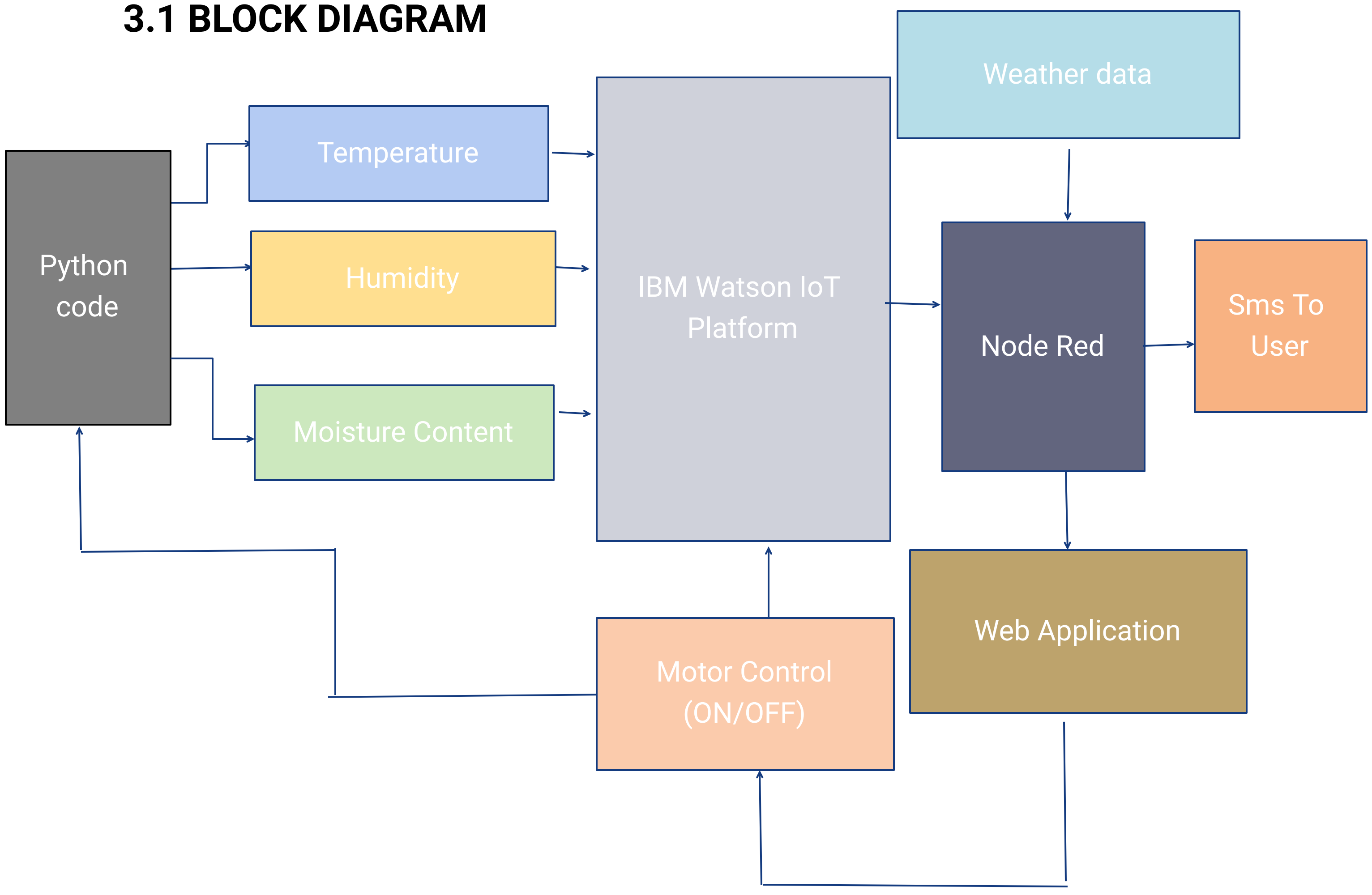
1) IoT:
The Internet of Things(IoT) is a system of interrelated computing devices, mechanical and digital machines provided with unique keys and ids and the ability to transfer data without requiring human-to-human interaction. In this project we integrate this technology with irrigation to make better agriculture process and solve the problems.

2) Cloud computing:
Cloud computing is the on-demand availability of computer system resources, especially data storage(cloud storage) and computing power, without direct active management by the user. With the help of cloud computing technology, we can store the data collected by the sensors and also weather conditions in that particular area and use it for future analysis. These data are being used to provide predictive insights in farming operations, drive real-time operational decisions, and redesign business processes.

This project is based on the idea to develop such a device which can monitor temperature of the surrounding, soil moisture and the temperature of the soil. This information can be generated randomly using python code as we do not have components right now. This data is sent to the cloud and we can get weather conditions details from the Open Weather API. Farmer can monitor the temperature, humidity and moisture content along with the weather forecast details through the web page application. Based on all parameters farmer can water the crop through web page user interface.

THEORITICAL ANALYSIS:

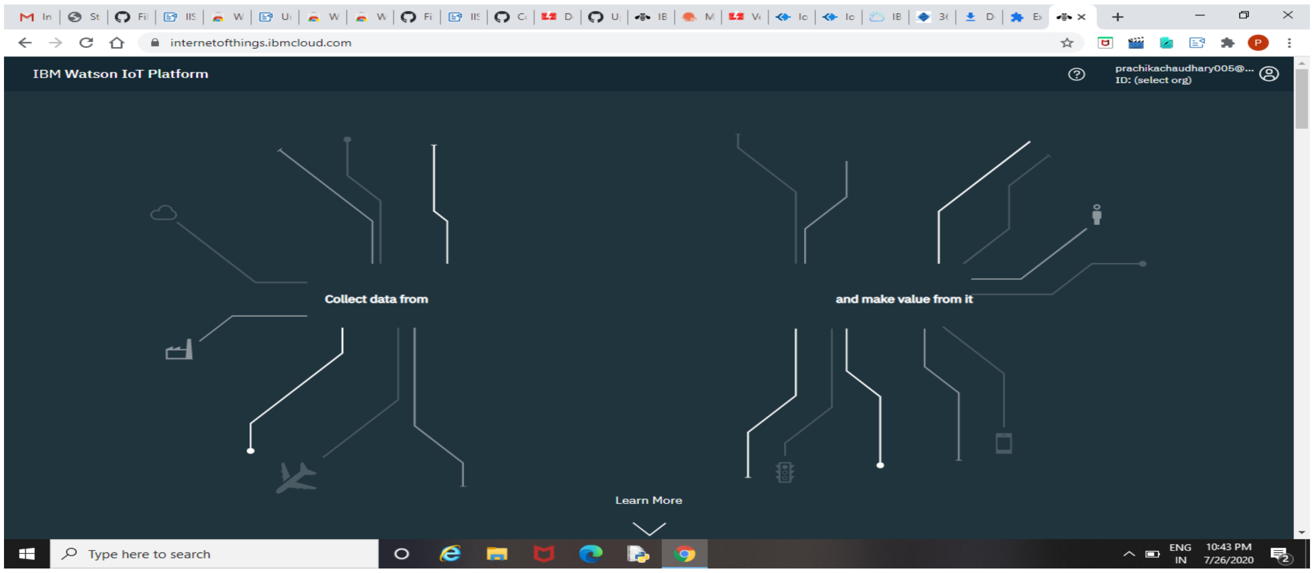
3.1 BLOCK DIAGRAM



3.2 SOFTWARE DESIGNING

- **IBM Watson IoT Platform**

Watson was created as a question answering (QA) computing system that IBM built to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering.
One important tool of IBM Watson is IBM Watson IoT Platform .
IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices, Watson IoT Platform and its additional add on service- enable organisations to capture and explore data for devices, equipment, machine discover insights that can drive better decision-making.
Start by register a device with Watson IoT Platform. Registering a device involves classifying the device as a device type, giving the device name, and providing device information. Then you provide a connection token or accept a token that is generated by Watson IoT Platform.
After you register your device with Watson IoT Platform, you can use the registration information to connect to device and start receiving device data securely up to the cloud. You can set up and manage your devices using your online dashboard or our secure APIs, so that your apps can access and use your live and historical data.
Before you can begin receiving data form your IoT devices, you must connect them to Watson IoT Platform. Connecting a device to Watson IoT Platform involves registering the device with Watson IoT Platform and then using the registration information top configure the device to connect to Watson IoT Platform.



- **Python IDE**

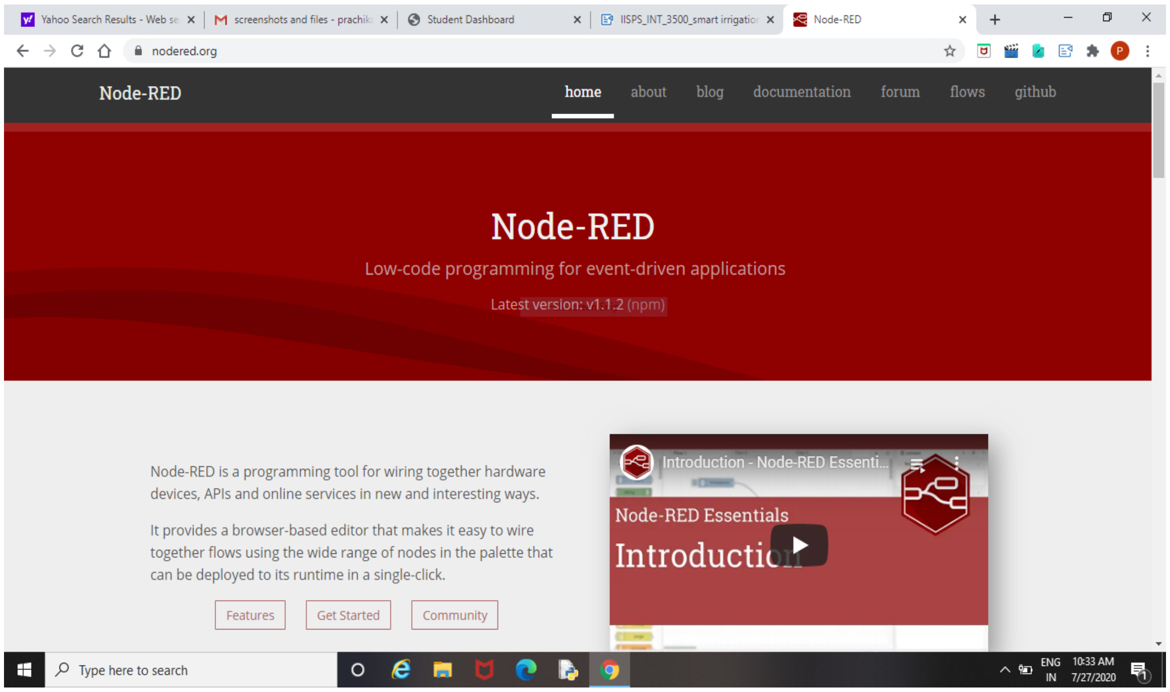
Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant white space. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

In our project we used it to simulate random values of Temperature,humidity,moisture content and receive commands of turning Motor ON/OFF from user interface.

- **Node Red**

Node REDNode-RED is a flow-based programming tool, originally developed source JS Foundation project.
In our project we use node red to create UI for the farmer. The values from the sensor simulator and the weather details fromNode Red is a flow-based programming tool, originally developed by IBM's Emerging Technology Services team and now a part of the JS Foundation.
Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things.
Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the

runtime in a single-click.
Node-RED provides a web browser-based flow editor, which can be used to crate functions. Element of applications can be saved or shared for re-use. The runtime is built on Node.js.
In 2016, IBM contribute Node-Red as an open source JS Foundation project.
In our project we use node red to create UI for the farmer.
 The values from the sensor simulator and the weather details from Open Weather can be viewed in the webpage.
With the help of the data viewed in the webpage, if the farmer feels that there is need to turn the motor on or off, he can do it through webpage application.



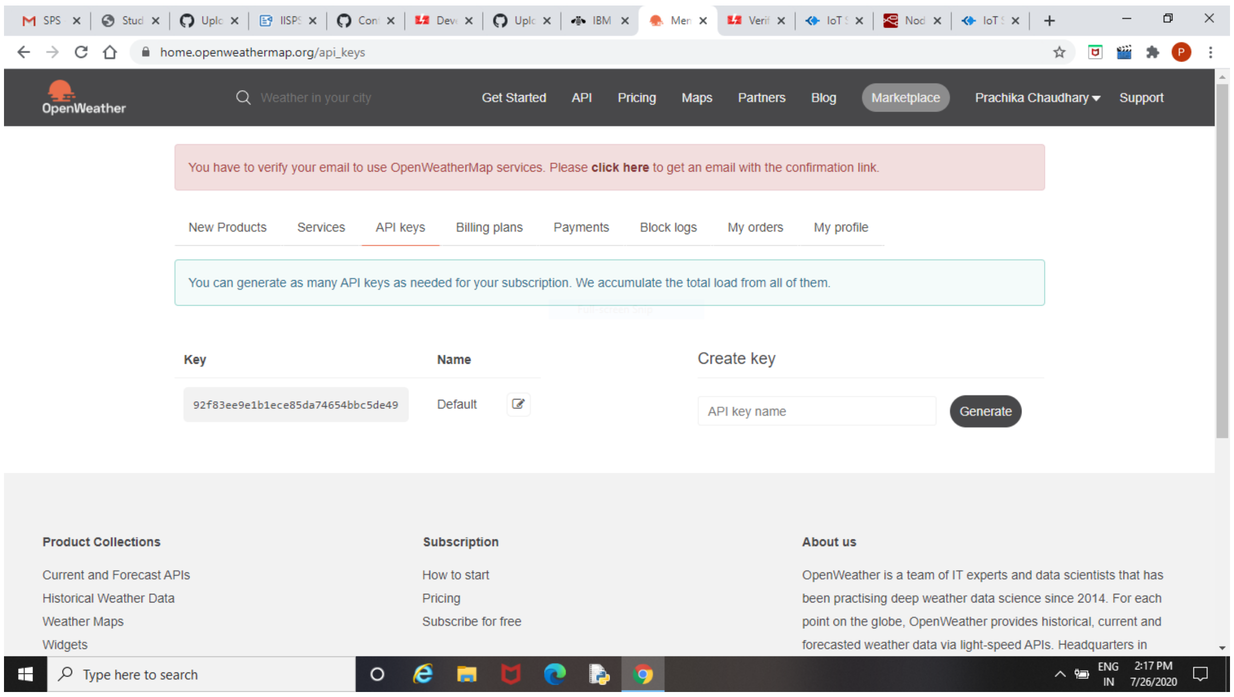
Yahoo Search Results - Web se
screenshots and files - prachi
Student Dashboard
IISPS_INT_3500_smart irrigatio
Node-RED
nodered.org
Node-RED
home
about
blog
documentation
forum
flows
github
Node-RED
Low-code programming for event-driven applications
Latest version: v1.1.2 (npm)
Introduction - Node-RED Essenti...
Node-RED Essentials
Introduction
Features
Get Started
Community
Type here to search
ENG IN
10:33 AM
7/27/2020

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

- **Open Weather API**

Open weather Map is an online server that provides weather data.It is owned by Open Weather Ltd headquartered in London, United Kingdom.
It provide current weather data, forecasts and historical data(starting from 1979) to more than 2 million customers, including Fortune 500 companies and thousands of other businesses globally.
More than twenty weather APIs have been created for getting different types of weather. data. They support multiple languages, units of measurement and data formats. Additionally, the Open Weather Map service allows any users to get basic weather data on the company's website.
With the help of Open Weather API, we can get the details like temperature, humidity, weather condition and even additional details like wind speed, pressure of the particular location.
Using url provided by the website these details can be viewed in web user interface by the farmer.



You have to verify your email to use OpenWeatherMap services. Please **click here** to get an email with the confirmation link.

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name
92f83ee9e1b1ece85da74654bbc5de49	Default

Create key

API key name

Generate

Product Collections

Current and Forecast APIs
Historical Weather Data
Weather Maps
Widgets

Subscription

How to start
Pricing
Subscribe for free

About us

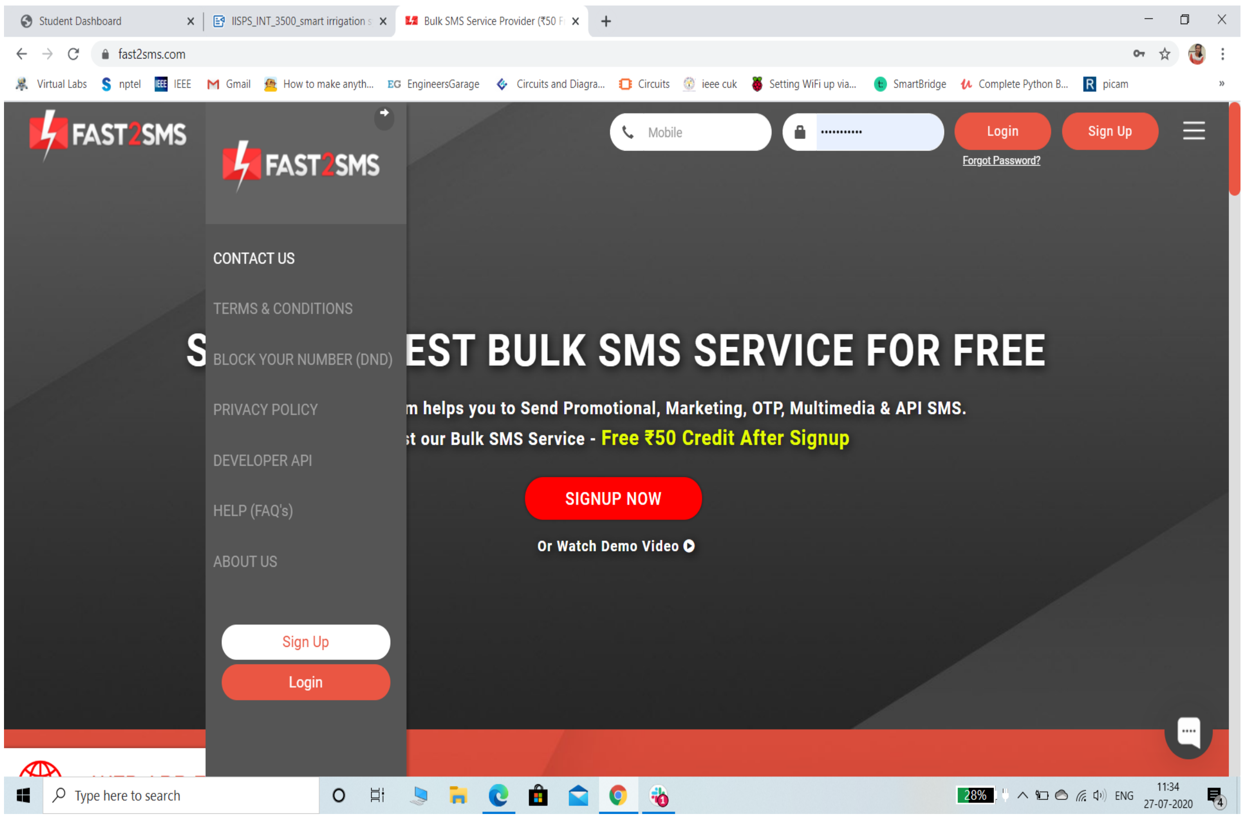
OpenWeather is a team of IT experts and data scientists that has been practising deep weather data science since 2014. For each point on the globe, OpenWeather provides historical, current and forecasted weather data via light-speed APIs. Headquarters in

- **Fast2SMS**

Fast2SMS is a popular bulk sms service provider in India. Famous for its performance driven messaging services, you can expect a high quality SMS services from it.
It is best ever android app available at the Playstore as it offer you the facility of sending Unicode.
SMS as well as in addition to English language.It offers you multi language support in different languages like Hindi, Telugu, Tamil, Urdu, Kannada, Punjabi and many more.
Taking care of the needs of the users, it has launched a unique feature in addition to the 2 prominent features. Mainly there are two types of methods by which you can send SMS.

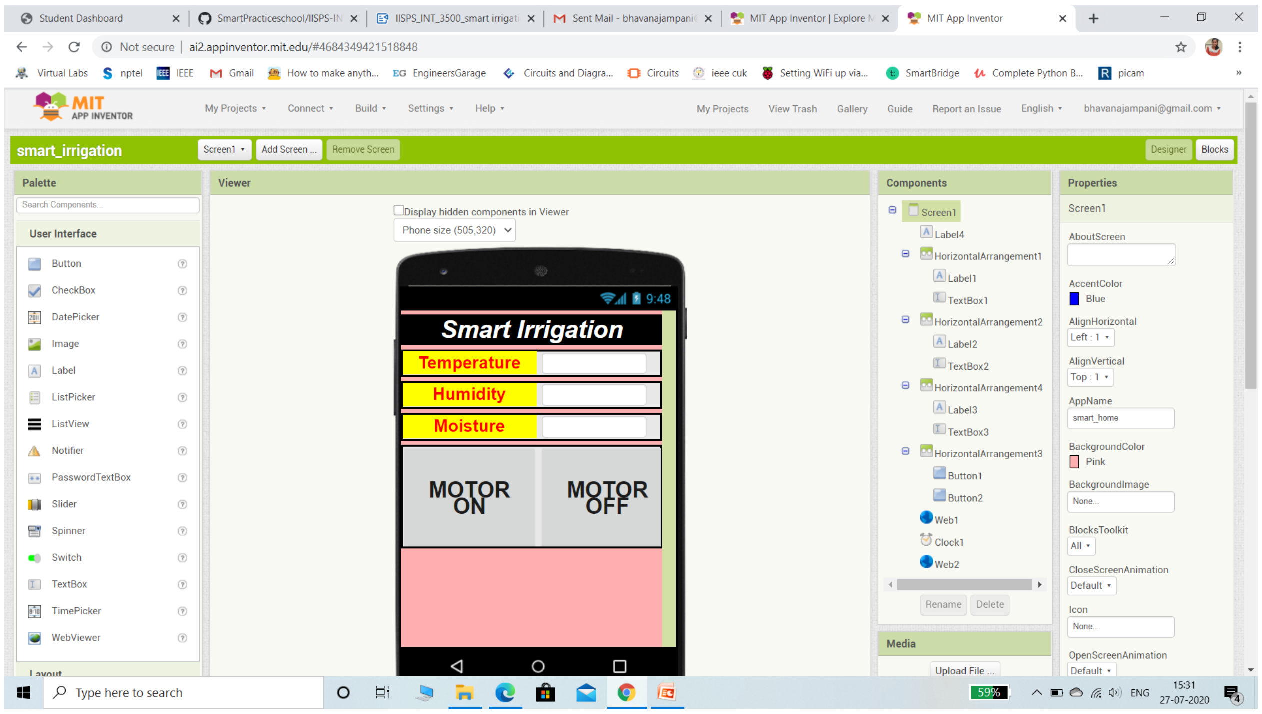
1. Transactional route
2. Promotional route

To send various message alerts to the farmers phone as sms we need Fast2SMS service.



- **MIT App Inventor**

MIT App Inventor is a web application integrated development environment originally provided by google, and now maintained by the Massachusetts Institute of Technology. It was released at 15 December which allows newcomers to computer programming to create application software for two operating system: ANDROID ans IOS, which, as of 8 July 2019, is in final beta testing. It is free and open-source software and it uses a graphical user interface(GUI) .



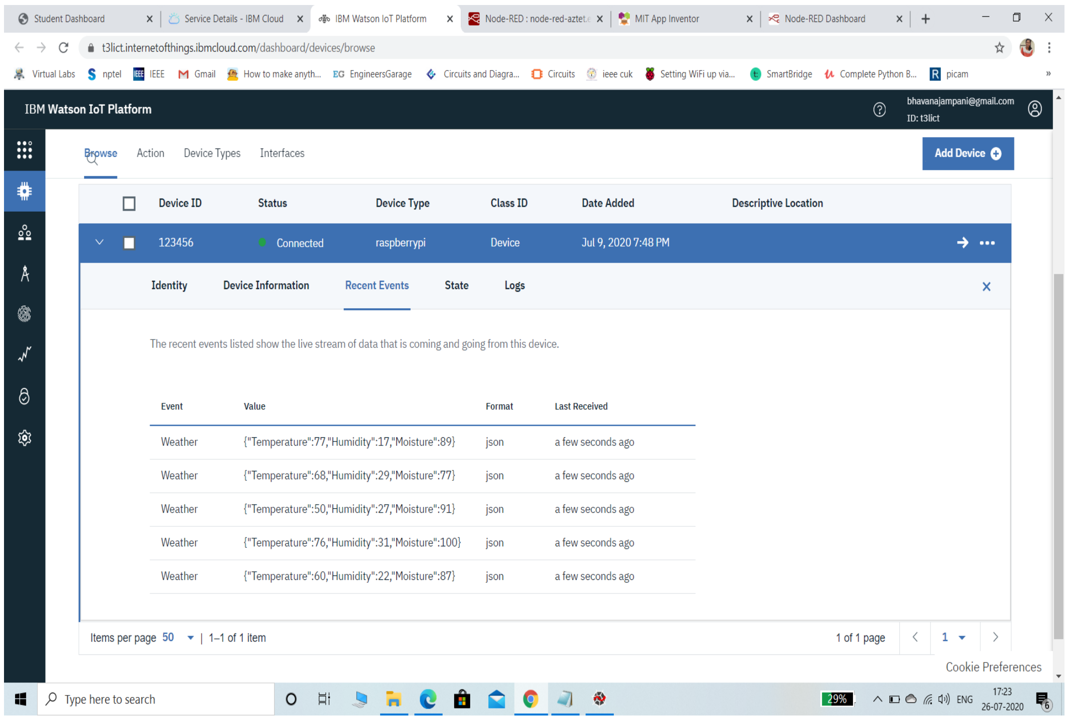
Properties: Screen1
AboutScreen:
AccentColor: Blue
AlignHorizontal: Left : 1
AlignVertical: Top : 1
AppName: smart_home
BackgroundColor: Pink
BackgroundImage: None...
BlocksToolkit: All
CloseScreenAnimation: Default
Icon: None...
OpenScreenAnimation: Default

EXPERIMENTAL INVESTIGATIONS:

The target of this project is to control the motor remotely by detecting the moisture, temperature and humidity and weather conditions of the particular area. The motor is operated using User Interface or a web application.
The procedure involved to achieve this is shown below.

STEP 1: Create a device in IBM Watson IoT Platform and save the details so that can we use it further in project.

STEP 2: To connect to device created in IBM IOT Platform we give user credentials in the code(i.e device id, device type, authentication token etc).After running the code it starts publishing data to the cloud.
We can view the simulated data in cloud platform by selecting the device type and then clicking recent activity.



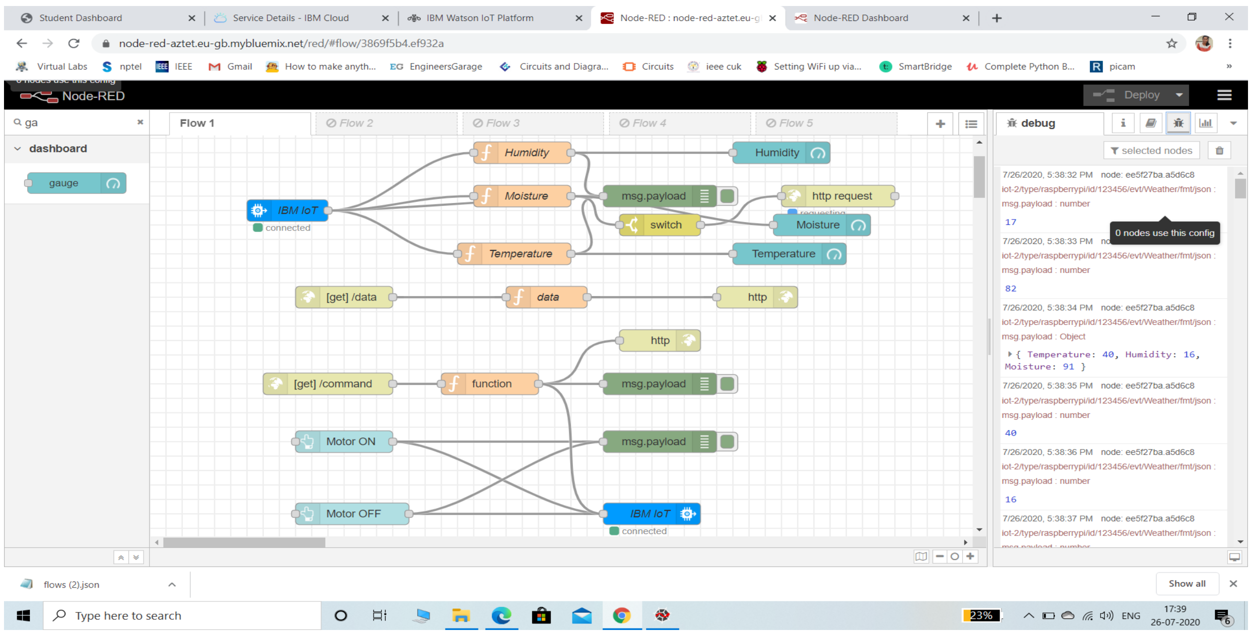
Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456	Connected	raspberrypi	Device	Jul 9, 2020 7:48 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
Weather	{"Temperature":77,"Humidity":17,"Moisture":89}	json	a few seconds ago
Weather	{"Temperature":68,"Humidity":29,"Moisture":77}	json	a few seconds ago
Weather	{"Temperature":50,"Humidity":27,"Moisture":91}	json	a few seconds ago
Weather	{"Temperature":76,"Humidity":31,"Moisture":100}	json	a few seconds ago
Weather	{"Temperature":60,"Humidity":22,"Moisture":87}	json	a few seconds ago

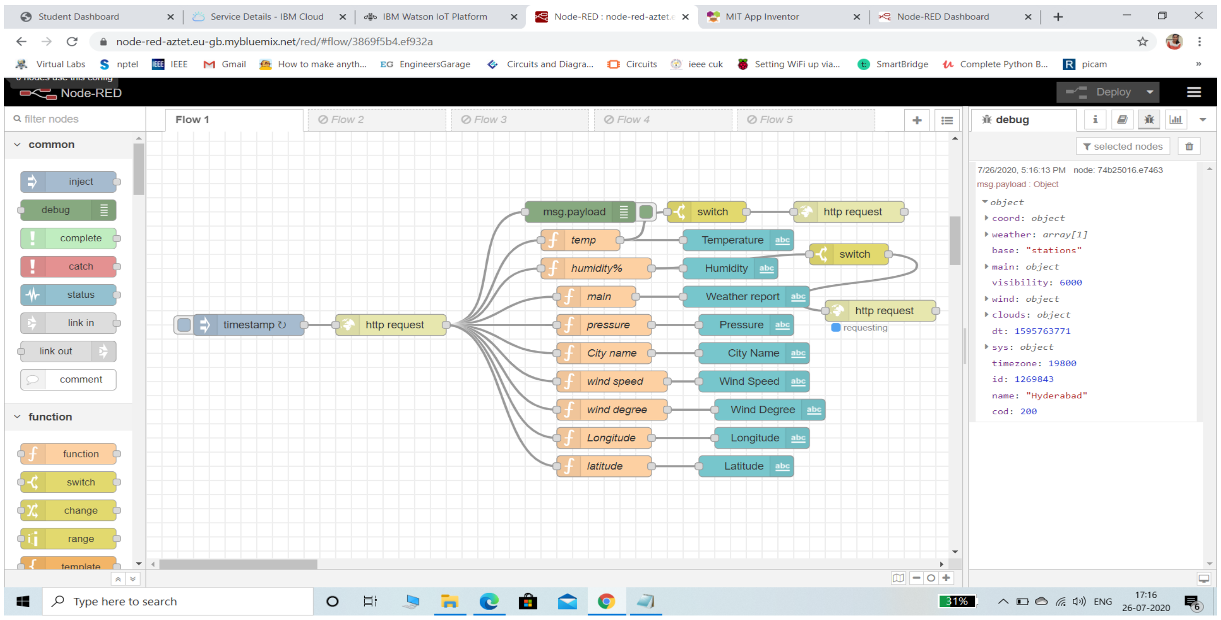
Items per page 50 | 1–1 of 1 item   1 of 1 page

STEP 3: Configure the Node red and install the required nodes to retrieve the data from IBM IoT Platform and send commands to device.
Even to send sms to user when moisture content is below threshold value via http request.



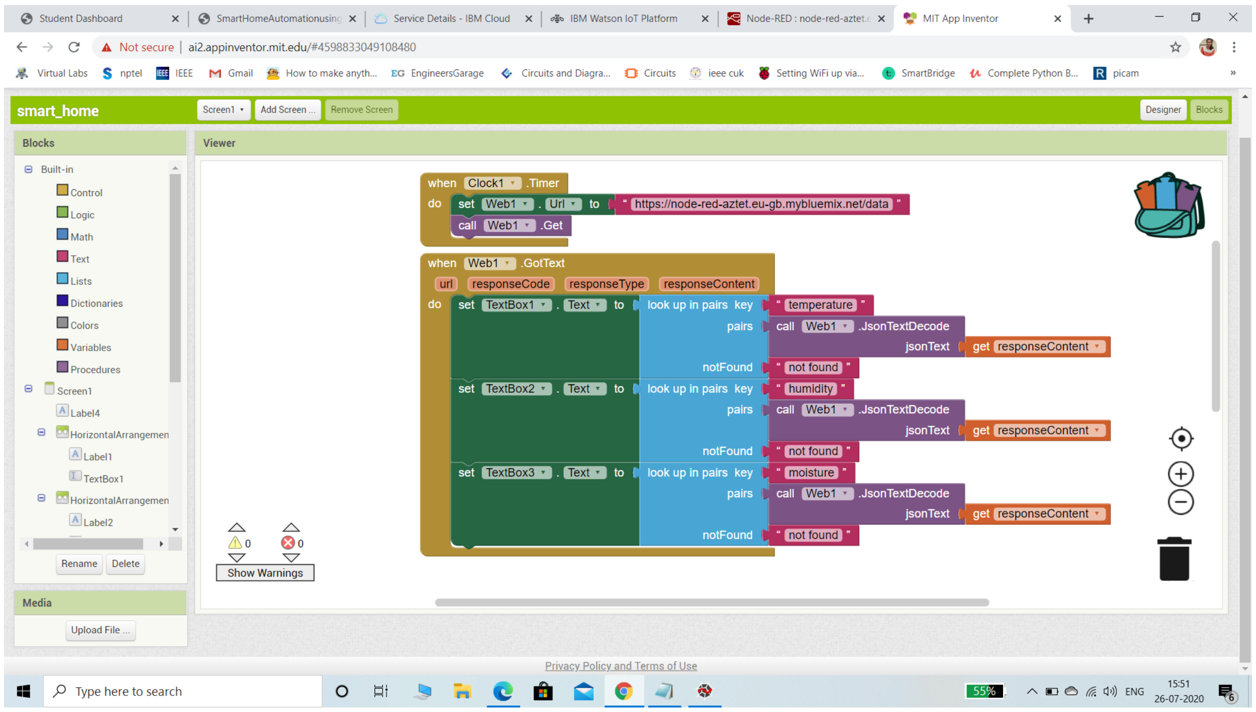
```
7/26/2020, 5:38:32 PM   node: ee5f27ba.a5d6c8
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json :
msg.payload : number
17
7/26/2020, 5:38:33 PM   no
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json :
msg.payload : number
82
7/26/2020, 5:38:34 PM   node: ee5f27ba.a5d6c8
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json :
msg.payload : Object
▸{ Temperature: 40, Humidity: 16,
Moisture: 91 }
7/26/2020, 5:38:35 PM   node: ee5f27ba.a5d6c8
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json :
msg.payload : number
40
7/26/2020, 5:38:36 PM   node: ee5f27ba.a5d6c8
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json :
msg.payload : number
16
7/26/2020, 5:38:37 PM   node: ee5f27ba.a5d6c8
iot-2/type/raspberrypi/id/123456/evt/Weather/fmt/json :
```

STEP 4: Configure the Node red and install the required nodes to get the weather details from Open Weather API by using http request and send message alerts to user .

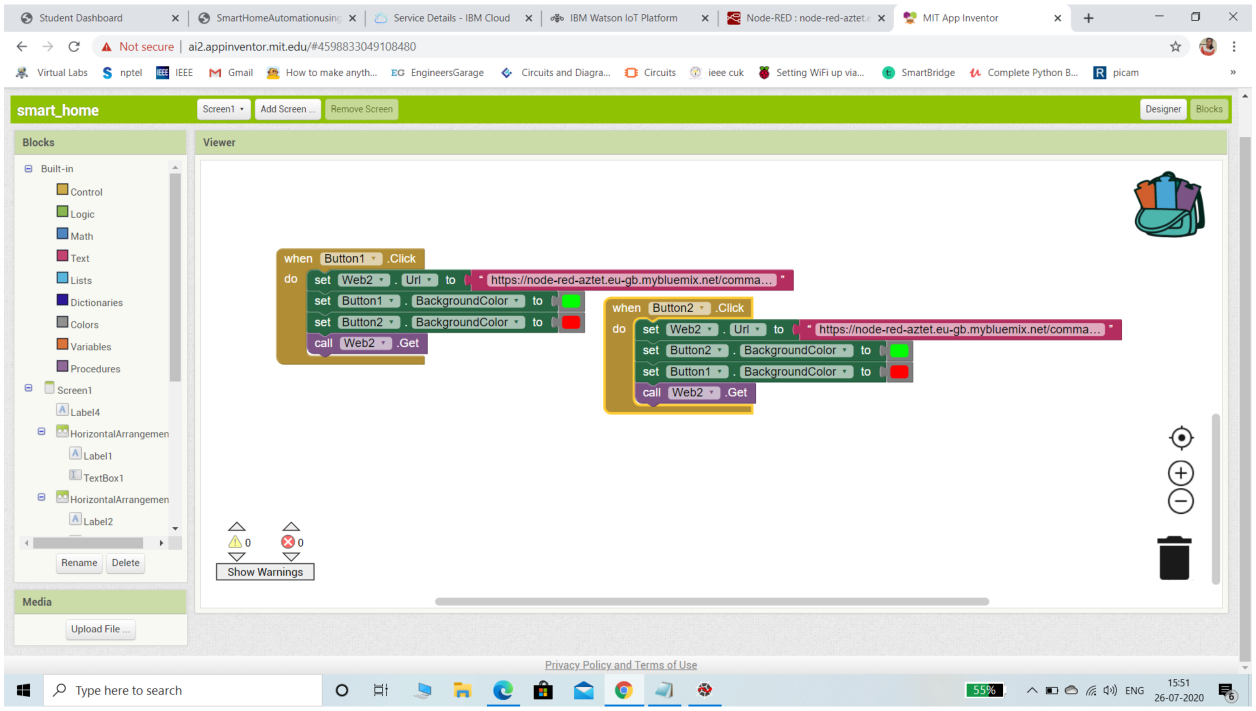


```
7/26/2020, 5:16:13 PM   node: 74b25016.e7463
msg.payload : Object
▾object
 ▸coord: object
 ▸weather: array[1]
  base: "stations"
 ▸main: object
  visibility: 6000
 ▸wind: object
 ▸clouds: object
  dt: 1595763771
 ▸sys: object
  timezone: 19800
  id: 1269843
  name: "Hyderabad"
  cod: 200
```

STEP 5: For mobile application we used MIT App Inventor and created an application for viewing details of Temperature,Humidity,Moisture content and send commands to Turn ON the motor or Turn OFF the motor.



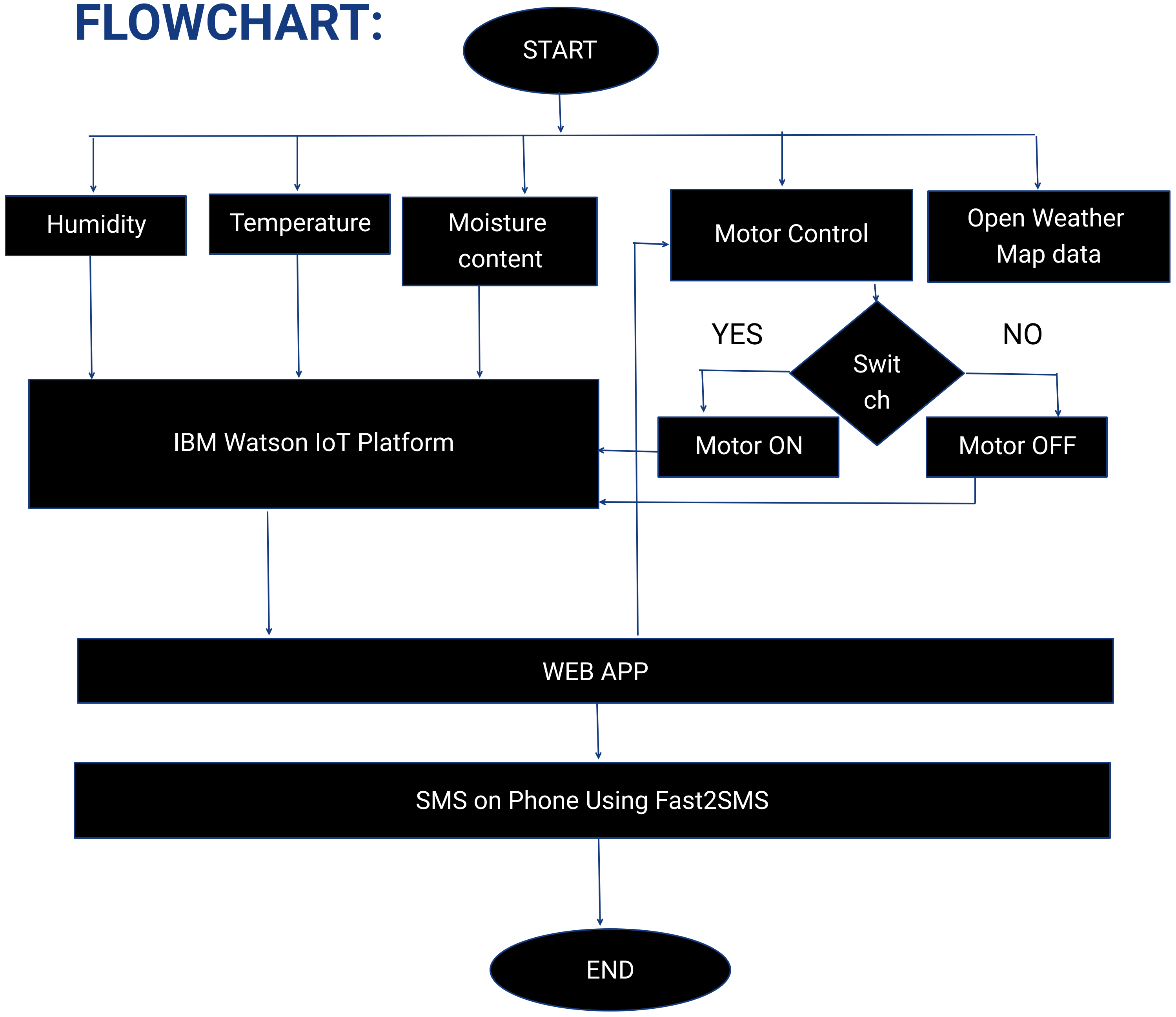
```
when Clock1 .Timer
do  set Web1 . Url to "https://node-red-aztet.eu-gb.mybluemix.net/data"
    call Web1 .Get

when Web1 .GotText
  url  responseCode  responseType  responseContent
do  set TextBox1 . Text to  look up in pairs key  "temperature"
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  "not found"
    set TextBox2 . Text to  look up in pairs key  "humidity"
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  "not found"
    set TextBox3 . Text to  look up in pairs key  "moisture"
                                           pairs  call Web1 .JsonTextDecode
                                                               jsonText  get responseContent
                                        notFound  "not found"
```



```
when Button1 .Click
do  set Web2 . Url to "https://node-red-aztet.eu-gb.mybluemix.net/comma..."
    set Button1 . BackgroundColor to 🟩
    set Button2 . BackgroundColor to 🟥
    call Web2 .Get

when Button2 .Click
do  set Web2 . Url to "https://node-red-aztet.eu-gb.mybluemix.net/comma..."
    set Button2 . BackgroundColor to 🟩
    set Button1 . BackgroundColor to 🟥
    call Web2 .Get
```

STEP 6: Finally after completing the above steps,we get a Mobile Application,webpage Application to View weather details,sensor values and control the motors. Even Message alerts are sent to user when moisture is low by fast2SMS service.

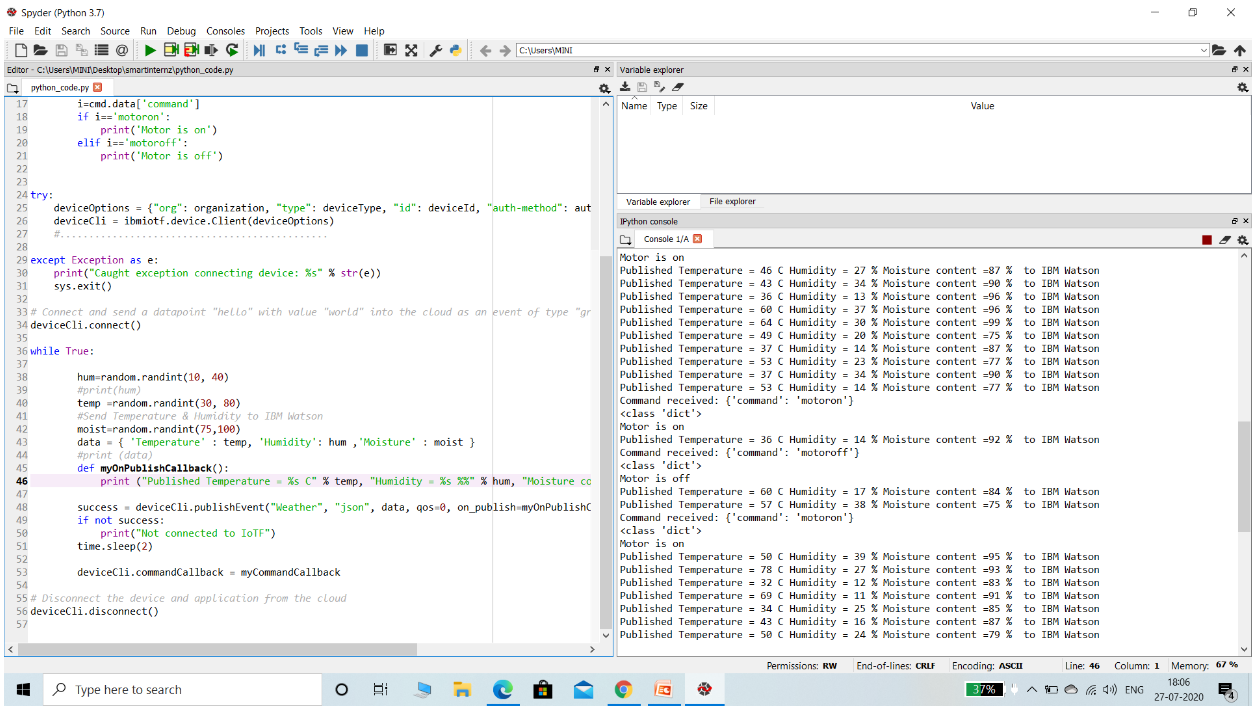
FLOWCHART:


START
Humidity
Temperature
Moisture content
Motor Control
Open Weather Map data
YES
Swit ch
NO
Motor ON
Motor OFF
IBM Watson IoT Platform
WEB APP
SMS on Phone Using Fast2SMS
END

RESULTS :

1. When we connect with IoT Platform, we get values like temperature, humidity and moisture content, which can be viewed and analysed in cloud.

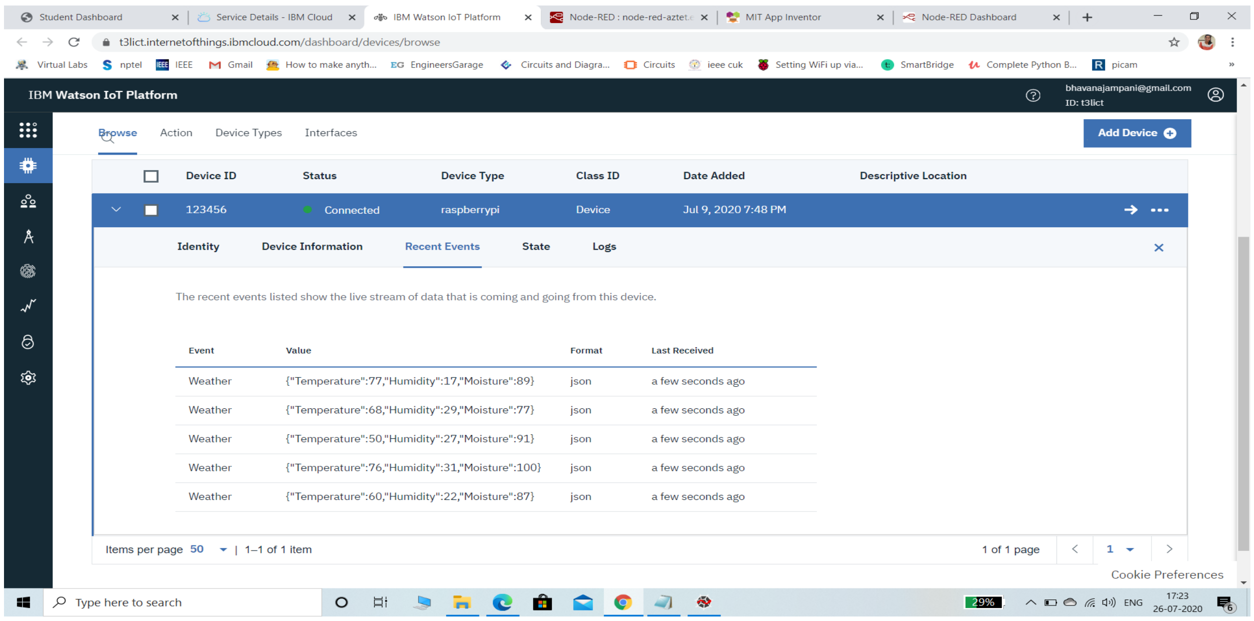
2. Python code to publish and subscribe to IBM IoT Platform for sending sensor values and receiving the commands. The details can be viewed in cloud and webpage user interface.


Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\MINI
Editor - C:\Users\MINI\Desktop\smartinternz\python_code.py
python_code.py
Variable explorer
Name Type Size Value
Variable explorer File explorer
IPython console
Console 1/A
Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 46 Column: 1 Memory: 67 %
Type here to search
37%
ENG
18:06
27-07-2020

```
        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": aut
    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 "gr
deviceCli.connect()

while True:

        hum=random.randint(10, 40)
        #print(hum)
        temp =random.randint(30, 80)
        #Send Temperature & Humidity to IBM Watson
        moist=random.randint(75,100)
        data = { 'Temperature' : temp, 'Humidity': hum ,'Moisture' : moist }
        #print (data)
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % hum, "Moisture co

        success = deviceCli.publishEvent("Weather", "json", data, qos=0, on_publish=myOnPublishC
        if not success:
            print("Not connected to IoTF")
        time.sleep(2)

        deviceCli.commandCallback = myCommandCallback

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

```
Motor is on
Published Temperature = 46 C Humidity = 27 % Moisture content =87 %  to IBM Watson
Published Temperature = 43 C Humidity = 34 % Moisture content =90 %  to IBM Watson
Published Temperature = 36 C Humidity = 13 % Moisture content =96 %  to IBM Watson
Published Temperature = 60 C Humidity = 37 % Moisture content =96 %  to IBM Watson
Published Temperature = 64 C Humidity = 30 % Moisture content =99 %  to IBM Watson
Published Temperature = 49 C Humidity = 20 % Moisture content =75 %  to IBM Watson
Published Temperature = 37 C Humidity = 14 % Moisture content =87 %  to IBM Watson
Published Temperature = 53 C Humidity = 23 % Moisture content =77 %  to IBM Watson
Published Temperature = 37 C Humidity = 34 % Moisture content =90 %  to IBM Watson
Published Temperature = 53 C Humidity = 14 % Moisture content =77 %  to IBM Watson
Command received: {'command': 'motoron'}
<class 'dict'>
Motor is on
Published Temperature = 36 C Humidity = 14 % Moisture content =92 %  to IBM Watson
Command received: {'command': 'motoroff'}
<class 'dict'>
Motor is off
Published Temperature = 60 C Humidity = 17 % Moisture content =84 %  to IBM Watson
Published Temperature = 57 C Humidity = 38 % Moisture content =75 %  to IBM Watson
Command received: {'command': 'motoron'}
<class 'dict'>
Motor is on
Published Temperature = 50 C Humidity = 39 % Moisture content =95 %  to IBM Watson
Published Temperature = 78 C Humidity = 27 % Moisture content =93 %  to IBM Watson
Published Temperature = 32 C Humidity = 12 % Moisture content =83 %  to IBM Watson
Published Temperature = 69 C Humidity = 11 % Moisture content =91 %  to IBM Watson
Published Temperature = 34 C Humidity = 25 % Moisture content =85 %  to IBM Watson
Published Temperature = 43 C Humidity = 16 % Moisture content =87 %  to IBM Watson
Published Temperature = 50 C Humidity = 24 % Moisture content =79 %  to IBM Watson
```

3. Status of sensor and motor on IBM IoT Platform and event created due to Motor On/Off command sent by the user interface.



Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456	Connected	raspberrypi	Device	Jul 9, 2020 7:48 PM	

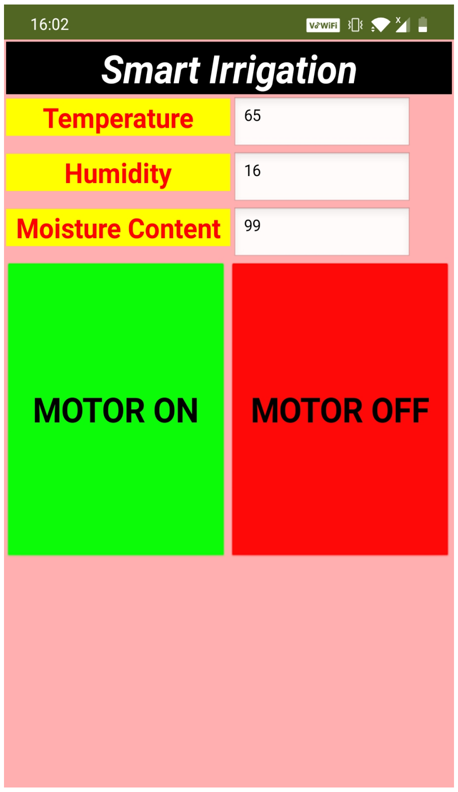
The recent events listed show the live stream of data that is coming and going from this device.

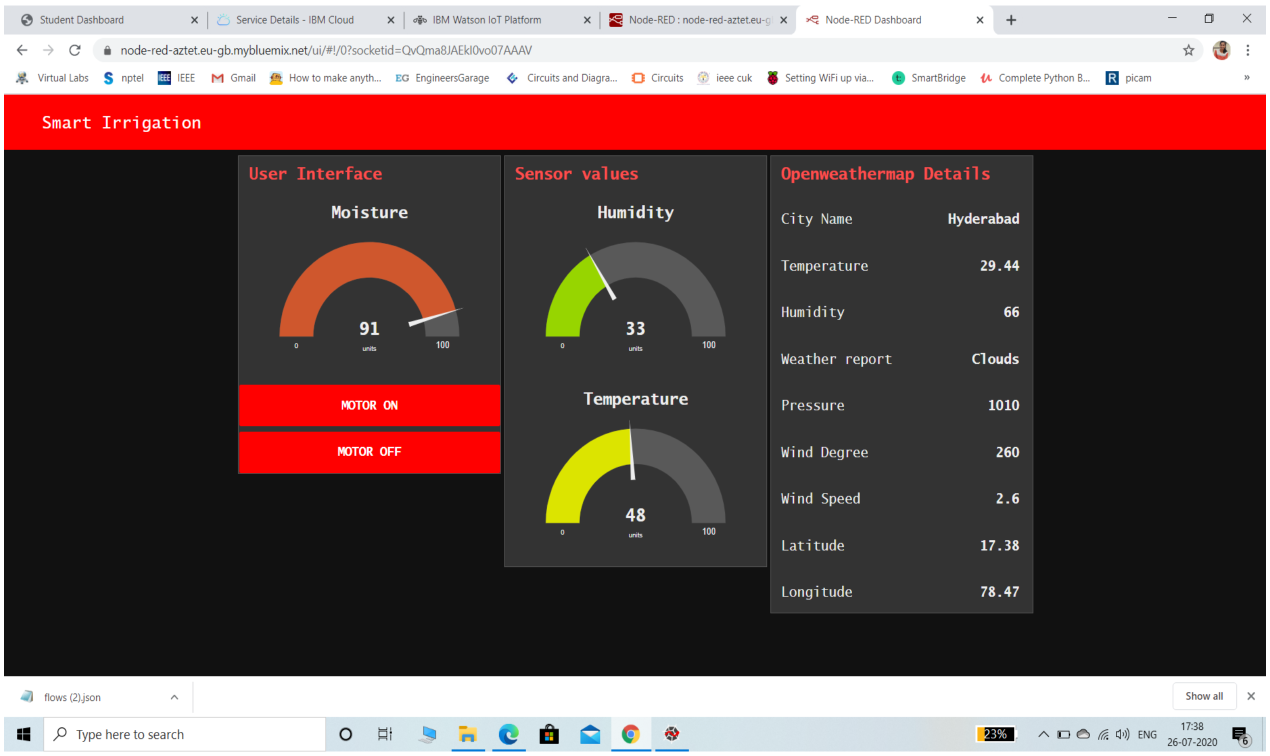
Event	Value	Format	Last Received
Weather	{"Temperature":77,"Humidity":17,"Moisture":89}	json	a few seconds ago
Weather	{"Temperature":68,"Humidity":29,"Moisture":77}	json	a few seconds ago
Weather	{"Temperature":50,"Humidity":27,"Moisture":91}	json	a few seconds ago
Weather	{"Temperature":76,"Humidity":31,"Moisture":100}	json	a few seconds ago
Weather	{"Temperature":60,"Humidity":22,"Moisture":87}	json	a few seconds ago

Items per page 50 | 1–1 of 1 item 1 of 1 page 1

Cookie Preferences

4. Mobile Application and webpage application for sensor values ,weather conditions and controlling motors accordingly.

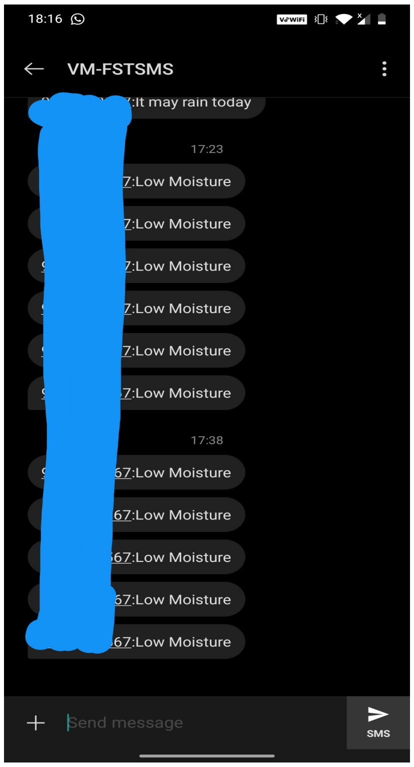




Smart Irrigation

User Interface	Sensor values	Openweathermap Details	
Moisture: 91	Humidity: 33	City Name	Hyderabad
MOTOR ON	Temperature: 48	Temperature	29.44
MOTOR OFF		Humidity	66
		Weather report	Clouds
		Pressure	1010
		Wind Degree	260
		Wind Speed	2.6
		Latitude	17.38
		Longitude	78.47

5. Sending sms to the phone using Fast2SMS when moisture content is low.



VM-FSTSMS

[redacted]:It may rain today

17:23

[redacted]7:Low Moisture

[redacted]7:Low Moisture

[redacted]7:Low Moisture

[redacted]7:Low Moisture

[redacted]7:Low Moisture

[redacted]7:Low Moisture

17:38

[redacted]67:Low Moisture

[redacted]67:Low Moisture

[redacted]67:Low Moisture

[redacted]67:Low Moisture

[redacted]67:Low Moisture

ADVANTAGES AND DISADVANTAGES:

ADVANTAGES:

- One of the greatest advantages of a smart irrigation system is ability to save water. Generally speaking, traditional watering methods can waste as much as 50% of the water used due to inefficiencies in irrigation, evaporation and overwatering.
- Farmers can visualize production levels, soil moisture, sunlight intensity and more in real time and remotely to accelerate decision making process.
- They can also get important information about the amount of air, wind, humidity and temperature of their area.
- Weather prediction and moisture content allows for water use only whenever it is needed.
- Analysing production quality and results in correlation to treatment can teach farmers to adjust processes to increase quality of the product.
- Local and commercial farmers can monitor multiple fields in multiple location around the globe from an internet connection. Decision can be made in real - time and from anywhere.
- Optimized crop treatment such as accurate planting, watering, pesticides application and harvesting directly affects production rates.

DISADVANTAGES

One huge disadvantage of smart irrigation system is that it requires an unlimited or continuous internet connection to be successful. This means that in rural communities, especially in the developing countries where we have mass crop production, it is completely impossible to operate the agriculture methods. In places where internet connections are frustratingly slow, smart farming will be impossible.

As pointed out earlier, smart irrigation uses high technology that require technical skills and precisions to make it a success. however, many farmers do not have these skills. Even finding someone with these skills is difficult or even expensive most of the times. So, this can be a discouraging factor hindering a lot of promising farmers from adopting it.

APPLICATIONS:

There are quite a few and the ranking of the best or not depends on the need of the farms land and region or let's say typically need of local farmers.

- Monitoring quality of soil in real time.
- Regulating water supply and controlling usage of water.
- Monitoring and measuring humidity, temperature etc.
- Crop health.
- Remote control of motor for irrigation.

CONCLUSION:

Since IoT farming application are making it worth able for farmers to gather important information leading to improvement in the quality of the crops. Many land owners must comprehend the capability of IoT usage for farming by introducing smart innovation to increase output. The need for increasing population can be fulfilled if the user can use IoT technology in a successful manner. In this report, the answer for analysing smart irrigation has been exhibited.

IoT based Smart Irrigation improves the entire Agriculture system by monitoring the field in real-time. With the help of sensors and interconnectivity, the Internet of Things in Irrigation has not only saved the time of the farmers but also reduced the extravagant use of resources such as Water and Electricity.

Thus the smart irrigation based on IoT will make it possible for farmers to understand the potential of IoT market for agriculture by installing smart technologies to increase competitiveness and sustainability to their productions.
With the population proliferating, the demand can be successfully met if the ranchers, as well as small farmers, implement agriculture IoT solutions in a prosperous manner.

FUTURE SCOPE:

Future scope for this project is with the concept of IoT, given below are some of the main future scope of IoT in field of irrigation:

- **Smart agri-logistics:** It is all about smart fooding and agri-business. It focuses on serving fresh product quality and natural production process with flexible chain- and compassing tracking and tracing system.
- **Smart Food Awareness:** It deals with customer profile, health and normal day's in the future super market. The demand for healthier but enjoyable diet is increasing, so we need to consider and serve it.
- **Precision Farming:** Precision farming, or precision agriculture, is a umbrella concept for IoT- based approaches that make farming more controlled and accurate. In simple words, plants and cattle get precisely the treatment they needed, determined by machines with superhuman accuracy.
- **Green Agriculture:** IoT driven smart greenhouse can intelligently monitor as well as control the climate, eliminating the need for manual interventions. Various sensors are deployed to measure the environmental parameters according to the specific requirement of the crops.

BIBLIOGRAPHY:

- https://www.manage.gov.in/studymaterial/CCA-E.pdf
- https://www.ijert.org/providing-smart-agricultural-solutions-to-farmers-for-better-yielding-using-iot
- https://cloud.ibm.com/docs/IoT?topic=IoT-getting-started
- https://nodered.org/about/
- https://en.wikipedia.org/wiki/OpenWeatherMap
- http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/22755/1/Smart%20Farming%20using%20IoT.pdf

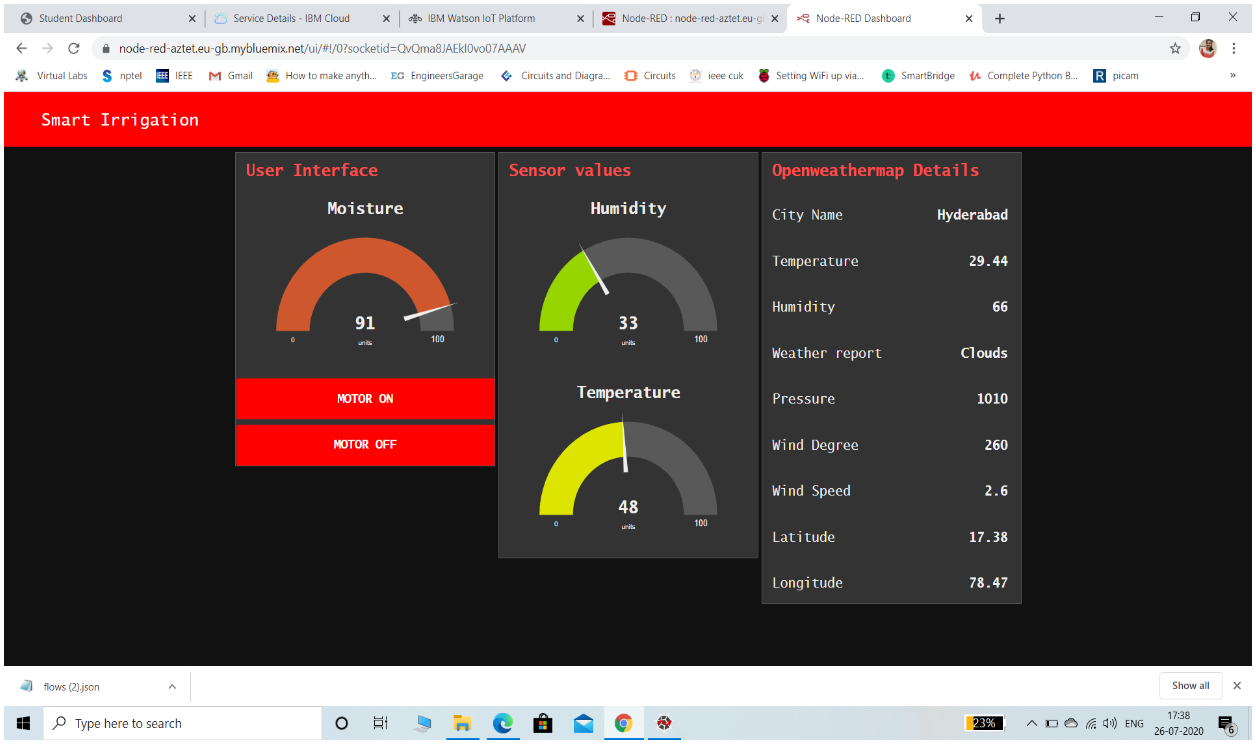
APPENDIX:

Source Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "t3lict"
deviceType = "raspberrypi"
deviceId = "123456"
authMethod = "token"
authToken = "12345678"


def myCommandCallback(cmd):
        print("Command received: %s" % cmd.data)#Commands
        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()
```

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

while True:

        hum=random.randint(10, 40)
        #print(hum)
        temp =random.randint(30, 80)
        #Send Temperature & Humidity to IBM Watson
        moist=random.randint(75,100)
        data = { 'Temperature' : temp, 'Humidity': hum ,'Moisture' : moist }
        #print (data)
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % hum,
"Moisture content =%s %% " % moist ,"to IBM Watson")

        success = deviceCli.publishEvent("Weather", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(2)

        deviceCli.commandCallback = myCommandCallback

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

UI Output:



Smart Irrigation

User Interface

Moisture: 91 units (0–100)

MOTOR ON

MOTOR OFF

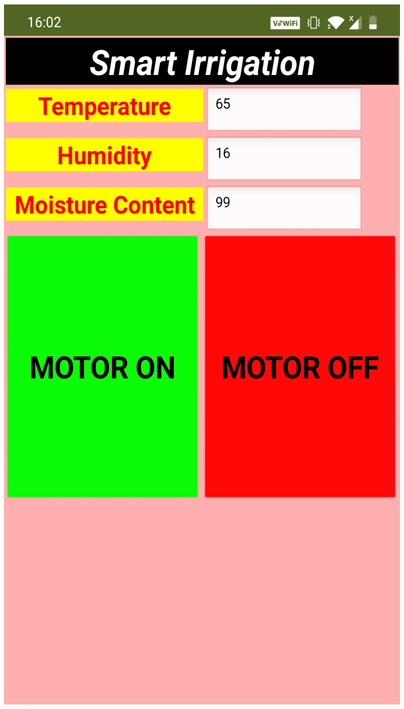
Sensor values

Humidity: 33 units (0–100)

Temperature: 48 units (0–100)

Openweathermap Details

City Name	Hyderabad
Temperature	29.44
Humidity	66
Weather report	Clouds
Pressure	1010
Wind Degree	260
Wind Speed	2.6
Latitude	17.38
Longitude	78.47



Smart Irrigation

Temperature	65
Humidity	16
Moisture Content	99

MOTOR ON

MOTOR OFF