PROJECT REPORT INDEX

CHAPTER 1

INTRODUCTION

1.1 Overview

The world population is exploding. Providing the basic need has become a challenge. The basic requirement for any person is healthy food. But due to the increasing population, the old farming methods are proving insufficient for providing food in large scale. Agriculture is also major source of income for the largest population in India and is major contributor to Indian economy. In past decade it is observed that there are not much crop development in agriculture sector. Food prices are continuously increasing because crop rate declined. There may be enormous reasons for this like excess water, not harvesting the crop on time or even due to climate changes.

So it is very important to make effective use of present technologies to make sure that farmer gets good yield and there is some positive change in the field of agriculture.

So we proposed to develop a Smart Farming System that uses advantages of cutting edge technologies such as IoT, Wireless Sensor Network and Cloud computing to help farmers enhance the way farming is done.

Using sensors like temperature, humidity, moisture (Online IoT simulator) etc. are used to get information about the field and help farmers to take precise decisions on insights and recommendations based on the collected data.

1.2 Purpose

By the report submitted by Forbes, till 2050 population of the world will be more than 9.6 billion consequently leading to 70% increase in food consumption. Since there is limited arable land the only way to overcome this crisis is to plan smartly. As smart technologies make headway across industries, the agriculture sector is not to be left behind.

In Agriculture data is very important for the success of a specific crop in a given geographical climate. Such information is particularly of interest to the farmer, Agriculture consultants, and Agriculture companies. Agriculture depends on data for information such as success rate of fertilizers, the impact of natural calamities on soil and more.

So, the purpose of project is developing Smart Agriculture System based on IoT which can monitor and generate data regarding soil moisture and climatic conditions to grow and yield a good crop. Farmer is provided a mobile app using which he can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details. Based on all the parameters he can water his crop by controlling the motors using the mobile application. Even if the farmer is not present near his crop he can water his crop by controlling the motors using the mobile application from anywhere.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing problem

We all know that agriculture is major contributor to Indian GDP. The agriculture sector is at the third spot and contributing around 16% of the Indian GDP. A number of different factors can cause agricultural productivity to increase or decrease.

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 agricultural sector as a whole. Studies by Indian Agricultural Research Institute (IARI) and others indicate greater expected loss in the Rabi crop. Every 1°C rise in temperature reduces wheat production by 4-5 Million 11 Tones.

Other problems faced by farmers include:

- Many trips have to be taken in order to check the soil humidity on a regular basis manually.
- Many trips have to be taken in order to manually check the soil humidity on a regular basis.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
- Manually measuring key data points about crops is often difficult, time-consuming, and more likely to be inaccurate.
- It is sometimes difficult to know the optimal time to plant without data.
- Overwatering crops could lead to higher water costs than what is really needed.

2.2 Proposed Solution

Thus, to overcome the problems mentioned above, we came up with a solution on developing smart agriculture system with the use of emerging technologies like IOT, cloud computing etc. The Internet of things (IoT), which is the ability for technology in everyday objects to send and receive data, will revolutionize how we do everything from transportation to communication. Agriculture also stands to benefit greatly from integrating this technology into simple electronics.

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. This data are being used to provide predictive insights in farming operations, drive real-time operational decisions, and redesign business processes.

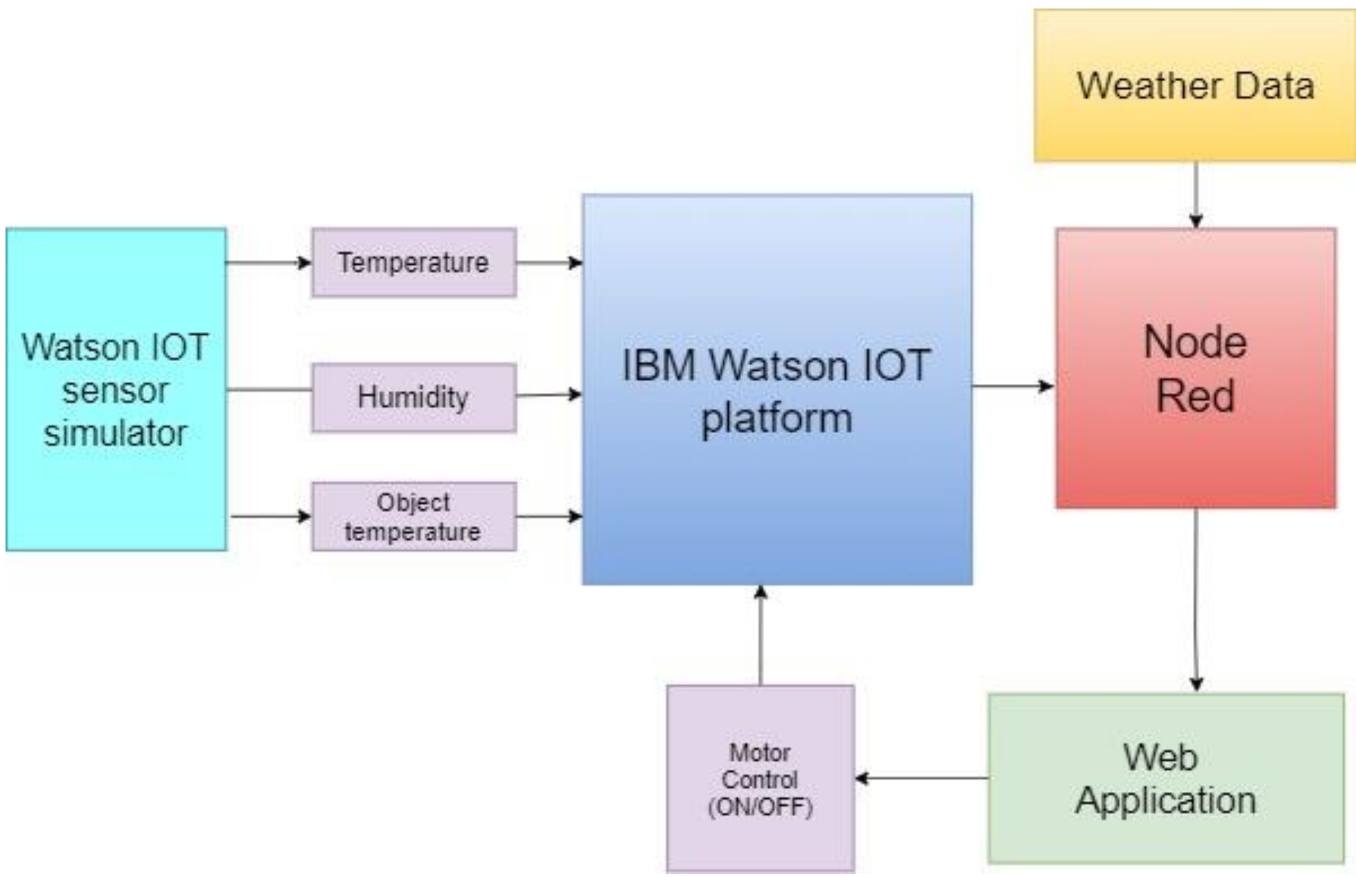
The central idea of our project is to develop a smart agriculture device which can monitor the temperature, soil moisture of the crop. This collected data is being continuously sent to cloud. The farmer can also get the real time weather forecasting data.Farmer can monitor the temperature, humidity and soil moisture parameters along with weather forecasting details through a web page. Based on all the parameters he can water his crop through the web page. Even if the farmer is not present near his crop he can water his crop remotely.

Here we are using online IOT simulator to the data regarding crop like temperature, humidity and soil moisture, we use Open Weather API to get weather report of that specific location, node red to create the webpage or UI and IBM Iot cloud to collect the data.

CHAPTER 3

THEORITICAL ANALYSIS

3.1 Block Diagram



3.2 Software Designing

- IBM Watson IOT Platform

IBM Watson IoT Platform is a fully managed, cloud-hosted service that makes it simple to derive value from Internet of Things (IoT) devices.

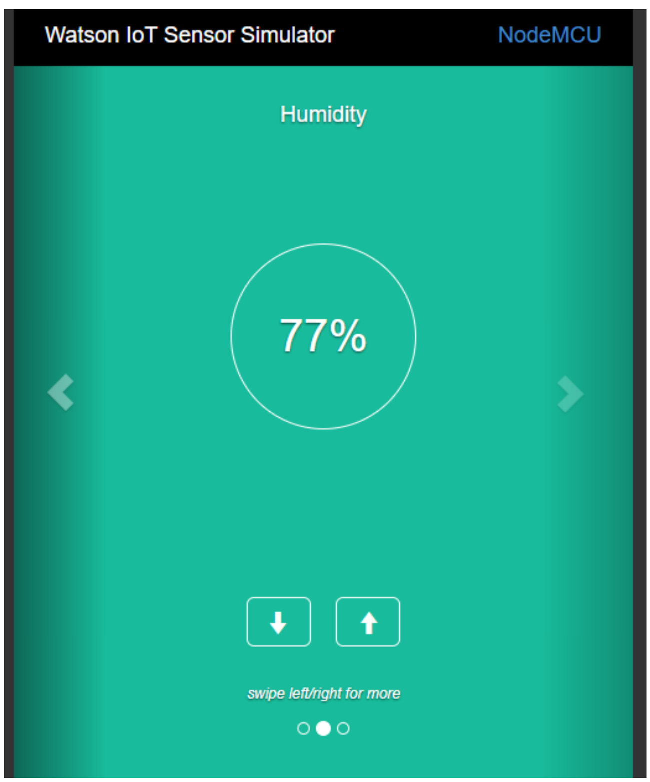
Simply register and connect your device, be it a sensor, a gateway, or something else, to Watson IoT Platform and start sending data securely up to the cloud using the open, lightweight MQTT messaging protocol. You can set up and manage your devices using your online dashboard or our secure APIs, so that your apps can access and use your live and historical data.

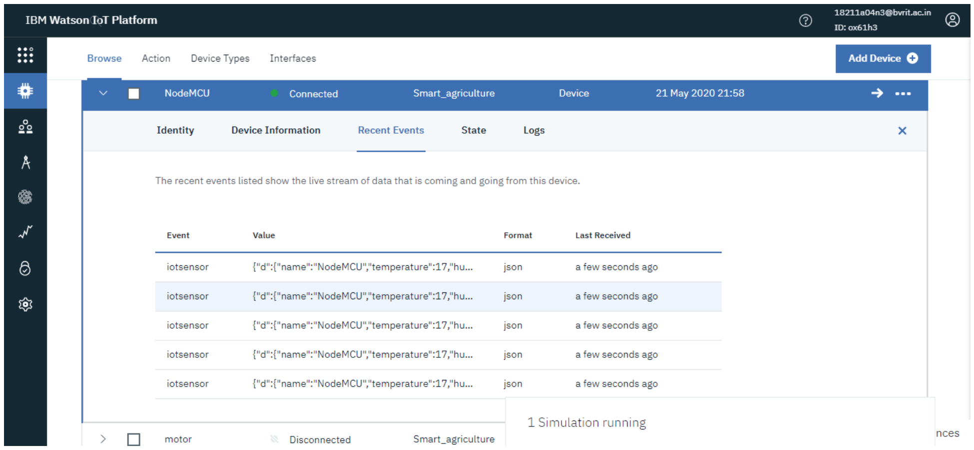
Before you can begin receiving data from your IoT devices, you must connect them to Watson IoT Platform. Connecting a device to Watson IoT Platform involves registering the device with Watson IoT Platform and then using the registration information to configure the device to connect to Watson IoT Platform.
After selecting the device type and registering the device, you get details like device type, device id, organisation id, authentication token etc. Save these details as they are important when you want to connect the device with IoT sensor simulation.

- Watson IOT Sensor Simulator

It is used as online sensor simulator, to get the values of temperature,humidity,object temperature. The working is as follows:

1. To view the simulated sensor, we must go to http://quickstart.internetofthings.ibmcloud.com/iotsensor, enter the saved details(i.e device id, device type, authentication token etc).The simulator connects automatically and starts publishing data. The simulator must remain connected to visualize the data.
2. We can use simulator buttons to change the simulated sensor readings. Data is published periodically.
3. We can view the simulated data in cloud platform by selecting the device type and then clicking recent activity to view the generated data.





- Open Weather API

OpenWeatherMap is an online service that provides weather data. It is owned by OpenWeather Ltd headquartered in London, United Kingdom.
It provides 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 developed for getting different types of weather data. They support multiple languages, units of measurement and data formats.
Additionally, the OpenWeatherMap 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.
These collected details are sent to webpage and can be viewed by the farmer.

OpenWeather

- Node RED

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 web browser-based flow editor, which can be used to create JavaScript functions. Elements of applications can be saved or shared for re-use. The runtime is built on Node.js.
In 2016, IBM contributed 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 the webpage.
With the help of data viewed in the webpage, if the farmer feels that there is a need to turn the motor on or off, he can do it remotely through the webpage.

- 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 use python programming for motor control though webpage.

CHAPTER 4

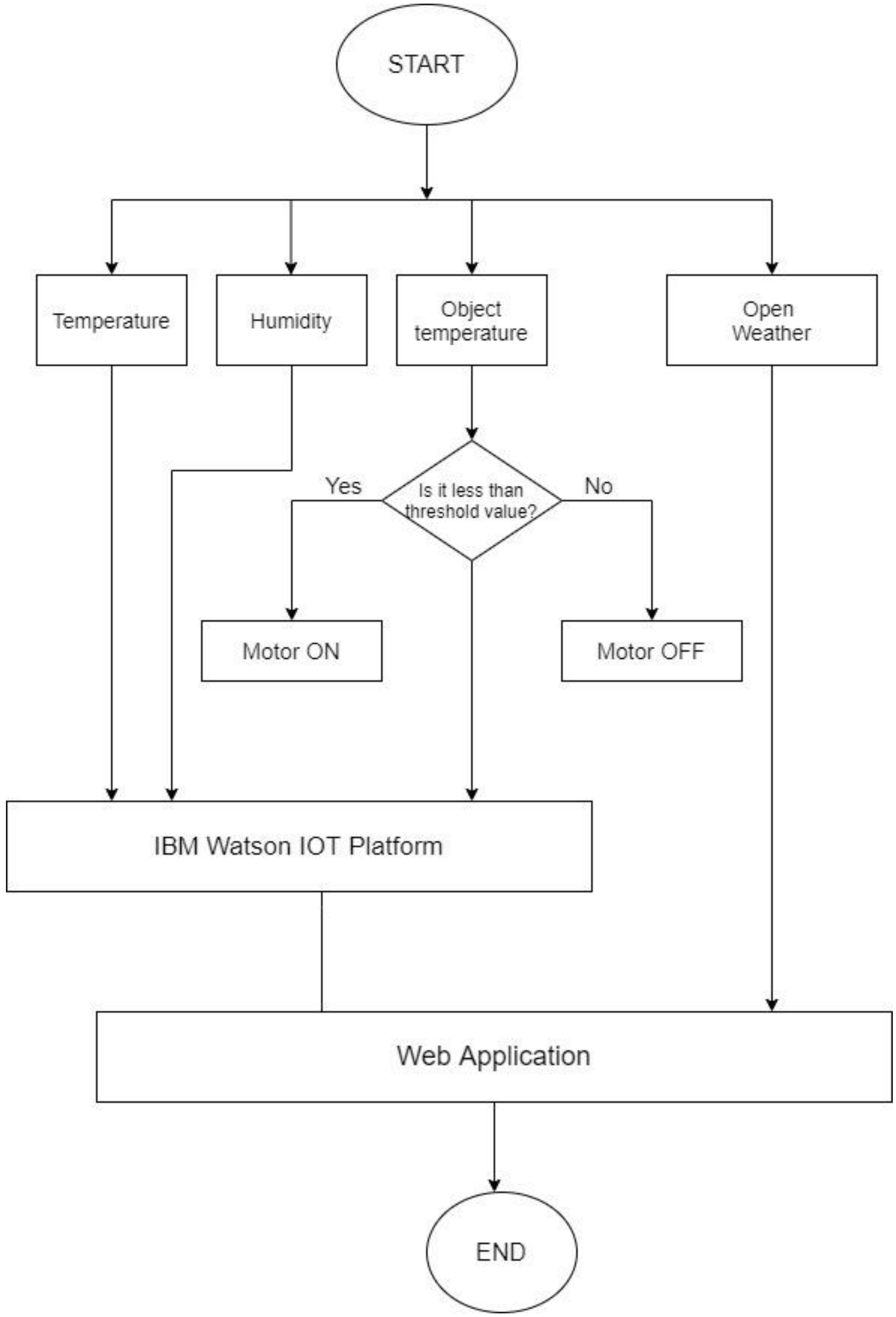
EXPERIMENTAL INVESTIGATIONS

The target of this project is to control the motor remotely by detecting the soil moisture, soil temperature
and weather conditions of the particular area. The motor is operated using User Interface or a web application. And the steps involved to achieve this are:

- Create a device in IBM Watson IOT platform, and save the details so that can we used further in project.
- After creating the device, enter the device details like device id, organisation id, device type authentication token in order to connect with IOT sensor simulator, so that we get the values like temperature, humidity and object temperature, which can be viewed and analysed in cloud.
- Configure the Nodered and install the required nodes to get the data from IBM IOT platform. Configure the Nodered and install the required nodes to get the weather details from Open Weather API using http requests.
- Configure the device to receive data from web application and control motors through we page. Here we write python code to subscribe to IBM IOT platform and get the commands. The details can be viewed in cloud.
- After completing the above steps, we finally get a UI or webpage which can be used by farmer to monitor his crop growth and know the weather details.

CHAPTER 5

FLOW CHART

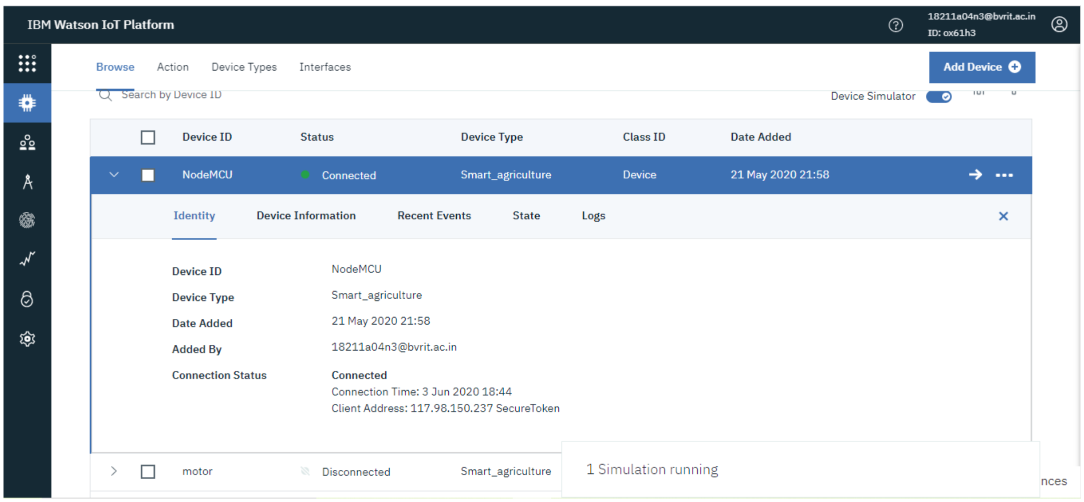


CHAPTER 6

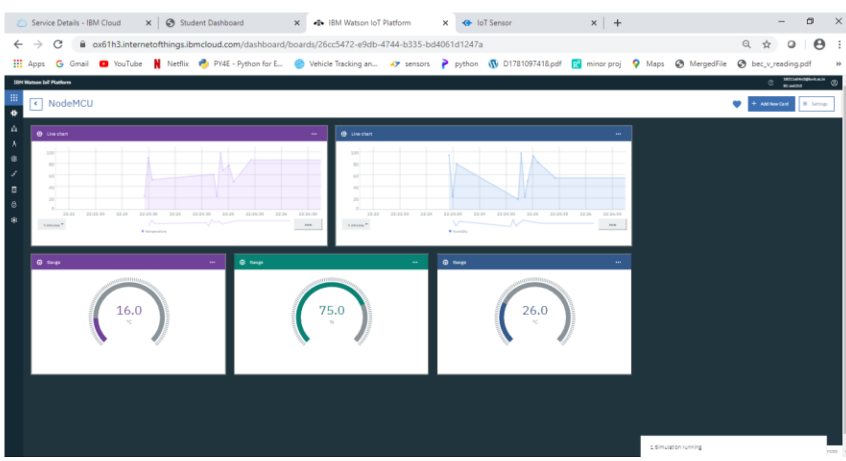
RESULTS

The main objective of the project is creating a webpage through which farmer can monitor soil moisture, temperature of the crop, the weather details in that particular area and also control the motor remotely through the web application.
The steps involved to achieve this and the results (in image format) are shown below.

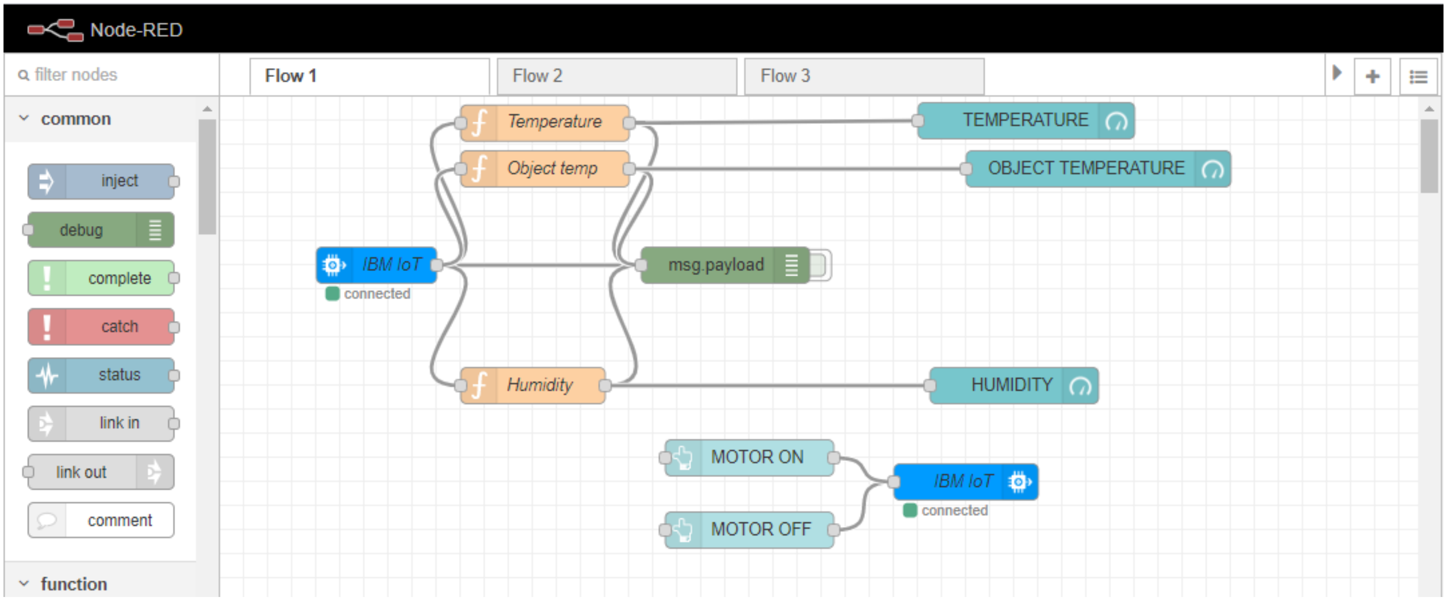
Step 1: Create a device in IBM Watson IOT platform, and save the details so that can we used further in project.



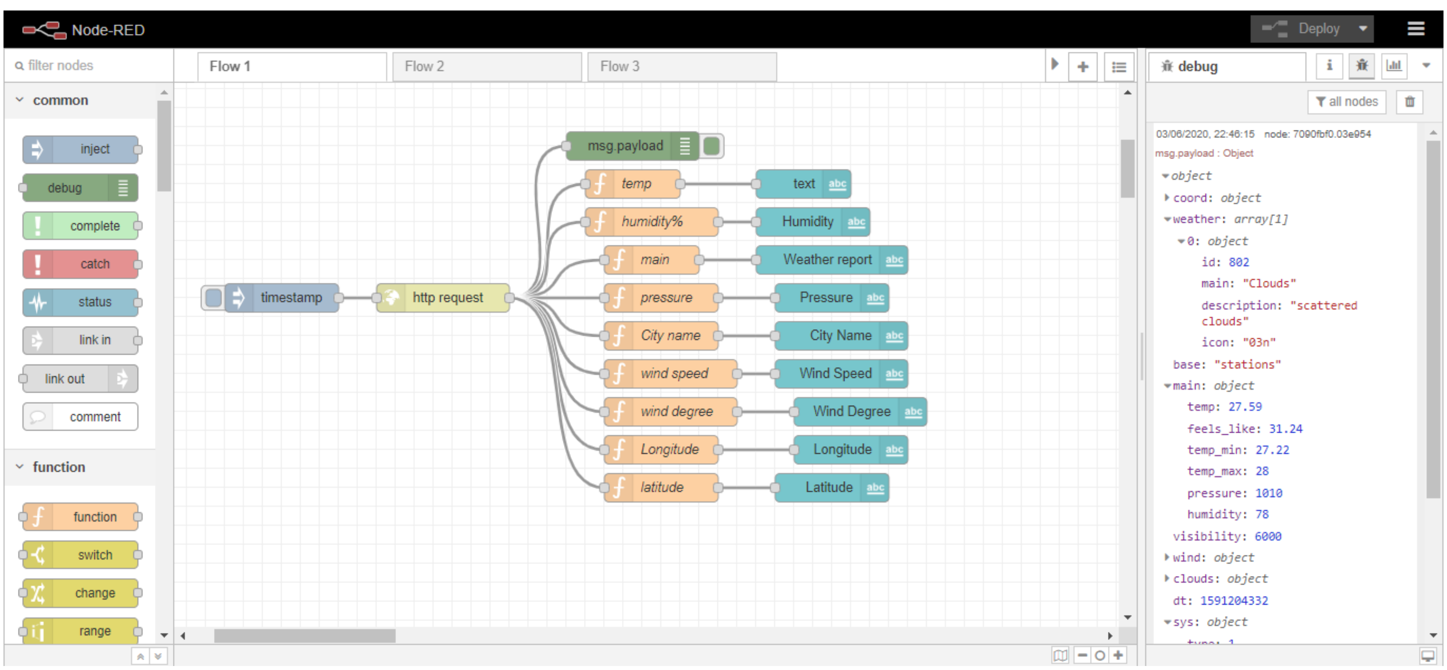
Step 2: After creating the device, enter the device details like device id, organisation id, device type authentication token in order to connect with IOT sensor simulator, so that we get the values like temperature, humidity and object temperature, which can be viewed and analysed in cloud.



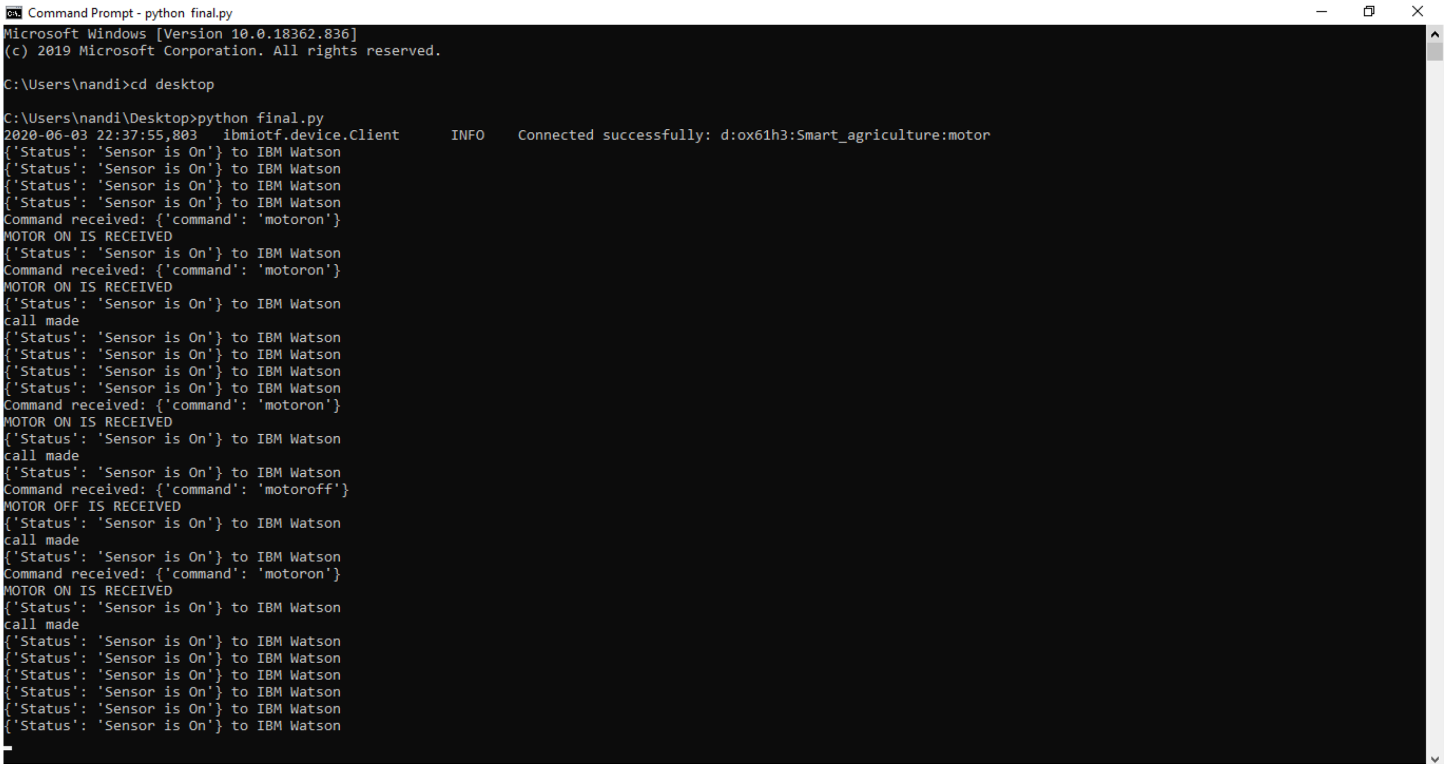
Step 3: Configure the Nodered and install the required nodes to get the data from IBM IOT platform.

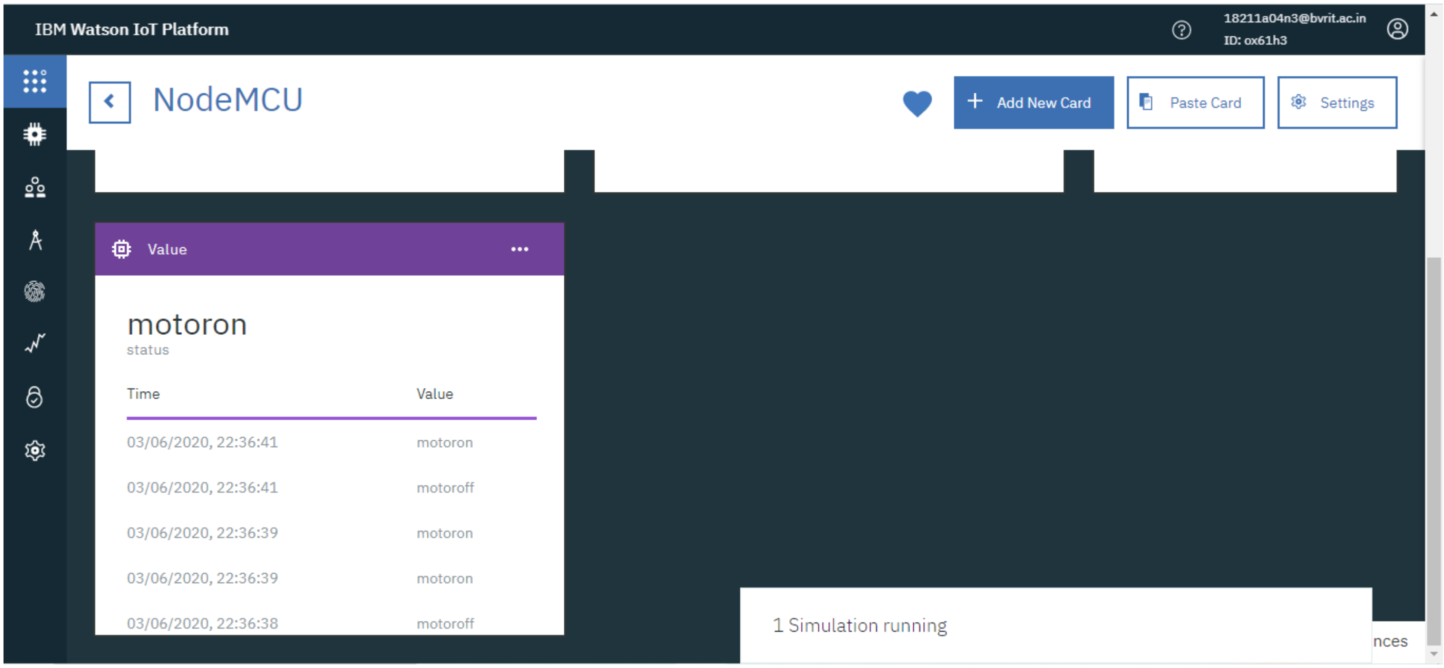


Step 4: Configure the Nodered and install the required nodes to get the weather details from Open Weather API using http requests.

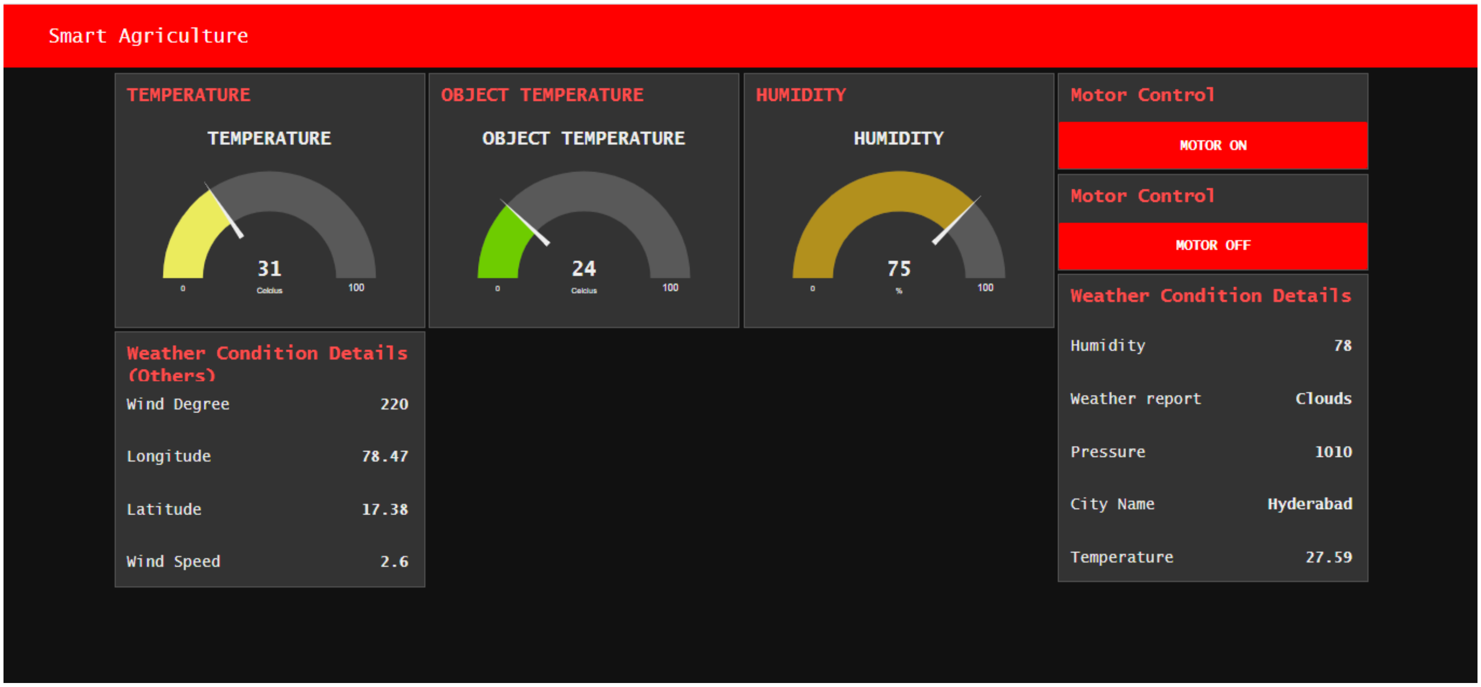


Step 5: Configure the device to receive data from web application and control motors through we page. Here we write python code to subscribe to IBM IOT platform and get the commands. The details can be viewed in cloud and webpage.





Step 6: After completing the above steps, we finally get a UI or webpage which can be used by farmer to monitor his crop growth and know the weather details.



CHAPTER 7

ADVANTAGES AND DISADVANTAGES

Advantages

- 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 to get important information about the amount of air and the levels of air, sound, humidity, and temperature of your environment.

- Weather predictions and soil moisture sensors allow for water use only when and where needed.

- Analyzing 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 locations around the globe from an internet connection. Decisions can be made in real-time and from anywhere.

- Optimized crop treatment such as accurate planting, watering, pesticide application and harvesting directly affects production rates.

Disadvantages

One huge disadvantage of smart agriculture 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 this agriculture method. In places where internet connections are frustratingly slow, smart farming will be an impossibility.

As pointed out earlier, smart agriculture uses high techs that require technical skill and precision to make it a success. However, many farmers do not have these skills. Even finding someone with this technical ability is difficult or even expensive to come by, at most. And, this can be a discouraging factor hindering a lot of promising farmers from adopting it.

CHAPTER 8

APPLICATIONS

There are quite a few and the ranking of bestor not depends on the need of the farmland and the region or lets say typically need of local farmer.

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

CHAPTER 9

CONCLUSION

Since IoT farming application are making it workable for farmer to gather important information leading to improvement in the quality of their crop. 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 agriculture has been exhibited

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

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

CHAPTER 10

FUTURE SCOPE

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

- **Smart agri-logistic:** It is all about smart fooding and agri-business. It focuses on servicing 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 awareness 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. Therefore we have to develop a system using iot which will aim for creating awareness in food quality.
- **Precision Farming**: Precision farming is a farming practice that are more accurate and controlled. It deals with production of crop along with raising livestock. In this farming techniques, we use component such as SN, system control, robots, autonomous vehicles, automated hardware. Such as crop metrics.
- **Green Agriculture:** This technique uses control mechanism technique for environment parameters. To control environmental factors for a smart greenhouse, we use different sensors that contribute to environment parameters such as soil quality and soil type.

CHAPTER 11

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://en.wikipedia.org/wiki/OpenWeatherMap

https://www.researchgate.net/publication/334131097_A_Project_Report_On_IoT_based_SMART_FARMING_SYSTEM_CERTIFICATE_OF_APPROVAL_Countersigned_by

https://nodered.org/about/

http://www.ir.juit.ac.in:8080/jspui/bitstream/123456789/22755/1/Smart%20Farming%20using%20IoT.pdf

CHAPTER 12

APPENDIX

Source code

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "ox61h3" #replace the ORG ID
deviceType = "Smart_agriculture"#replace the Device type wi
deviceId = "motor"#replace Device ID
authMethod = "token"
authToken = "gu*tlqz@Z@9cEF5yDP" #Replace the authtoken

def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='motoron':
        print("MOTOR ON IS RECEIVED")

    elif cmd.data['command']=='motoroff':
        print("MOTOR OFF IS RECEIVED")
    '''

    if cmd.command == "setInterval":

        if 'interval' not in cmd.data:
            print("Error - command is missing required information: 'interval'")
        else:
            interval = cmd.data['interval']
    elif cmd.command == "print":
        if 'message' not in cmd.data:
            print("Error - command is missing required information: 'message'")
```

```
else:
        output=cmd.data['message']
       print(output)
       '''
       data = {"Command" : cmd.data['command']}
       success = deviceCli.publishEvent("event", "json", data, qos=0,
on_publish=myOnPublishCallback)
       if not success:
          print("Not connected to IoTF")

       myCommandCallback.has_been_called = True

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:

       '''
       T=50;
       H=32;
       ot=45

       data = {'d':{ 'Temperature' : Status, 'Humidity': H,'objTemp':ot }}
       #Send Temperature & Humidity to IBM Watson
       '''
       myCommandCallback.has_been_called = False
```

```
  Status = "Sensor is On"
    #cmd.data['command'] = "Rest"

 #Send Status to IBM Watson
      data= {'Status' : Status}
     #data2 = {'Command RECEIVED' : cmd.data['command'] }
     #print data
     def myOnPublishCallback():
        print (data, "to IBM Watson")
        #print (data2, "to IBM Watson")

     success = deviceCli.publishEvent("event", "json", data, qos=0, on_publish=myOnPublishCallback)
     if not success:
        print("Not connected to IoTF")
     time.sleep(1)

     deviceCli.commandCallback = myCommandCallback
     if myCommandCallback.has_been_called == True :
        print("call made")


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