INDEX

1 INTRODUCTION

1.1 OVERVIEW:

Smart agriculture is a farming management concept using modern technology to increase the quantity and quality of agriculture products. Farmers in the 21st century have access to GPS, soil scanning, data management, and Internet of Things. This project involves all these technologies in order to increase the yield. The method for using technologies in farming and cultivation requires knowledge of deep learning about the agricultural procedures and science. Many factors must be considered and investigated deeply while constructing a system that should make best cultivation process making agriculture system more effective and sustainable. In order to make agricultural system more precise that can be used by many farmers and can be applied in different context we need to be able to predict few things before hand.

This project based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop. The farmer can also get the real time weather forecasting data by using external platforms like Open Weather API.

1.2 PURPOSE:

Agriculture is the most labor intensive a important field of occupation. Migration of labor from rural to urban affects agriculture. The main purpose is meeting the food demands of largely populated country like India, in an organic method. Improving the farming methods using IoT enables in improving the yield and help the farmer in many ways.

The purpose can be defined in 3 ways

1)Increase output
2)Stabilize the growth
3)Sustain farming

These 3 factors will be immensely affected in a positive way through this project.

2 LITERATURE SURVEY

2.1 Existing Problem

While agriculture’s share in India’s economy has progressively declined to less than 15% due to the high growth rates of the industrial and services sectors, the sector’s importance in India’s economic and social fabric goes well beyond this indicator. First, nearly three-quarters of India’s families depend on rural incomes. Second, the majority of India’s poor (some 770 million people or about 70 percent) are found in rural areas. And third, India’s food security depends on producing cereal crops, as well as increasing its production of fruits, vegetables and milk to meet the demands of a growing population with rising incomes. To do so, a productive, competitive, diversified and sustainable agricultural sector will need to emerge at an accelerated pace.

Although India is the second largest irrigated country of the world after China, only one-third of the cropped area is under irrigation. Irrigation is the most important agricultural input in a tropical monsoon country like India where rainfall is uncertain, unreliable and erratic India cannot achieve sustained progress in agriculture unless and until more than half of the cropped area is brought under assured irrigation.

In spite of the large scale mechanization of agriculture in some parts of the country, most of the agricultural operations in larger parts are carried on by human hand using simple and conventional tools and implements like wooden plough, sickle, etc.

All these above stated problems can be eradicated with the use of below solution.

2.2 Proposed Solution

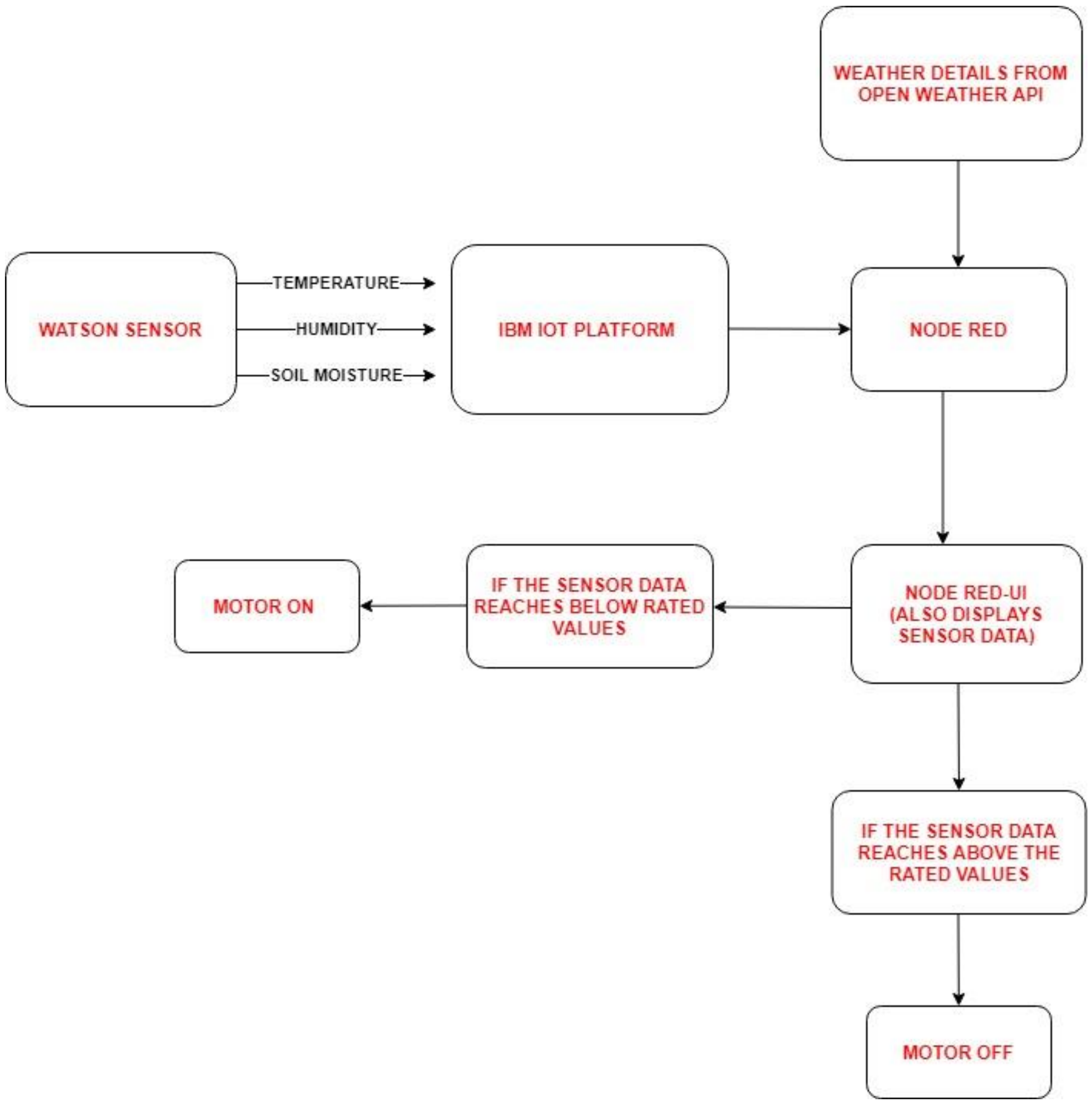
The above stated problems can be solved using IBM IoT Watson platform, Node Red, Open Weather API and cloud services. The technical details of the solution are given completely in the technical sector of this report.

- Smart Agriculture System based on IoT can monitor soil moisture and climatic conditions to grow and yield a good crop.

- The farmer can also get the real time weather forecasting data by using external platforms like Open Weather API.
- 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.
- Here we are using the Online IoT simulator for getting the Temperature, Humidity and Soil Moisture values.

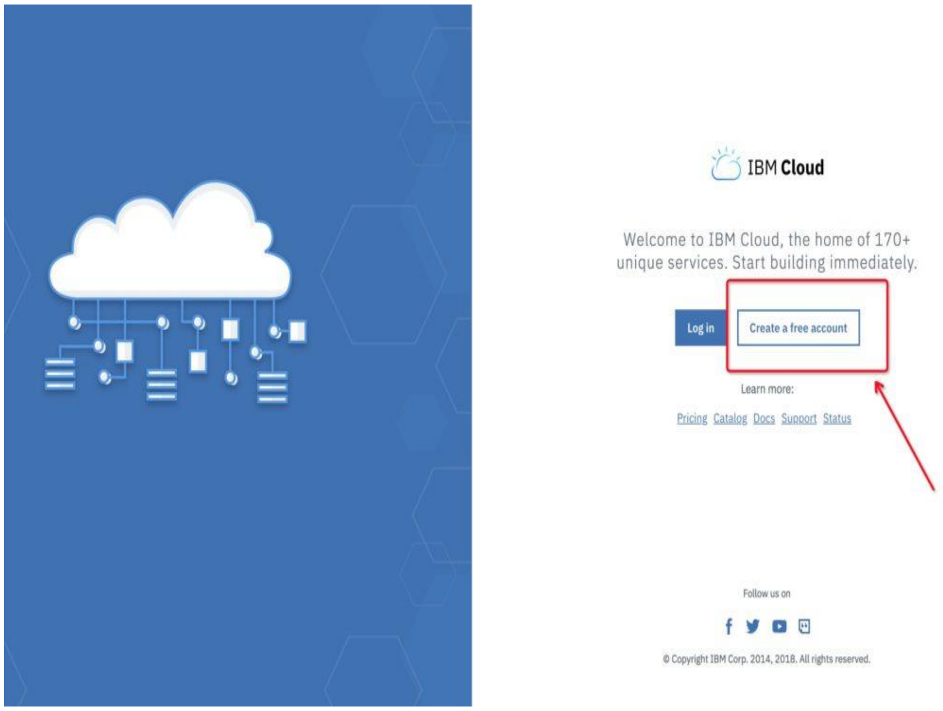
THEORETICAL ANALYSIS

3.1 Block Diagram


WEATHER DETAILS FROM OPEN WEATHER API
WATSON SENSOR
TEMPERATURE
HUMIDITY
SOIL MOISTURE
IBM IOT PLATFORM
NODE RED
MOTOR ON
IF THE SENSOR DATA REACHES BELOW RATED VALUES
NODE RED-UI (ALSO DISPLAYS SENSOR DATA)
IF THE SENSOR DATA REACHES ABOVE THE RATED VALUES
MOTOR OFF

3.2 Software Designing

➢ Creation of IBM cloud account



Welcome to IBM Cloud, the home of 170+ unique services. Start building immediately.

© Copyright IBM Corp. 2014, 2018. All rights reserved.

We can start building on the IBM Cloud using the account created. We can Build, deploy and scale apps for AI, IoT, data and mobile

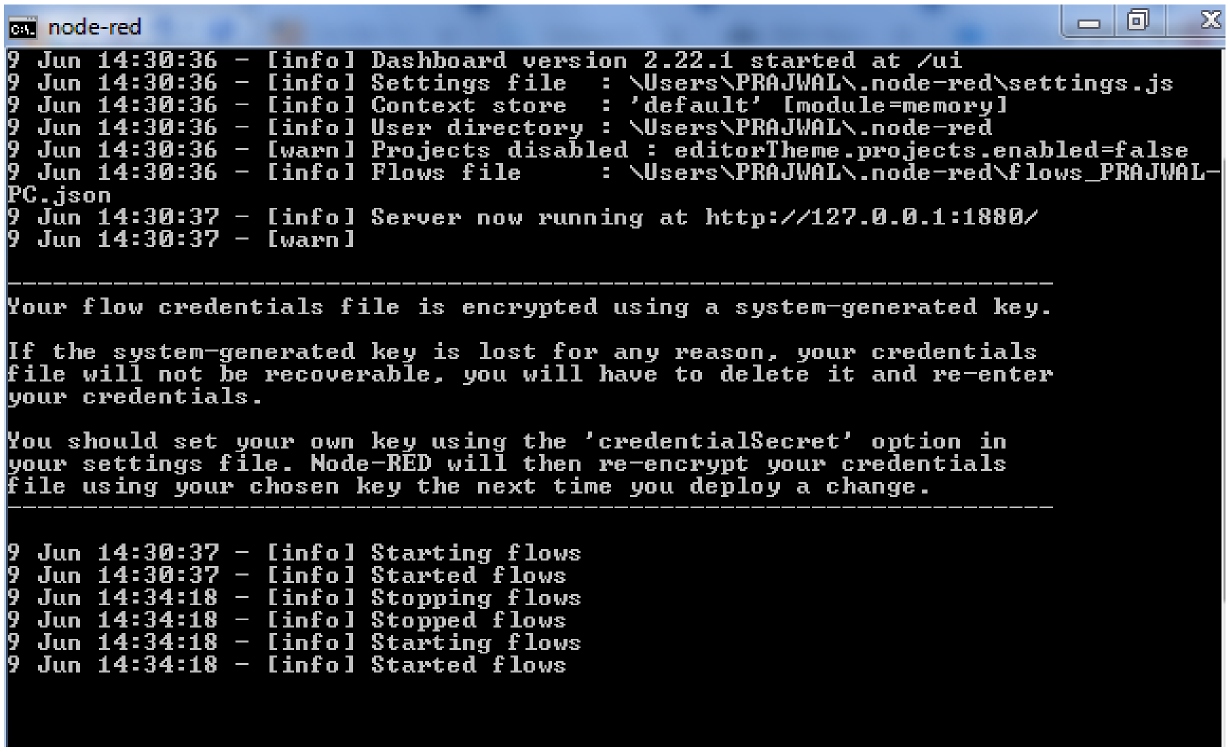
➢ Installing the node-red

Step 1: we need to install node-js from www.**nodejs.org** (recommended for most users version is sufficient).

Step 2: open the command prompt and follow the procedure given in below link.

https://nodered.org/docs/getting-started/local

Step 3: after successful installation when we open the command prompt and type node-red the output should be similar to

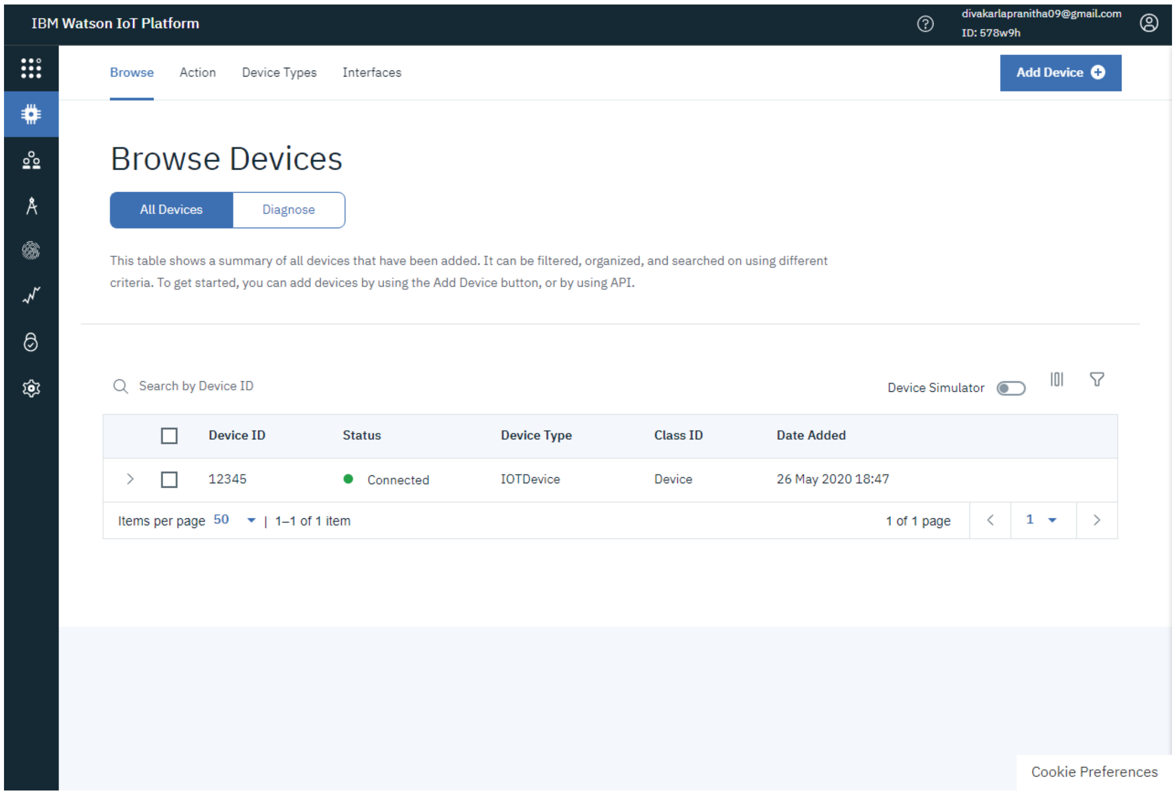


```
9 Jun 14:30:36 - [info] Dashboard version 2.22.1 started at /ui
9 Jun 14:30:36 - [info] Settings file  : \Users\PRAJWAL\.node-red\settings.js
9 Jun 14:30:36 - [info] Context store  : 'default' [module=memory]
9 Jun 14:30:36 - [info] User directory : \Users\PRAJWAL\.node-red
9 Jun 14:30:36 - [warn] Projects disabled : editorTheme.projects.enabled=false
9 Jun 14:30:36 - [info] Flows file     : \Users\PRAJWAL\.node-red\flows_PRAJWAL-
PC.json
9 Jun 14:30:37 - [info] Server now running at http://127.0.0.1:1880/
9 Jun 14:30:37 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

9 Jun 14:30:37 - [info] Starting flows
9 Jun 14:30:37 - [info] Started flows
9 Jun 14:34:18 - [info] Stopping flows
9 Jun 14:34:18 - [info] Stopped flows
9 Jun 14:34:18 - [info] Starting flows
9 Jun 14:34:18 - [info] Started flows
```

Step 4: using the URL displayed, we can get access to node red.

- Creating Device In IBM Iot Watson Platform

A device is created in the Watson platform in order to get access to the sensor values form the ibm iot sensor. Each device has unique device credentials that the user can create and use. These credentials are to be saved and used in order to connect to the sensor.



Browse Devices

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Device ID	Status	Device Type	Class ID	Date Added
12345	● Connected	IOTDevice	Device	26 May 2020 18:47

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

The below link is used as a reference.

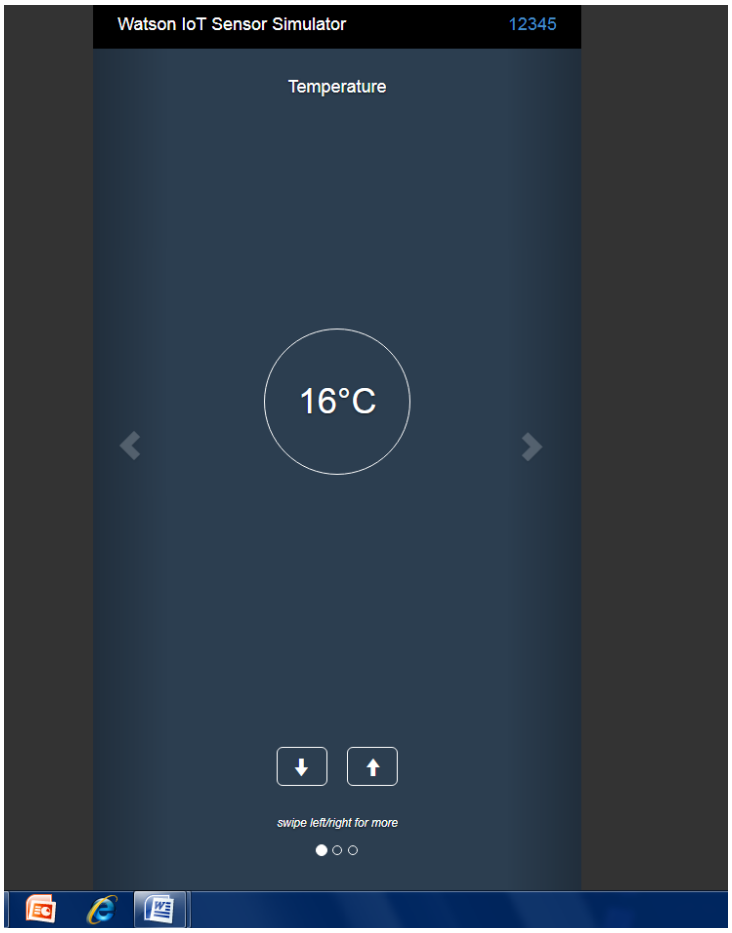
https://thesmartbridge.com/documents/pdf/IoT-Device-Creation.pdf

➢ Installing Python

https://www.python.org/downloads/

Use the above link and proceed as follows for the download.

➢ Connection Of IoT Simulator To Watson IoT Platform



https://watson-iot-sensor-simulator.mybluemix.net/

Use this link and fill in the device credentials. The output should be similar to the above picture.

- Open Weather API

  Open Weather Map is an online service that provides weather data. It provides current weather data, forecasts and historical data.

  In order to know the right time for irrigation we need to know the weather conditions near the farm, hence we need to configure the node red and open weather API system together.

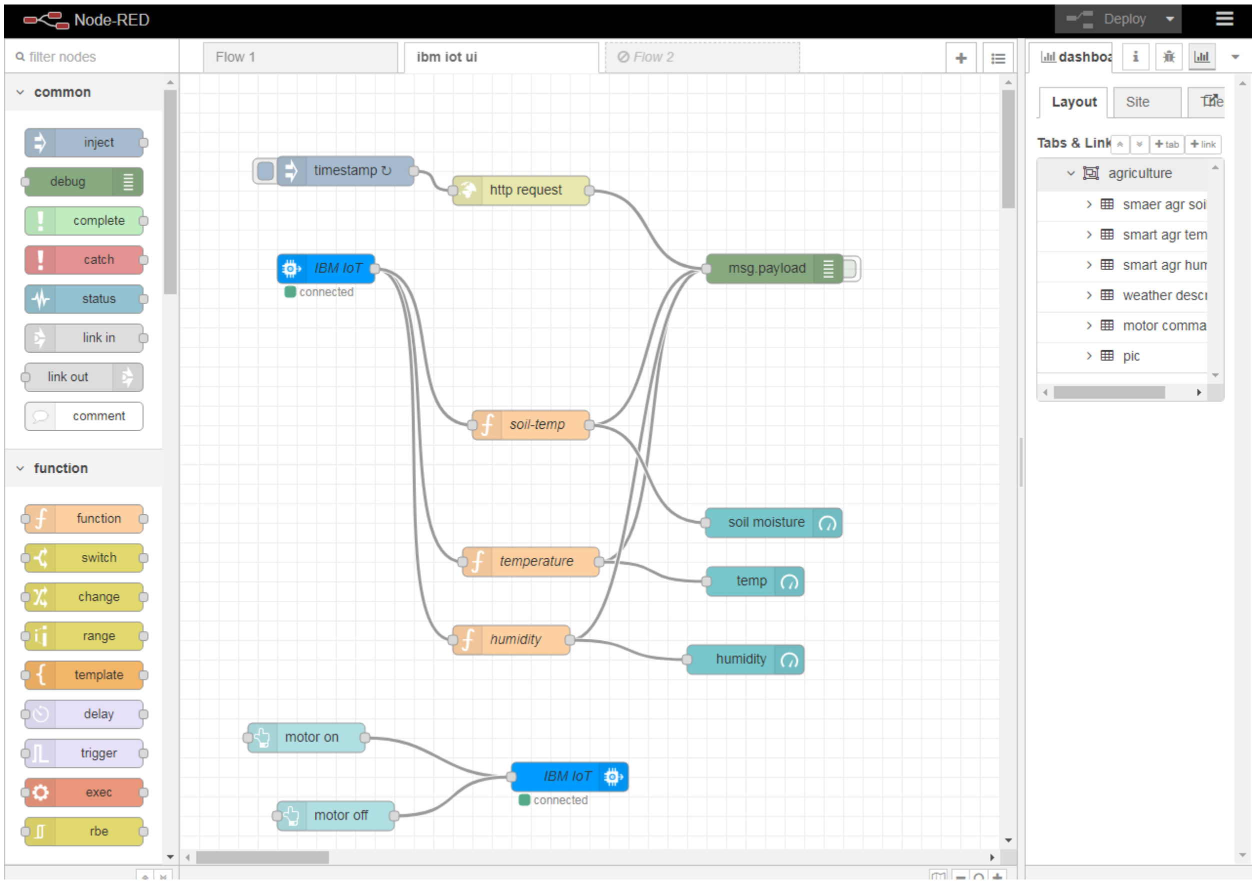
  We used the below reference link to obtain the open weather API and URL.

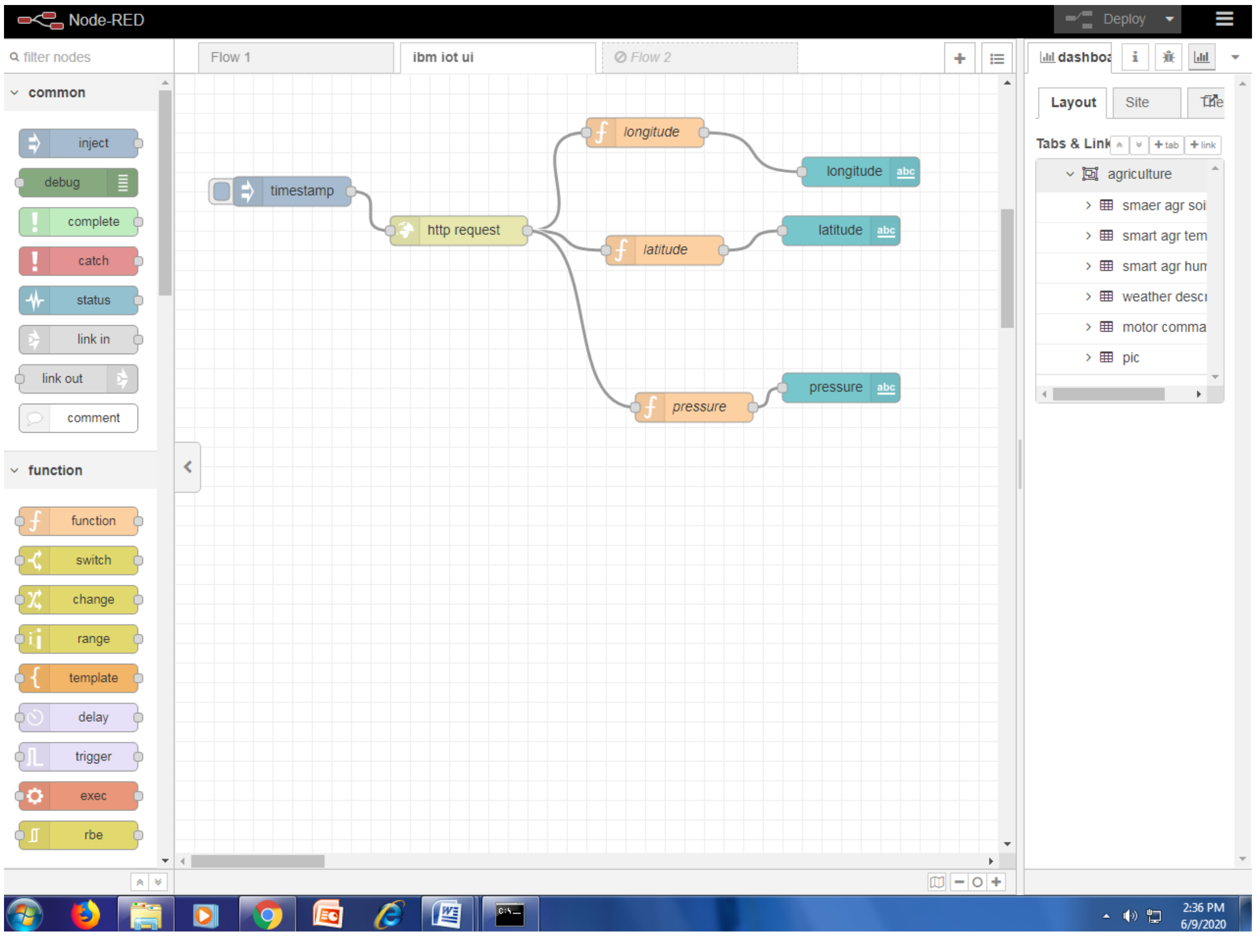
  https://smartinternz.com/assets/docs/Sending%20Http%20request%20to%20Open%20weather%20map%20website%20to%20get%20the%20weather%20forecast.pdf

➢ Configuring The Node-Red

After typing this URL http://127.0.0.1:1880/ in the browser the node red website appears. We have to use the flow present, to create the user interface.

We have to configure the node-red and create a user interface that displays the data from the sensor and weather details from open weather API.



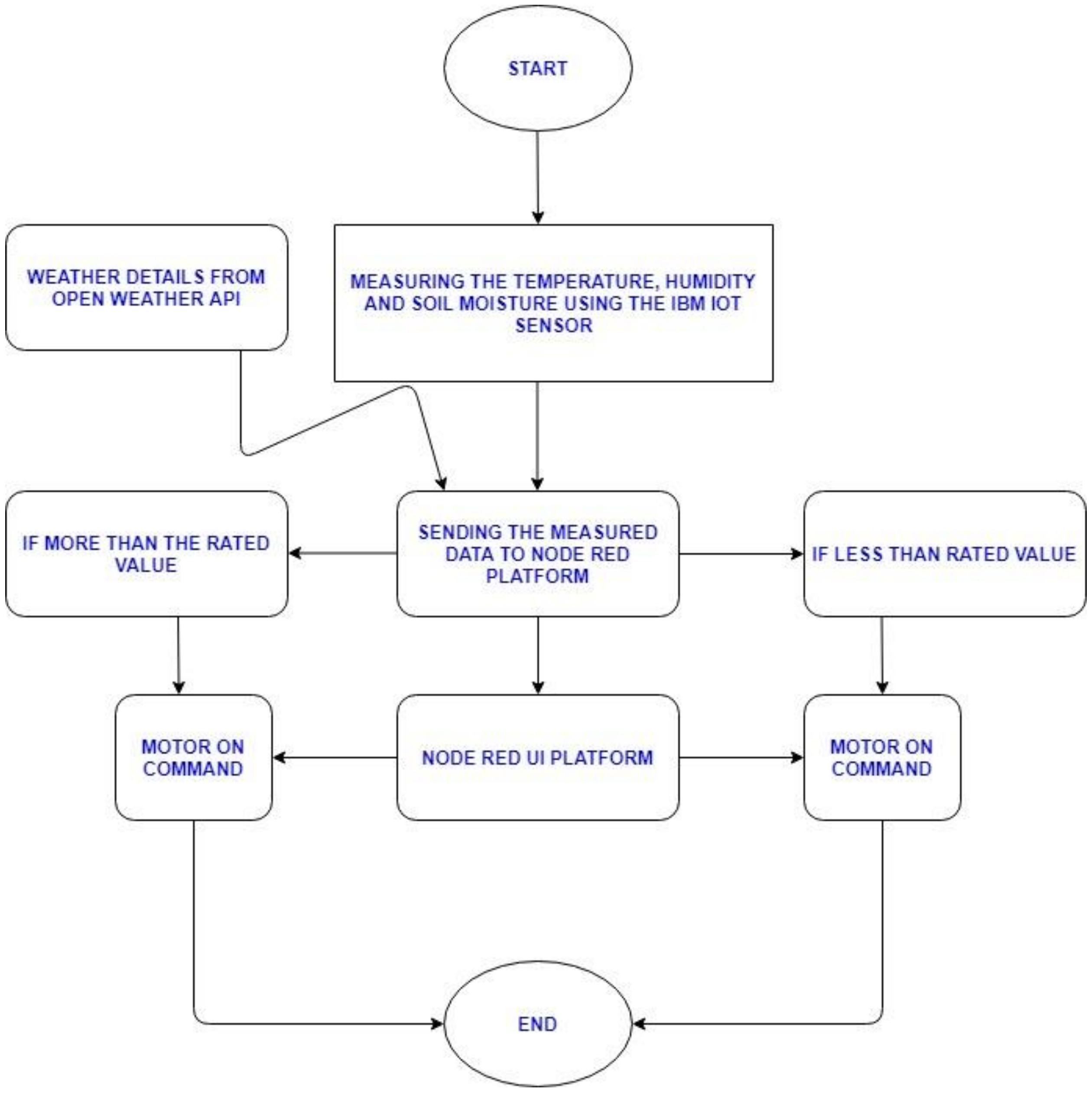


After installing the required nodes from manage palette and configuring them according the unique device credentials from IBM IoT Watson platform and connecting them accordingly we will have the screen similar to above. It is then deployed and the output is displayed in the debug section. The user interface window can be viewed by clicking the arrow in the dashboard section.

EXPERIMENTAL INVESTIGATIONS

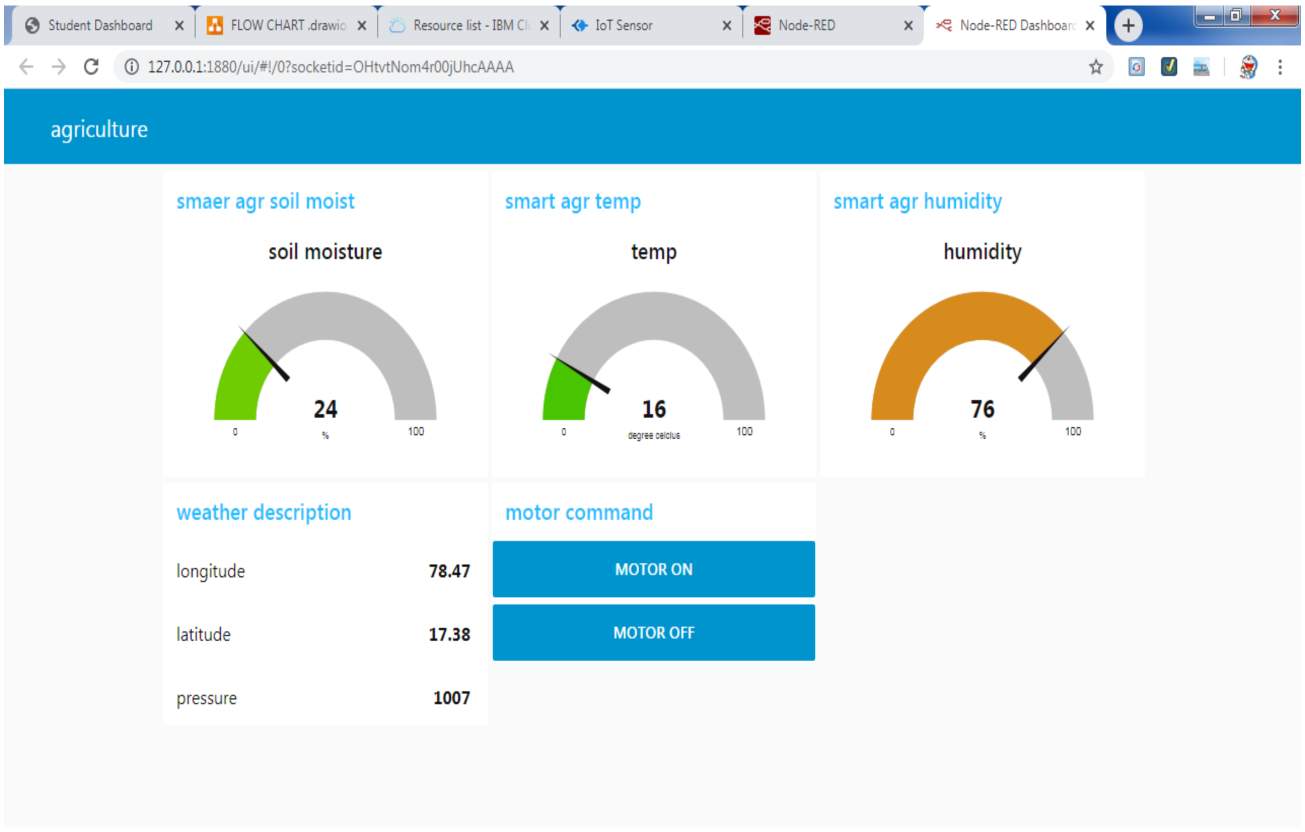
The experimental investigations usually include the outcomes of the project when it is first executed. Since the project is completely software, there were no issues executing it. The research topic, hypothesis, results , photographs and the designing were all carried out ethically without any obligations.

FLOW CHART


START
WEATHER DETAILS FROM OPEN WEATHER API
MEASURING THE TEMPERATURE, HUMIDITY AND SOIL MOISTURE USING THE IBM IOT SENSOR
IF MORE THAN THE RATED VALUE
SENDING THE MEASURED DATA TO NODE RED PLATFORM
IF LESS THAN RATED VALUE
MOTOR ON COMMAND
NODE RED UI PLATFORM
MOTOR ON COMMAND
END

RESULT

NODE RED UI WINDOW



longitude	78.47
latitude	17.38
pressure	1007

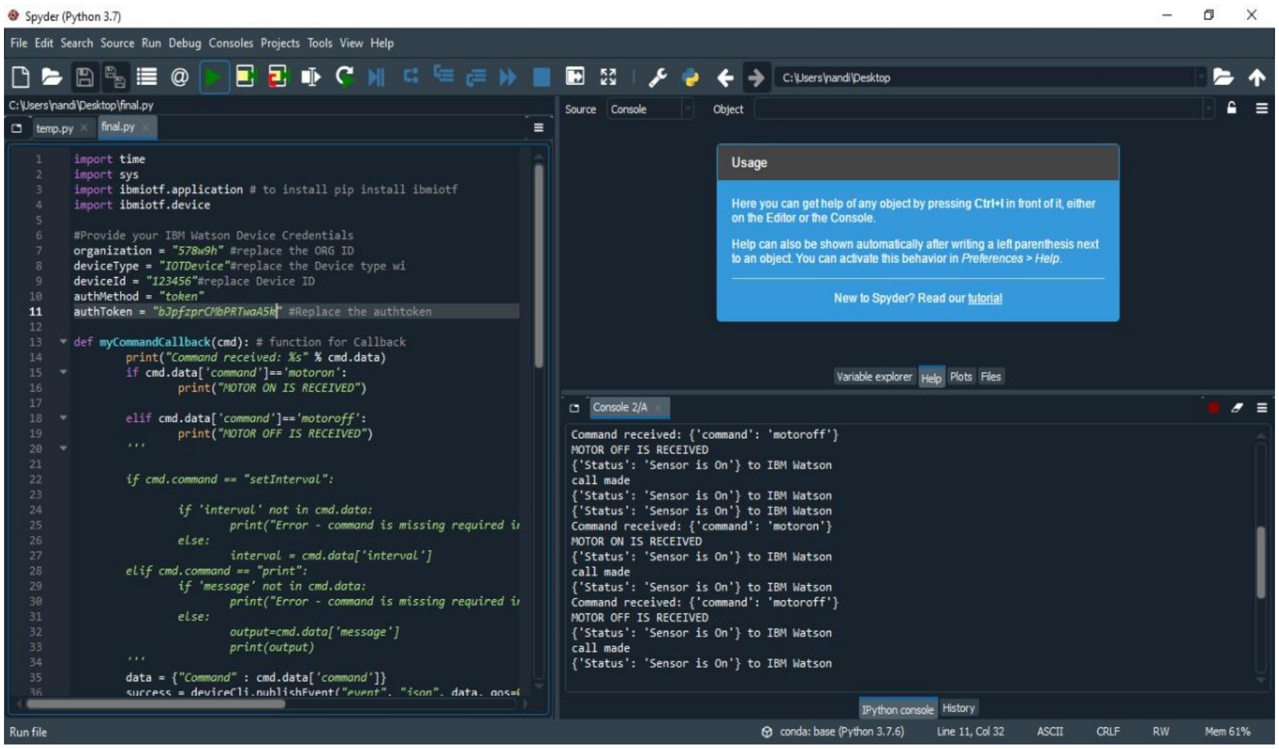
This is the result of the project that is the user interface through which the farer can give motor on and off commands.

The 3 gauge scales display the amount of soil moisture, temperature and humidity respectively that came from the IOT sensor.

The weather description tab displays the longitude and latitude of the location of the farm (location where we measure the weather conditions) and wind pressure.

The motor command tab is a button tab through which the farmer can give motor on and off commands.

MOTOR COMMAND OUTPUT



Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

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

#Provide your IBM Watson Device Credentials
organization = "578w9h" #replace the ORG ID
deviceType = "IOTDevice"#replace the Device type wi
deviceId = "123456"#replace Device ID
authMethod = "token"
authToken = "bJpfzprCMbPRTwaA5k" #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 i
                else:
                        interval = cmd.data['interval']
        elif cmd.command == "print":
                if 'message' not in cmd.data:
                        print("Error - command is missing required i
                else:
                        output=cmd.data['message']
                        print(output)
        '''
        data = {"Command" : cmd.data['command']}
        success = deviceCli.publishEvent("event", "json", data, qos=
```

```
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
{'Status': 'Sensor is On'} to IBM Watson
call made
{'Status': 'Sensor is On'} to IBM Watson
{'Status': 'Sensor is On'} to IBM Watson
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
{'Status': 'Sensor is On'} to IBM Watson
call made
{'Status': 'Sensor is On'} to IBM Watson
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
{'Status': 'Sensor is On'} to IBM Watson
call made
{'Status': 'Sensor is On'} to IBM Watson
```

The commands received from the node red UI are displayed in the output window of python. The output window is placed above.

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

One of the really good things about this branch of farming is that it allows for Soil Sensing. This aspect of smart farming gives room for you as a farmer to test your soil for information and also measure it for a wide range of important and nutritious constituents necessary in securing the good health of your farm produce. Smart agriculture makes use of AI to improve the process of wireless monitoring, regulation and data collection. With these inputs on your farm, all thanks to smart farming, you can be sure of high-quality crop production and delivery. Smart farming systems reduce waste, improve productivity, and enable management of a greater number of resources through remote sensing

DISADVANTAGES

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

As pointed out earlier, smart farming makes use of high techs that require technical skill and precision to make it a success. It requires an understanding of robotics and ICT. 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.

APPLICATIONS

Today, the combination of smart irrigation and control being linked to local sensors, as well as sensing for pH and other environmental conditions, including isolation and local temperature, can stave off many issues that traditionally had been accounted for by "walking the field." Remote monitoring through smart farming systems enables production yields to increase because farmers have more time to attend to their farm's real issues:

applying their expertise to solving problems with pests, watering in any location, amending soil conditions -- all through the use of sensing and automation. There are numerous examples of leveraging IoT technologies in agriculture from versatile data analytics and management systems to futuristic robot pollinators.

CONCLUSION

Markets will grow and collapse, disruptive business models will emerge or die, but people will always need to eat and drink. For this reason, the development of such areas as food and agriculture will always be a priority, especially given the dynamics we observe in the world today. Therefore, IoT used in agriculture has a big promising future as a driving force of the efficiency, sustainability and scalability in this industry. The high efficiency of integrated agriculture production systems delivers socio-economic and ecological benefits that benefit farmers as well the whole society. The sustainable intensification of integrated agriculture production systems requires: a better understanding of the impacts of changes in climate and climate variability on these systems; the generation and sharing of local and global knowledge, experiences and practices; capacity development through research and development, dialogue and dissemination of information; and support and coordination of policies, particularly policies that can provide incentives and create enabling institutions.

FUTURE SCOPE

Smart Farming and IoT-driven agriculture are paving the way for what can be called a Third Green Revolution.

Following the plant breeding and genetics revolutions, the Third Green Revolution is taking over agriculture. That revolution draws upon the combined application of data-driven analytics technologies, such as precision farming equipment, IoT, “big data” analytics, Unmanned Aerial Vehicles (UAVs or drones), robotics, *etc*.

In the future this smart farming revolution depicts, pesticide and fertilizer use will drop while overall efficiency will rise. IoT technologies will enable better food traceability, which in turn will lead to increased food safety. It will also be beneficial for the environment, through, for example, more efficient use of water, or optimization of treatments and inputs.

Therefore, smart farming has a real potential to deliver a more productive and sustainable form of agricultural production, based on a more precise and resource-efficient approach. New farms will finally realize the eternal dream of mankind. It’ll feed our population, which may explode to 9.6 billion by 2050.

BIBLIOGRAPHY

Few references from www.google.com were taken.

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 = "578w9h" #replace the ORG ID
deviceType = "IOTDevice"#replace the Device type wi
deviceId = "123456"#replace Device ID
authMethod = "token"
authToken = " bJpfzprCMbPRTwaA5k" #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()
```