PROJECT REPORT

1. INTRODUCTION:

1.1 Overview:

The main objectives of this project is to propose IoT based Smart Farming System which will enable farmers to have live data of soil moisture and environment temperature at very low cost so that live monitoring can be done. To avoid the problems faced by the farmers we have developed an IOT based app which will help predict weather conditions of that specific area and also consist of a motor ON/OFF button so that the farmer can operate the motor through this IOT based app. Monitoring crops and weather forecasting will be an easy task for farmers.

1.2 Purpose:

Farmers in the 21st century have access to GPS, soil scanning, data management, and Internet of Things technologies. By precisely measuring variations within a field and adapting the strategy accordingly, farmers can greatly increase the effectiveness of pesticides and fertilizers, and use them more selectively. In addition, we can also manage the amount of water supplied to the crops since we are detecting the soil moisture content. Due to this, the chances of crops getting damaged is reduced also sustainability is maintained. Similarly, using Smart Farming techniques, farmers can better monitor the needs of individual animals and adjust their nutrition correspondingly, thereby preventing disease and enhancing herd health.

2. LITERATURE SURVEY:

2.1 Existing problem:

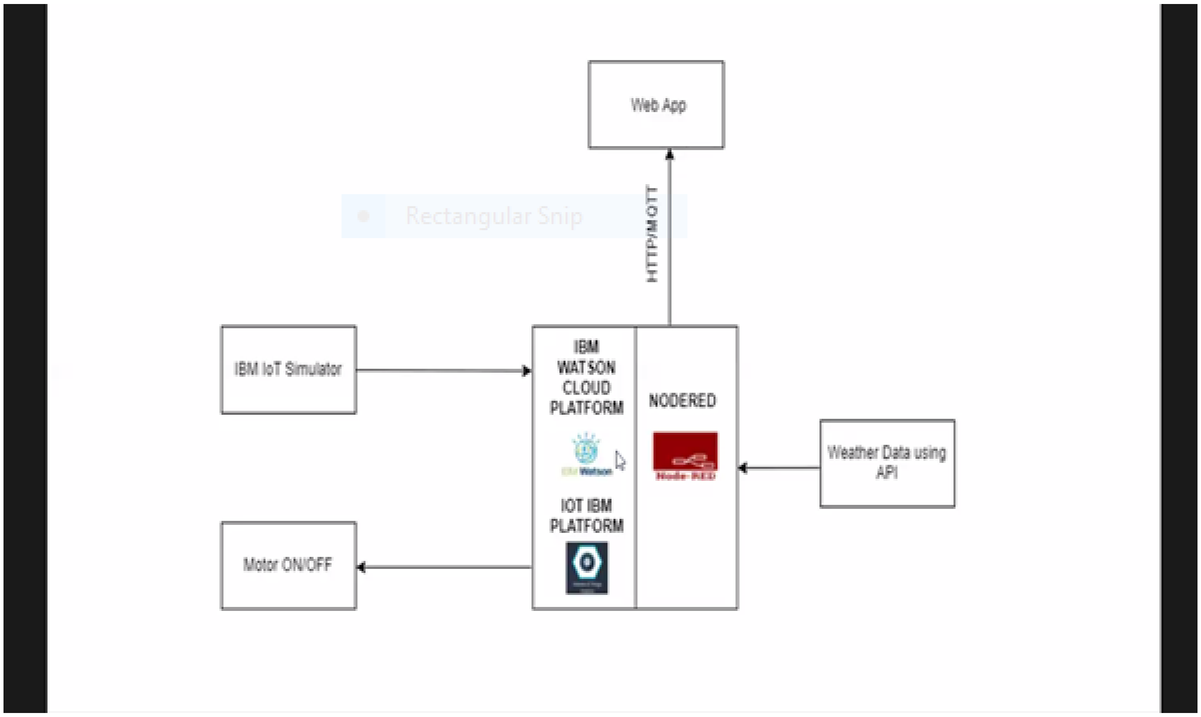
- Uncertain, unreliable & erratic rainfall leads to water wastage due to lack of modern equipment.
- Inefficient agricultural methods due to absence of technological interface leads to large manpower & high cost.
- Absence of proper crop monitoring process doesn’t leads to enhancement in crop output efficiency.
- Since, water management is a major issue around the globe, inappropriate watering to the crops leads to one of sustainable issues.

2.2 Proposed Solution:

To avoid the problems faced by the farmers we have developed an IOT based web app which will help us provide information of weather conditions. The app will also provide the data from the sensor attached in the farm. Through this information the farmer can monitor his crops and farm remotely, depending upon the soil moisture content he can water his plant the way he want. In addition, we have added a switch to control electric motor, so that the farmer can operate the motor remotely. Monitoring the crops and weather forecasting will be an easy task for farmer. We are using IBM cloud to generate the device and then we interfaced it with IBM IOT sensor. For developing the app we have used Node-Red and by adding different nodes we took the inputs from the IBM device, also showed the status of the motor as well as displayed the weather details.

3. THEORITICAL ANALYSIS:

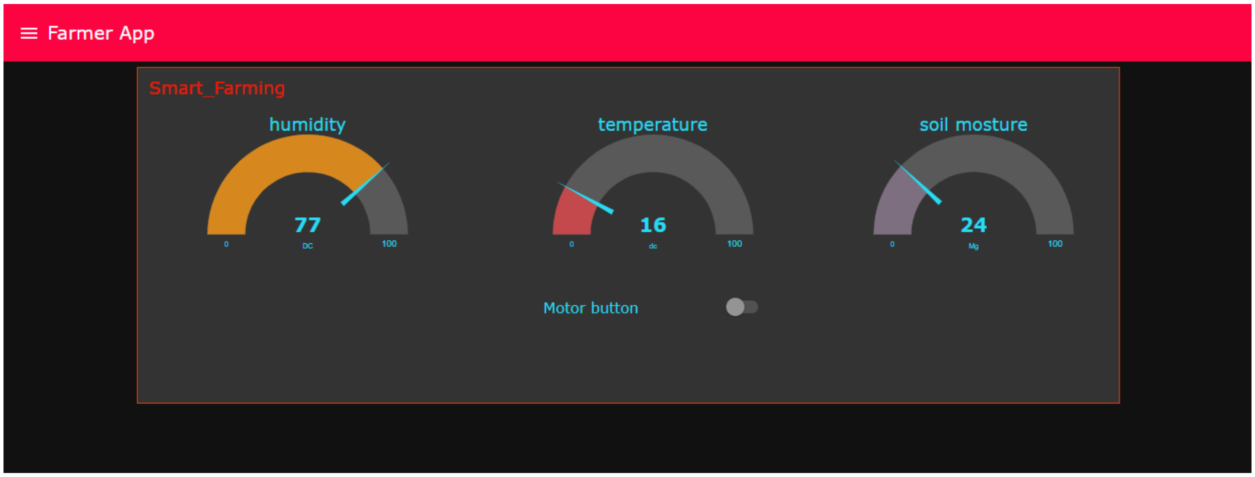
3.1 Block diagram:

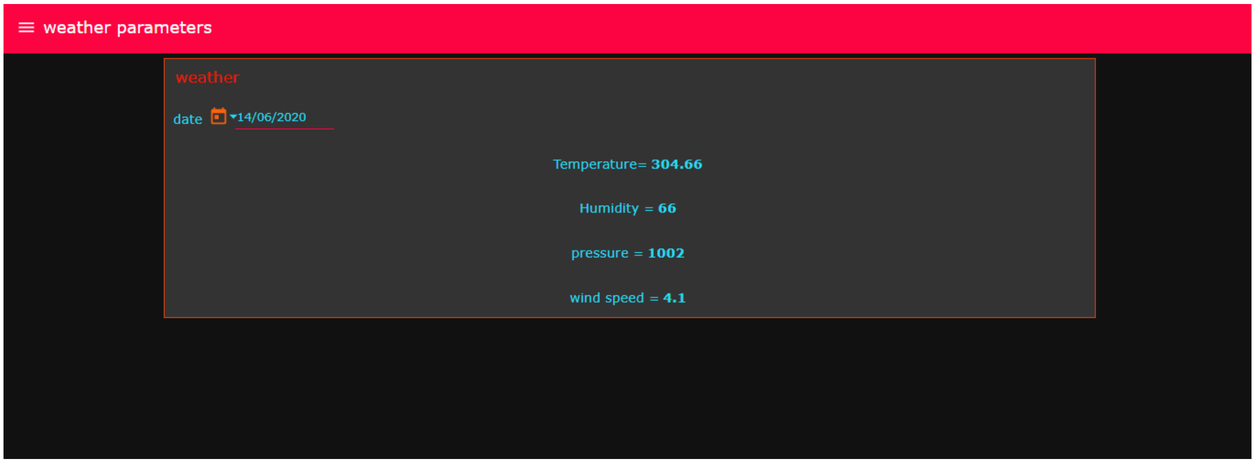


3.2 Hardware / Software designing:

- We have generated the IBM device using IBM cloud platform and also created Api keys to connect with Node red.
- Created a board consisting cards which displays experimental values using gauge, charts etc.
- Connected the Watson IOT Sensor Simulator with the IBM device.
- Created a web App using node-red consisting of different nodes like INJECT, DEBUG etc.
- Took the weather forecasting details of a particular area through Open Weather Api and then displayed its status using the 'http' request, Change and debug Node.

4. Result:




weather parameters

weather

date 14/06/2020

Temperature= **304.66**

Humidity = **66**

pressure = **1002**

wind speed = **4.1**

The above image is of node-red dashboard where it is showing the sensor data i.e temperature, humidity and soil moisture content in the form of gauge. I have also added two buttons (Motor ON/OFF) through which I can control the electric motor remotely. The next figure shows me the weather forecasting details of Mumbai, IN.

5. Advantages/Disadvantages:

5.1 Advantages:

- Ease of workflow in agricultural sector.
- Sustainable Development
- Bettter productivity

5.2 Disadvantages:

- Cannot rely on weather conditions, as they are predicted.
- Needs Internet connectivity.

6. Applications:

Using this system in agricultural sector will not only helpin productivity but also economic growth. It will help decrease manual work.

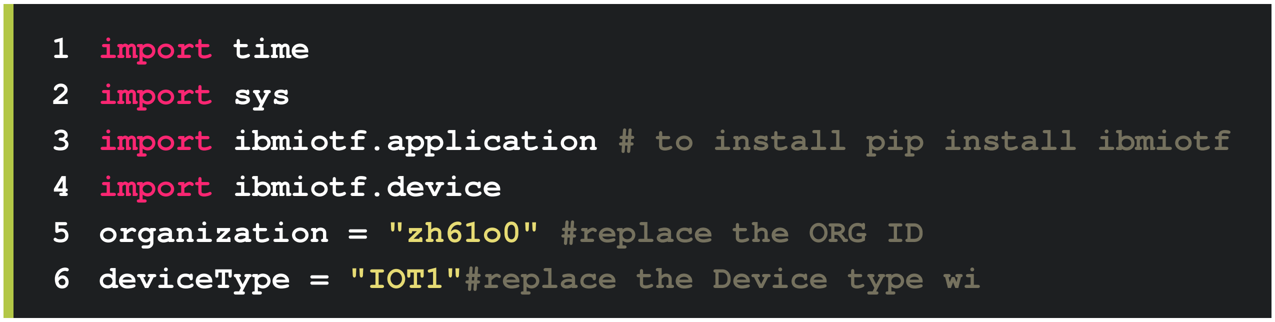
7. Conclusion:

Smart agriculture is an emerging concept, because IOT sensors are capable of providing information about agriculture fields and then act upon based on the user input.The system aims at making use of evolving technology i.e. IOT and smart agriculture using automation. Monitoring environmental conditions is the major factor to improve yield of the efficient crops. The feature of this project includes development of a system which can monitor temperature, humidity, moisture. Smart farming and precision agriculture involve the integration of advanced technologies into existing farming practices in order to increase production efficiency and the quality of agricultural products. As an added benefit, they also improve the quality of life for farm workers by reducing heavy labor and tedious tasks.

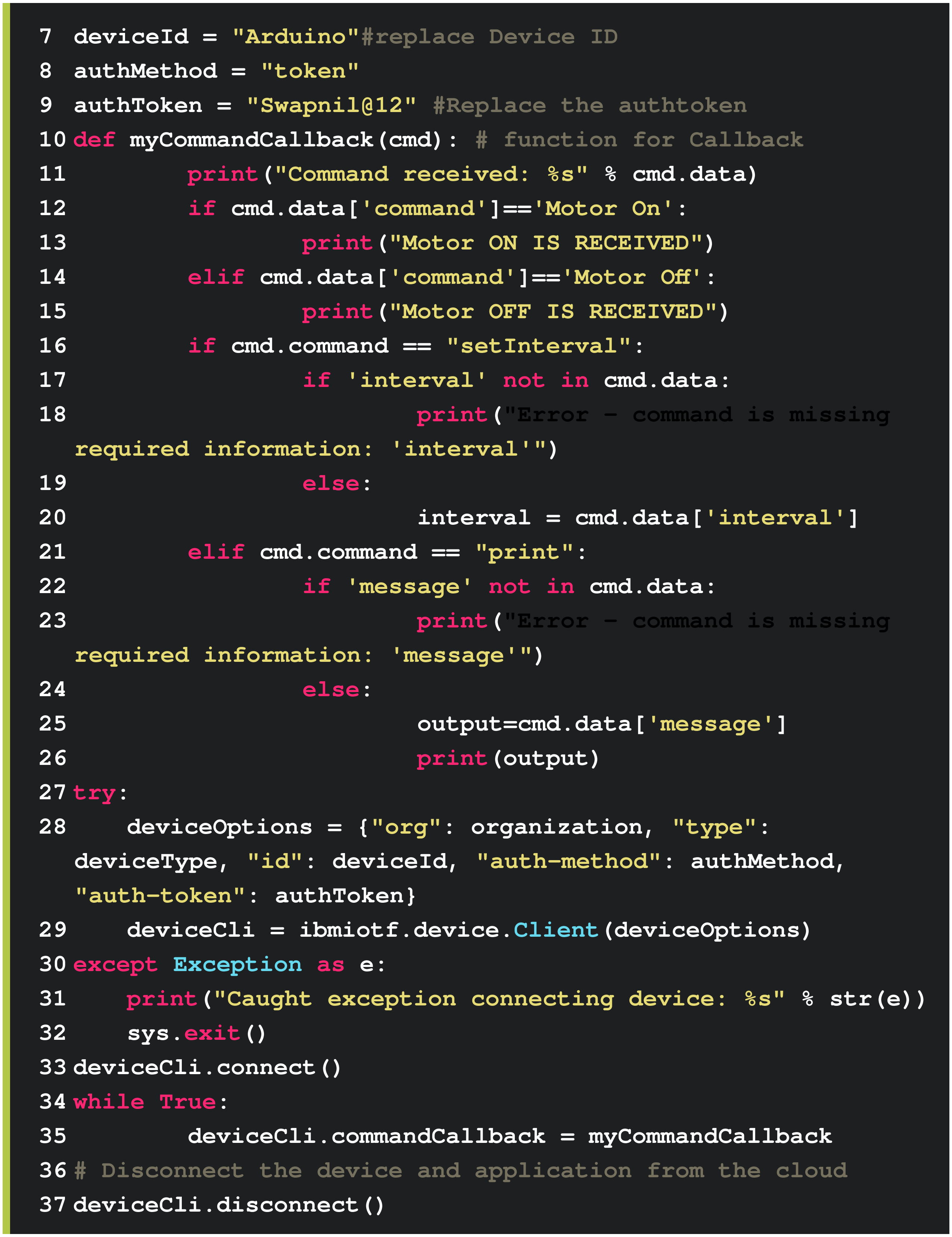
8. Future Scope:

A population of this magnitude brings a lot of challenges, food production chief among them. Weeding and pest control are both critical aspects of plant maintenance and tasks that are perfect for autonomous robots. We can also implement machines for weeding and pest control purpose. Seeding improved on that with seeding machines, which can cover more ground much faster than a human. Even the movement of animals can be monitored which may destroy the crops in agricultural field and in case of any discrepancy send a SMS notification as well.

9. Appendix:



```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device
organization = "zh61o0" #replace the ORG ID
deviceType = "IOT1"#replace the Device type wi
```



```python
deviceId = "Arduino"#replace Device ID
authMethod = "token"
authToken = "Swapnil@12" #Replace the authtoken
def myCommandCallback(cmd): # function for Callback
        print("Command received: %s" % cmd.data)
        if cmd.data['command']=='Motor On':
                print("Motor ON IS RECEIVED")
        elif cmd.data['command']=='Motor Off':
                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)
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:
        deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```