IOT Enabled Smart Farming Application

NALAIYA THIRAN PROJECT BASED LEARNING

on

~~PROFESSIONAL READINESS FOR INNOVATION,~~
EMPLOYABILITY AND ENTREPRENEURSHIP

Project Report Submitted by

911519205002– ABUASLAM A

911519205003– AKASH N

911519205009 - FOUZAL HITHYA

911519205011-JOTHIKA

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

MOHAMED SATHAK ENGINEERING COLLEGE

KILAKARAI-623 806.

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER – 2022

CONTENTS

1. Introduction

1.1 Project Overview
1.2 Purpose

2.Litreature Survey

2.1 Existing Problem

2.2 References

2.3 Problem Statement Definition

3. Ideation & Proposed Solution

3.1 Prepare Empathy Map

3.2 Ideation

3.3 Proposed Solution

3.3 Proposed Solution-fit

4. Requirement Analysis

4.1 Functional Requirement

4.2 Non- Functional Requirement

5. Project Design

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. Project Planning & Scheduling

6.1 Sprint Delivery Planning & Estimation

6.2 Sprint Delivery Schedule

7. Coding and Solution

7.1 Feature - 1

7.2 Feature – 2

7.3 Database Schema

8.Testing

8.1 Test Cases

8.2 User Acceptance Testing

9,Results

9.1 Performance Metrics

10,Advantages & Disadvantages

11.Conclusion

12. Future Scope

13. Appendix

Source code

Github & Project Demo Link

1. Introduction

1.1 Project Overview

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.

1.2 Purpose

Increasing control over production leads to **better cost management and waste reduction**. The ability to trace anomalies in crop growth or livestock health, for instance, helps eliminate the risk of losing yields. Additionally, automation boosts efficiency. Smart farming **reduces the ecological footprint of farming**. Minimized or site-specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases.

2. Literature Survey

2.1 Existing Problem

IoT based Smart Farming **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. Climate plays a very critical role for farming. And having improper knowledge about climate heavily deteriorates the quantity and quality of the crop production. Precision Agriculture/Precision Farming is one of the most famous applications of IoT in Agriculture. It makes the farming practice more precise and controlled by realizing smart farming applications such as livestock monitoring, vehicle tracking, field observation, and inventory monitoring. To make our greenhouses smart, IoT has enabled weather stations to automatically adjust the climate conditions according to a particular set of instructions. Adoption of IoT in Greenhouses has eliminated the human intervention, thus making entire process cost-effective and increasing accuracy at the same time.

2.2 References

1**,** Sustainable agriculture by the Internet of Things – A practitioner's approach to monitor sustainability progress. 2022, Computers and Electronics in Agriculture.

2, The Interplay between the Internet of Things and agriculture: A metric analysis and research agenda. 2022, International Journal of Intelligent Networks.

3, Agriculture 4.0 and its Barriers in the Agricultural Production Chain Development in Southern Brazil. 2022, SSRN

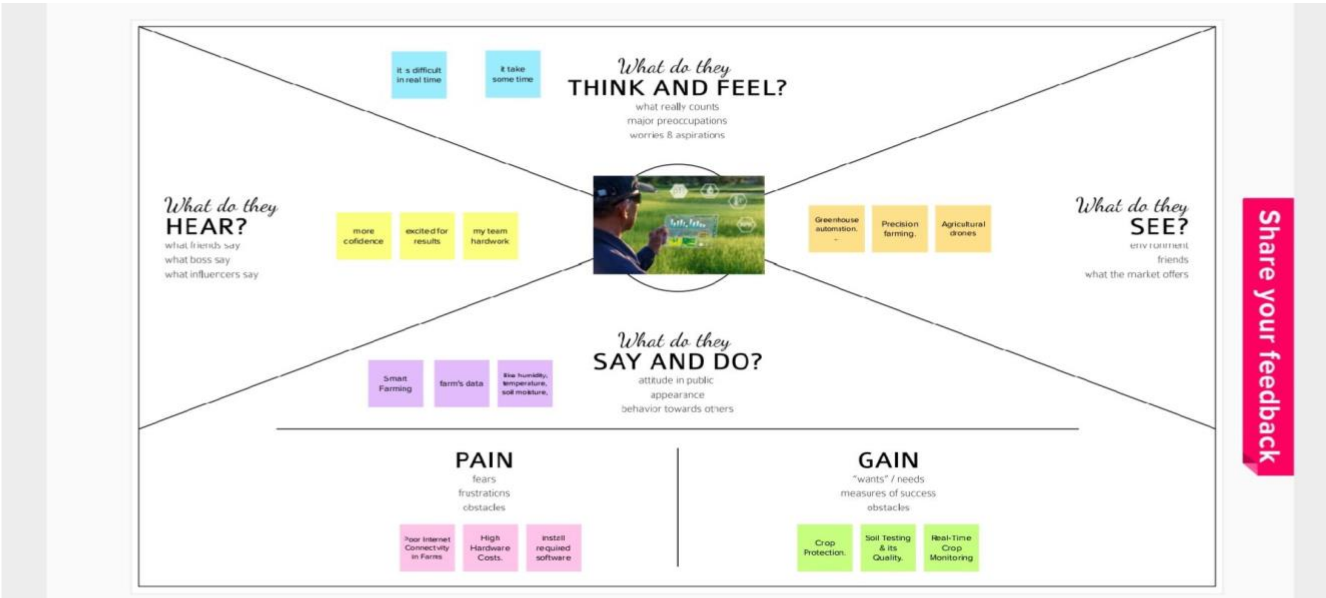
4, IoT based Agriculture (IoTA): Architecture, Cyber Attack, Cyber Crime and Digital Forensics Challenges. 2022, Research Square.

2.3 Problem Statement Solution

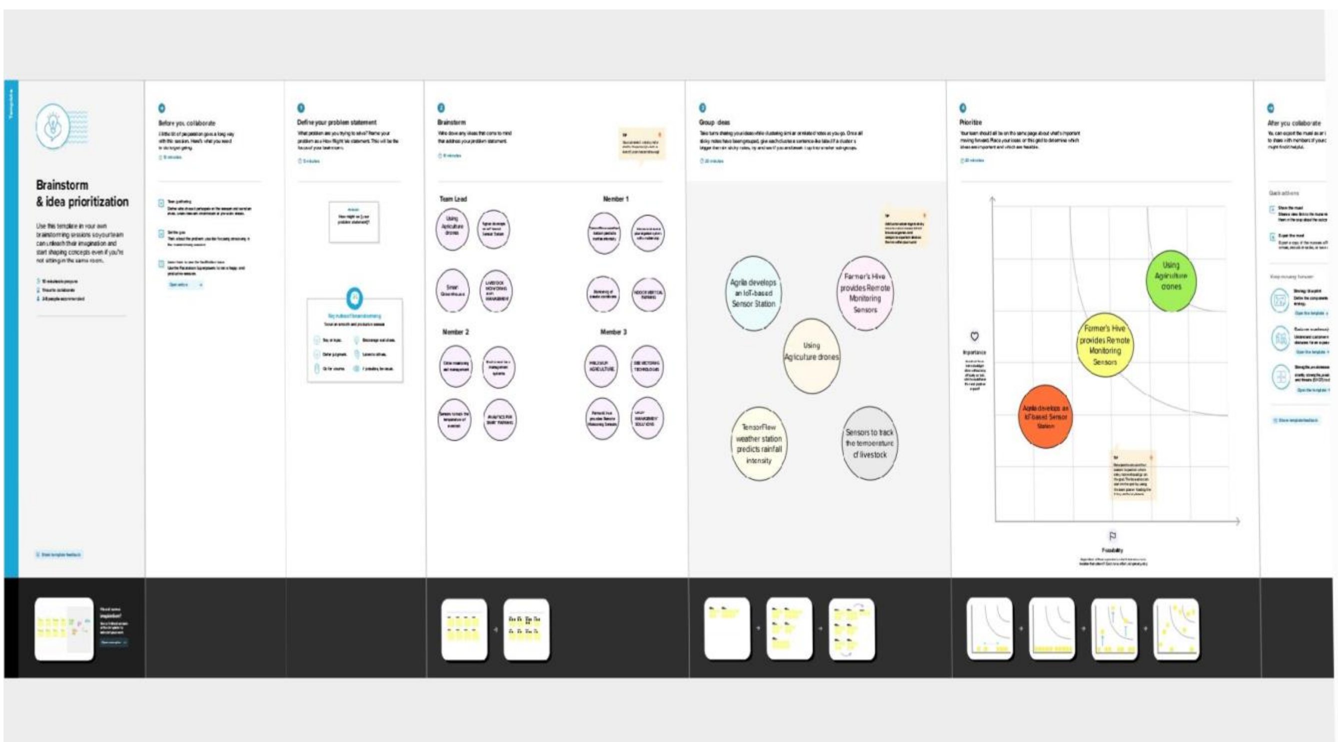
The traditional agriculture and allied sector cannot meet the requirements of modern agriculture which requires high-yield, high quality and efficient output. Thus, it is very important to turn towards modernization of existing methods and using the information technology and data over a certain period to predict the best possible productivity and crop suitable on the very particular land. The adoptions of access to high-speed internet, mobile devices, and reliable, low-cost satellites (for imagery and positioning) are few key technologies characterizing the precisionprecisionagriculture0 agriculture trend. Precision agriculture is one of the most famous applications of IoT in the agricultural sector and numerous organizations are leveraging this technique around the world. Some products and services in use are VRI optimization, soil moisture probes, virtual optimizer PRO, and so on. VRI (Variable Rate Irrigation) optimization maximizes profitability on irrigated crop fields with topography or soil variability, improve yields, and increases water use efficiency.IoT has been making deep inroads into sectors such as manufacturing, health-care and automotive. When it comes to food production, transport and storage, it offers a breadth of options that can improve India's per capita food availability. Sensors that offer information on soil nutrient status, pest infestation, moisture conditions etc. which can be used to improve crop yields over time. Some of the sample problem statements related to Agriculture & allied sectors where IoT application will be beneficial are given below.

3. Ideation & Proposed Solution

3.1 Prepare Empathy Map



3.2 Ideation



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Our project will be give the problem statement in Smart farming application using IOT. History-based soil health parameters like soil moisture, pHlevel, temperature etc.
2.	Idea / Solution description	The most frequently used applications of IoT in agriculture are drones for monitoring fields and spraying crops, health assessment of livestock and irrigation.
3.	Novelty / Uniqueness	Smart farming, which involves the application of sensors and automated irrigation practices, can help monitor agricultural land, temperature, soil moisture, etc. This would enable farmers to monitor crops from anywhere
4.	Social Impact / Customer Satisfaction	Increased production: the optimisation of all the processes related to agriculture and livestock-rearing increases production rates. Water saving: weather forecasts and sensors that measure soil moisture mean watering only when necessary and for the right length of time
5.	Business Model (Revenue Model)	Climate-smart agriculture is a pathway towards development and food security built on three pillars: increasing productivity and incomes, enhancing resilience of livelihoods and ecosystems and reducing and removing greenhouse gas emissions from the atmosphere
6.	Scalability of the Solution	Smart Farming systems uses modern technology to increase the quantity and quality of agricultural products. Livestock tracking and Geo fencing. Smart logistics and warehousing. Smart pest management. Smart Greenhouses

3.4 Proposed Solution Fit

Project Title: Smart Farmer – IoT Enabled Smart Farming Application

Project Design Phase-I - Solution Fit

Team ID: PNT2022TMID33110

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

The customer for this product is a farmer who grows crops. Our goal is to help them, monitor field parameters remotely. This product saves agriculture from extinction.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

Using a large number of sensors is difficult. An unlimited or continuous internet connection is required for success.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper

The irrigation process is automated using IoT. Meteorological data and field parameters were collected and processed to automate the irrigation process. Disadvantages are efficiency only over short distances, and difficult data storage.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides

The purpose of this product is to use sensors to acquire various field parameters and process them using a central processing system. The cloud is used to store and transmit data using IoT. The Weather API is used to help farmers make decisions. Farmers can make decisions through mobile applications.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

Frequent changes and unpredictable weather and climate made it difficult for farmers to engage in agriculture. These factors play an important role in deciding whether to water your plants. Fields are difficult to monitor when the farmer is not at the field, leading to crop damage.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Use a proper drainage system to overcome the effects of excess water from heavy rain. Use of hybrid plants that are resistant to pests.

Focus on J&P, tap into BE, understand RC

4.Requirement Analysis

4.1 Functional Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	As a user Registration through Gmail
FR-2	User Confirmation	As a user Confirmation via Email then generate the Confirmation via OTP
FR-3	Log in to system	Once confirmation message received after login the system and Check Credentials
FR-4	Check Credentials	Once check the credentials after go to the Manage modules.
FR-5	Manage modules	In this manage modules described the below functions like Manage System Admins Manage Roles of User Manage User permission and etc..
FR-6	Logout	Then check Temperature, humidity and moisture after then logout or exist the application.

4.2 Non- Functional Requirements

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

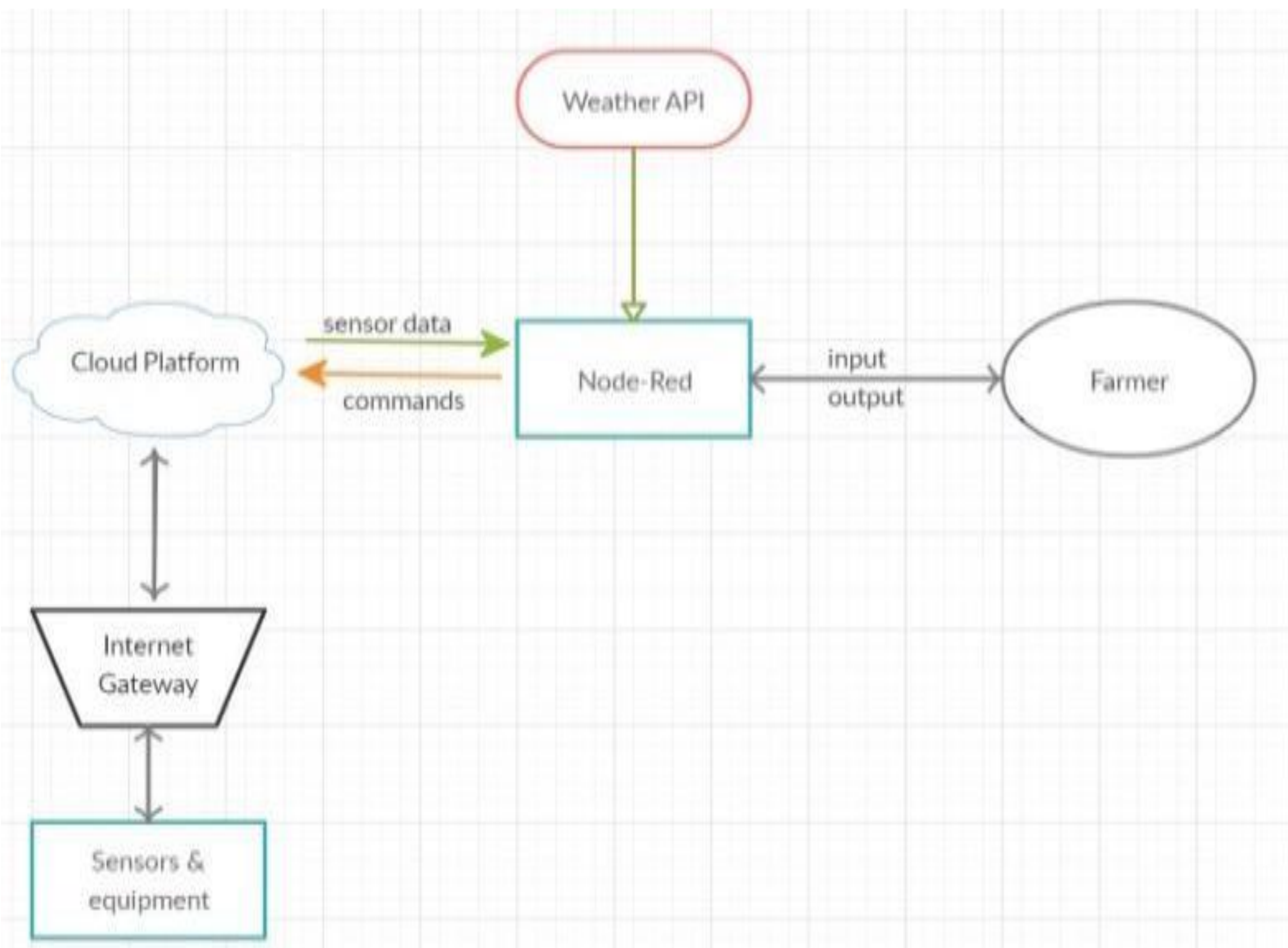
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability includes easy learn ability, efficiency in use, remember ability, lack of errors in operation and subjective pleasure.
NFR-2	**Security**	Sensitive and private data must be protected from their production until the decision-making and storage stages.
NFR-3	**Reliability**	The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages.
NFR-4	**Performance**	the idea of implementing integrated sensors with sensing soil and environmental or ambient parameters in farming will be more efficient for overall monitoring.
NFR-5	**Availability**	Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.
NFR-6	**Scalability**	scalability is a major concern for IoT platforms. It has been shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousand.

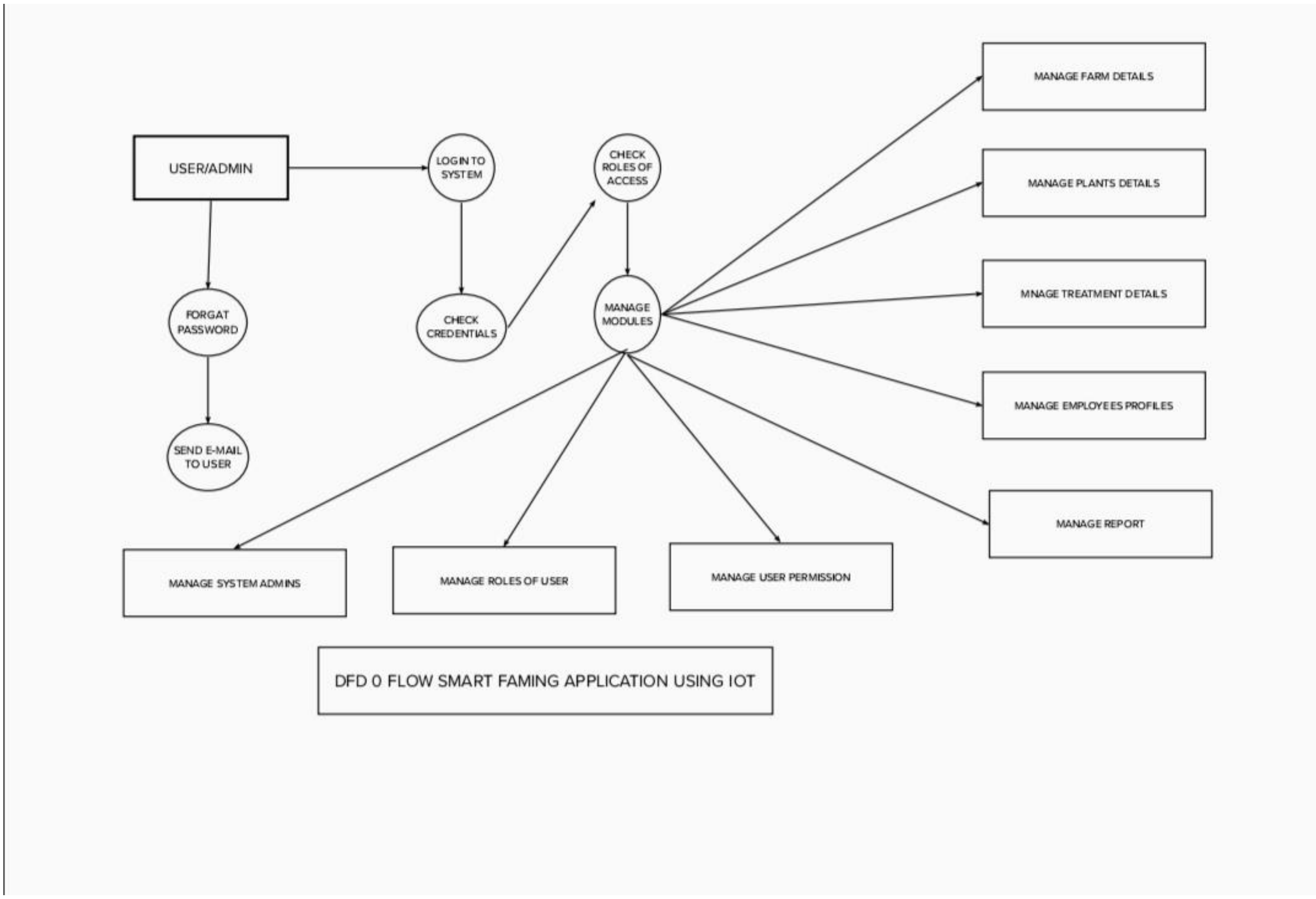
5.Project Design

5.1 Data Flow Diagrams

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.





• The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the ibm cloud.

• Aurdino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.

• NODE-RED is used as a programming tool to write the hardware, software and APIs. The MQTT protocol is followed for the communication.

• All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could make a decision through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.

5.2 Solution & Technical Architecture

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:

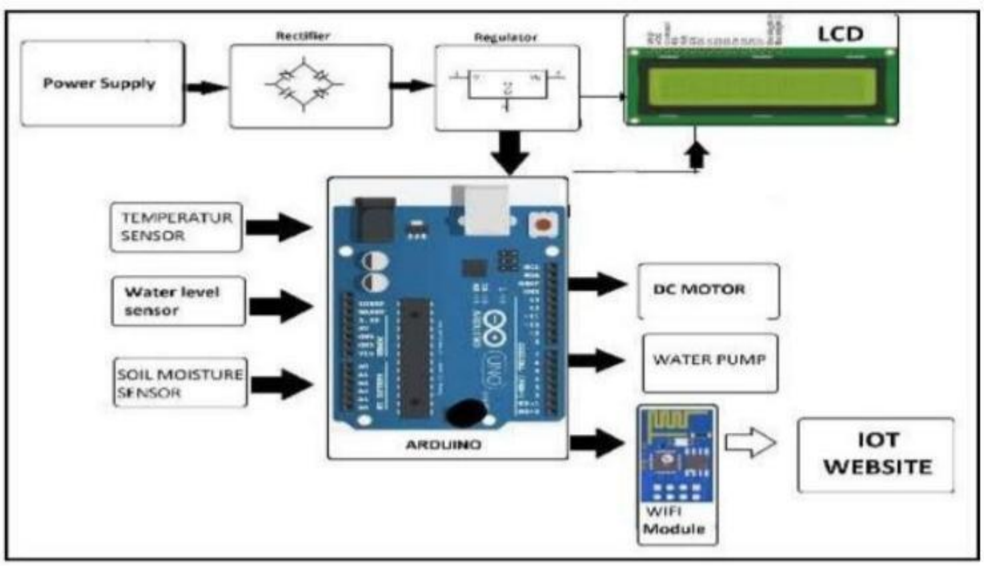


Figure 1: IOT ENABLED SMART FARMING APPLICATION

- The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the ibm cloud.

- Aurdino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.
- NODE-RED is used as a programming tool to write the hardware, software and APIs. The MQTT protocol is followed for the communication.
- All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could make a decision through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.

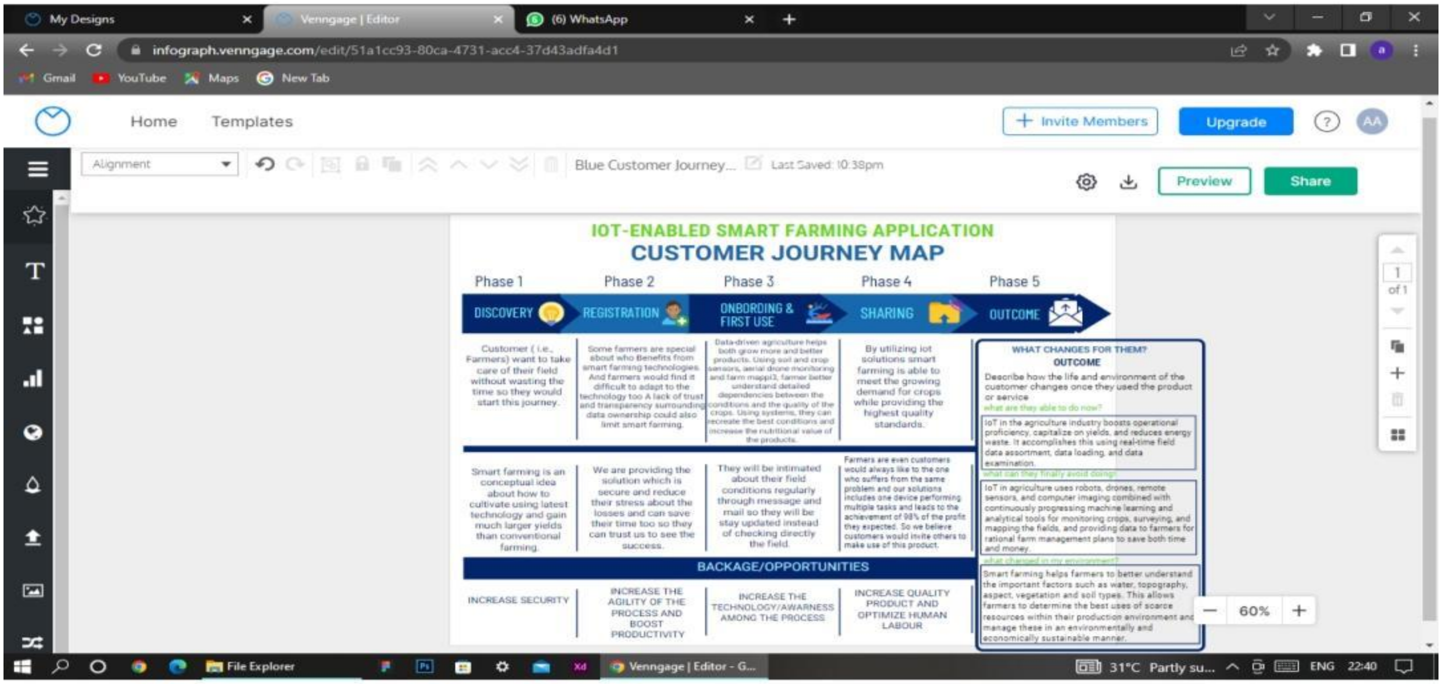
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson IOT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM Cloud
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	Sensitive and private data must be protected from their production until the decision-making and storage stages.	e.g. Node-Red, Open weather App API, MIT App Inventor , etc
3.	Scalable Architecture	scalability is a major concern for IoT platforms. It has been shown that different architectural choices of IoT platforms affect system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousand.	Technology used
S.No	**Characteristics**	**Description**	**Technology**
4.	Availability	Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.	Technology used
5.	Performance	The idea of implementing integrated sensors with sensing soil and environmental or ambient parameters in farming will be more efficient for overall monitoring.	Technology used

5.3 User Stories

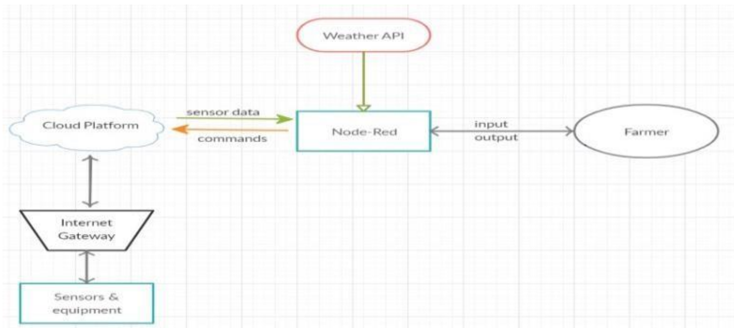


6. Project planning & Scheduling

6.1 Sprint Delivery planning & Estimation

SPRINT DELIVERY OVERVIEW :

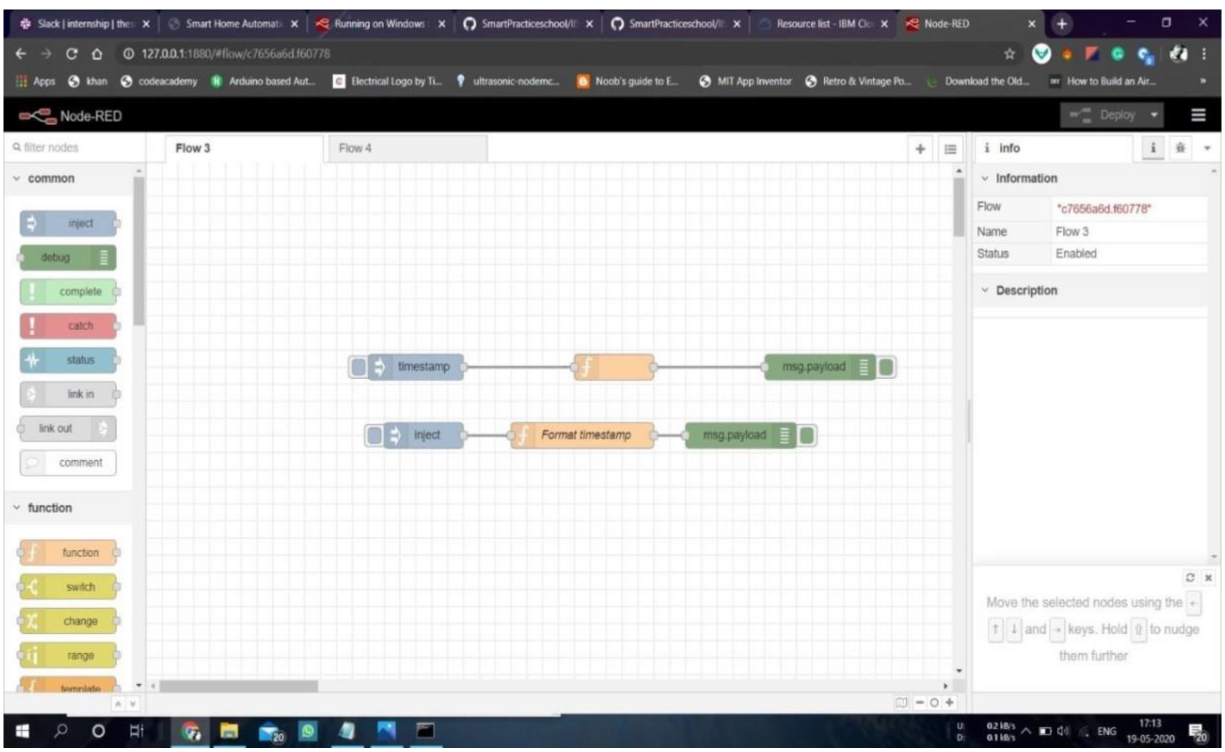
In order to implement the solution , the following approach as shown in the block diagram is used



1, Required Software Installation

1.1 A 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. NodeRED provides a web browser-based flow editor, which can be used to create JavaScript functions.



Installation :

- First install npm/node.js
- Open cmd prompt
- Type => npm install node-red **To run the application :**

- Open cmd prompt
- Type=> node-red

- Then open http://localhost:1880/ in browser

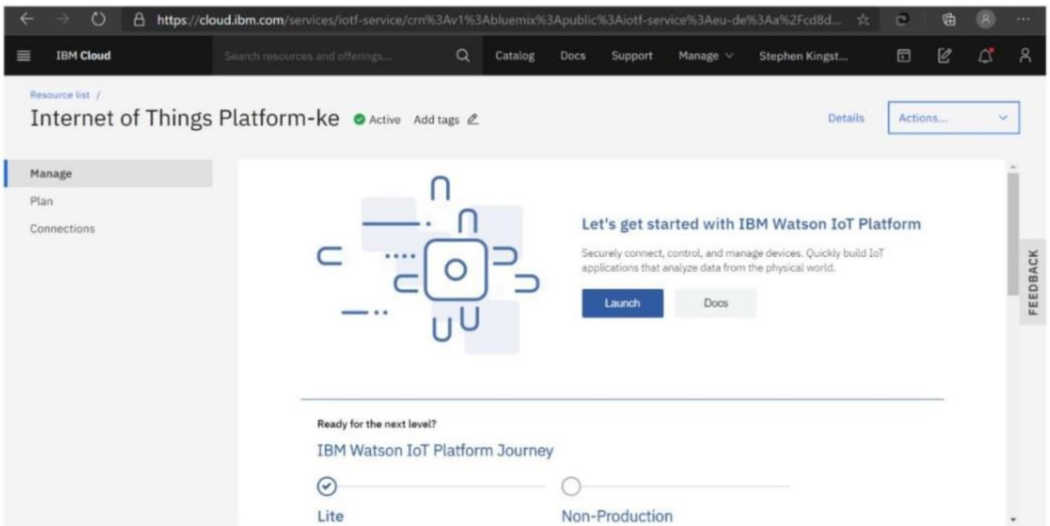
Installation of IBM IoT and Dashboard nodes for Node-Red

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required

1. IBM IoT node
2. Dashboard node

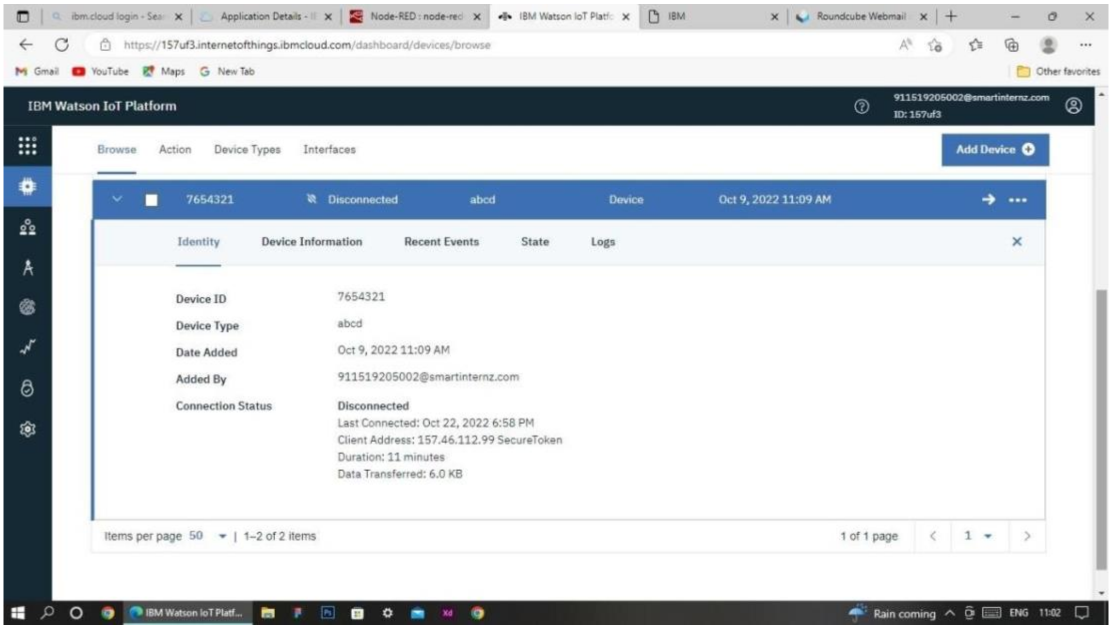
1.2.B IBM Watson IoT Platform

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage. IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices.



Steps to configure:

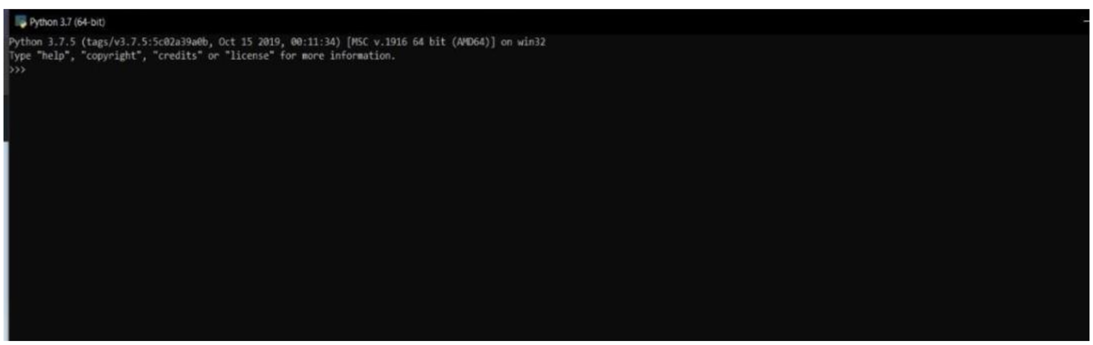
- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token • Create API key and store API key and token elsewhere.



1.2.C Python IDE

Install Python3 compiler

Install any python IDE to execute python scripts, in my case I used Spyder to execute the code.



1.3 IoT Simulator

In our project in the place of sensors we are going to use IoT sensor simulator which give random readings to the connected cloud.

The link to simulator: https://watson-iot-sensor-simulator.mybluemix.net/

We need to give the credentials of the created device in IBM Watson IoT Platform to connect cloud to simulator.

1.4 OpenWeather API

OpenWeatherMap is an online service that provides weather data. It provides current weather data, forecasts and historical data to more than 2 million customer.

Website link: https://openweathermap.org/guide

Steps to configure:

○ Create account in OpenWeather ○ Find the name of your city by searching ○ Create API key to your account ○ Replace “city name” and “your api key” with your city and API key in below red text **api.openweathermap.org/data/2.5/weather?q={city name}&appid={your api key}** Link I used in my project:

http://api.openweathermap.org/data/2.5/weather?q=Gudur,in&appid=62354068e45f41ffa6a5b164714145fe

2. Building Project

2.1 Connecting IoT Simulator to IBM Watson IoT Platform

Open link provided in above section 4.3

Give the credentials of your device in IBM Watson IoT Platform

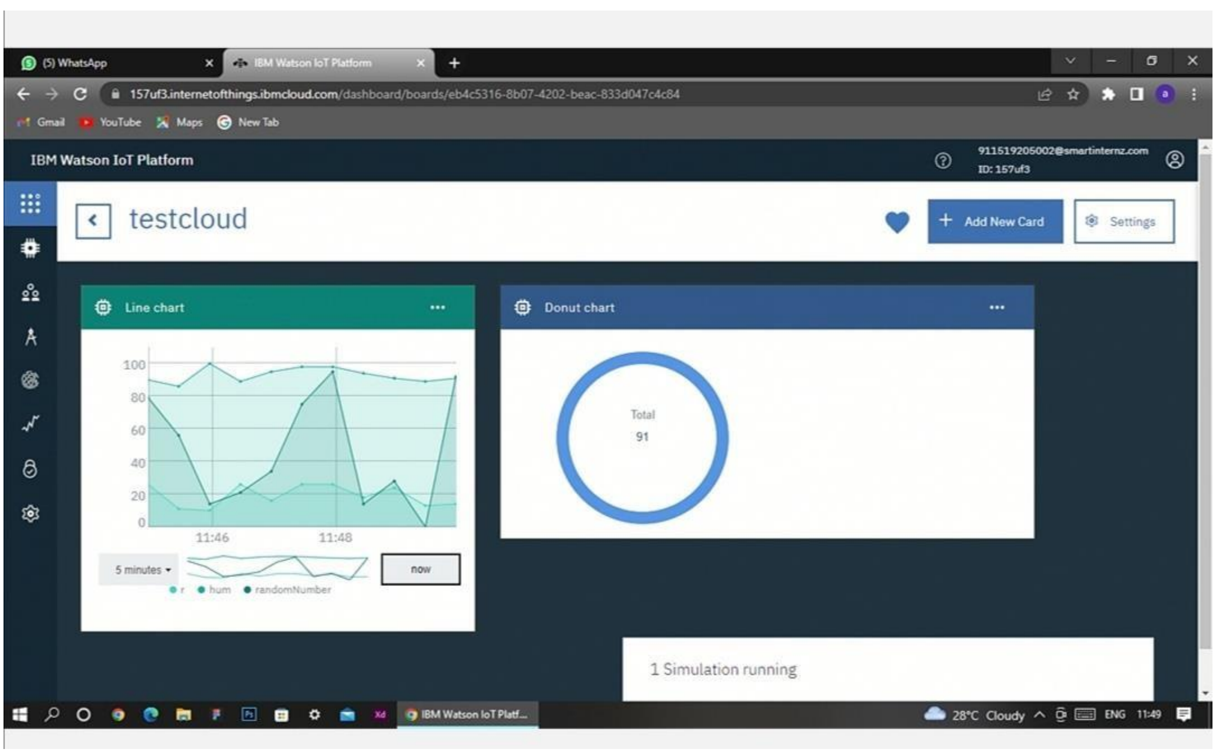
Click on connect

My credentials given to simulator are:

OrgID : 157uf3 api: a-9wbx5m-1qfklrf7jl
Device type: abcd token: JcU(4(9Z37PdL!Rmz(

Device ID : 7654321

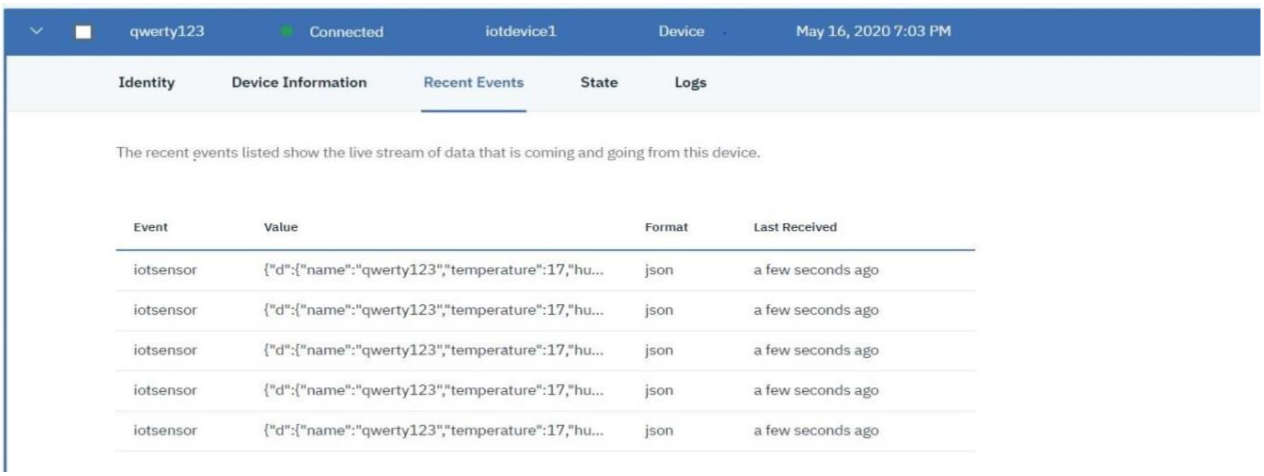
Device Token : 87654321



- You will receive the simulator data in cloud
- You can see the received data in Recent Events under your device
- Data received in this format(json)

{ ○ "d": {

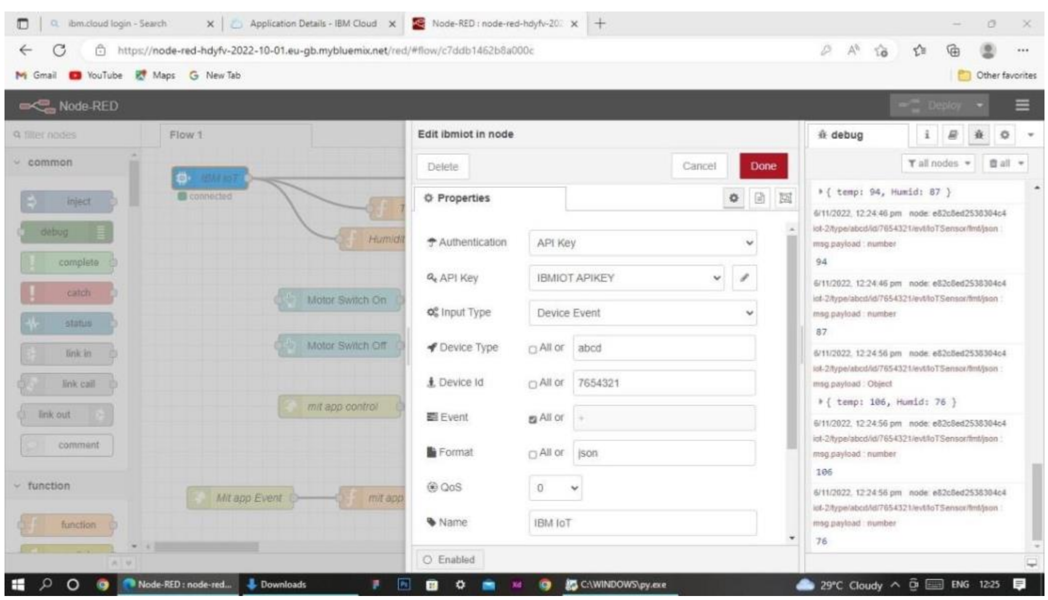
- "name": "qwerty123",
- "temperature": 17,
- "humidity": 76,
- "objectTemp": 25 ○ } }



You can see the received data in graphs by creating cards in Boards tab

2.2 Configuration of Node-Red to collect IBM cloud data

The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red



Once it is connected Node-Red receives data from the device

Display the data using debug node for verification

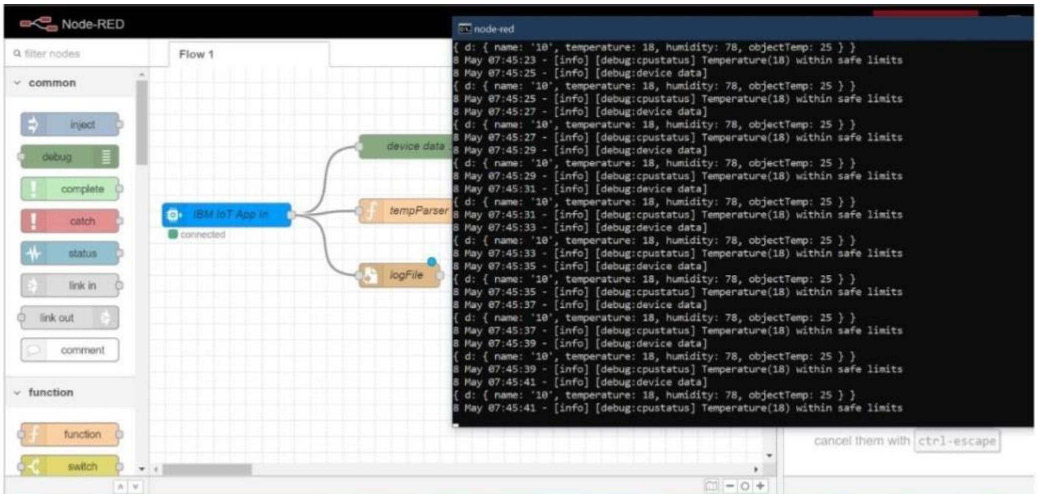
Connect function node and write the Java script code to get each reading separately.

The Java script code for the function node is:

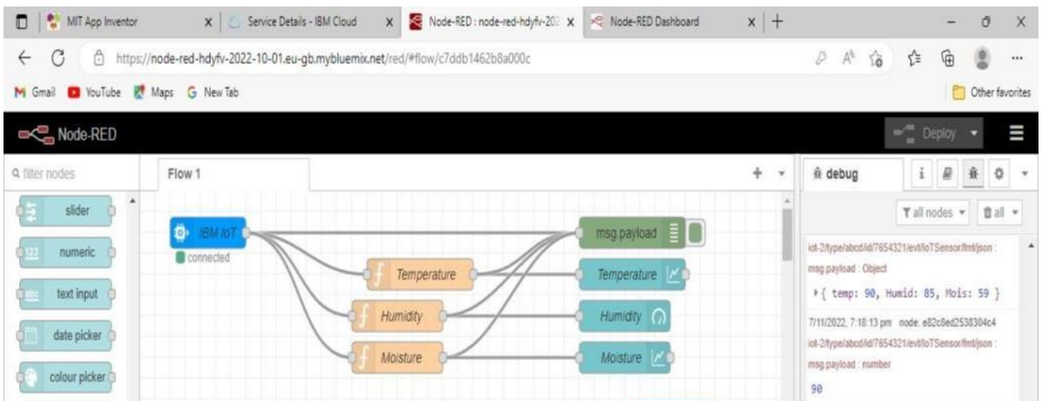
- msg.payload=msg.payload.d.temperature
- return msg;

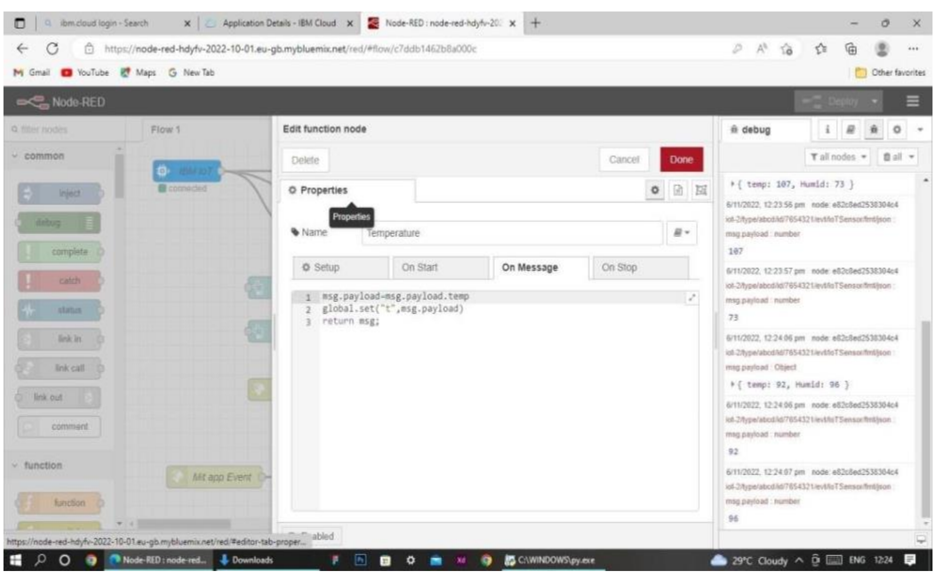
Finally connect Gauge nodes from dashboard to see the data in UI

Nodes connected in following manner to get each reading separately



Data received from the cloud in Node-RED console





This is the Java script code I written for the function node to get Temperature separately.

2.3 Configuration of Node-Red to collect data from OpenWeather

The Node-Red also receive data from the OpenWeather API by HTTP GET request. An inject trigger is added to perform HTTP request for every certain interval.

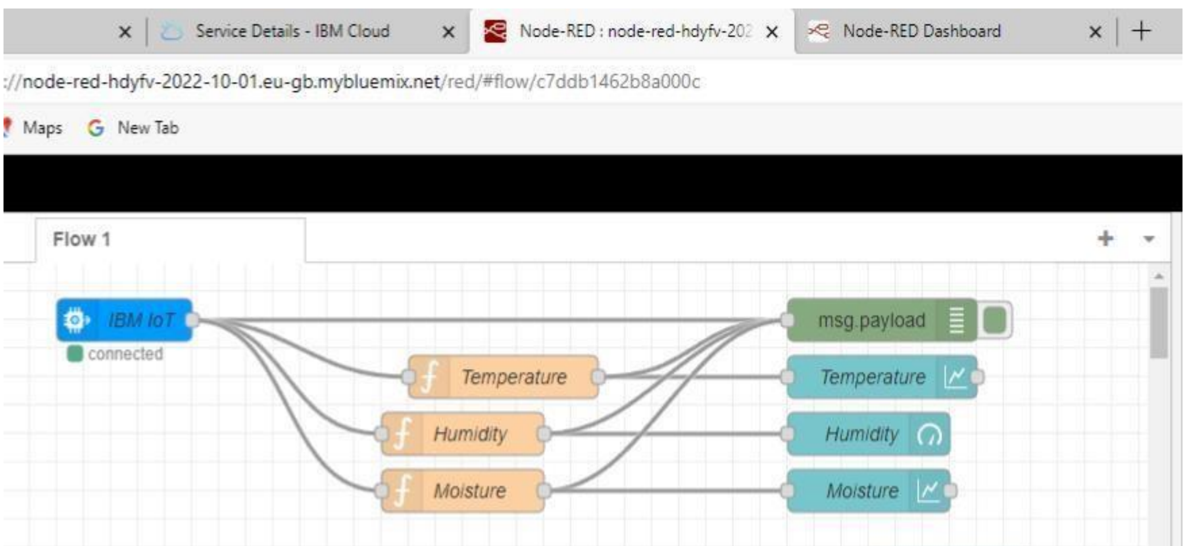
HTTP request node is configured with URL we saved before in section 4.4

The data we receive from OpenWeather after request is in below JSON format:

{"coord":{"lon":79.85,"lat":14.13},"weather":[{"id":803,"main":"Clouds","description":"broken clouds","icon":"04n"}],"base":"stations","main":{"temp":307.59,"feels_like":305.5,"temp_min":30 7.
59,"temp_max":307.59,"pressure":1002,"humidity":35,"sea_level":1002,"grnd_level":1000},"wind": {
"speed":6.23,"deg":170},"clouds":{"all":68},"dt":1589991979,"sys":{"country":"IN","sunrise":1589
93 3553,"sunset":1589979720},"timezone":19800,"id":1270791,"name":"Gūdūr","cod":200} In order to parse the JSON string we use Java script functions and get each parameters

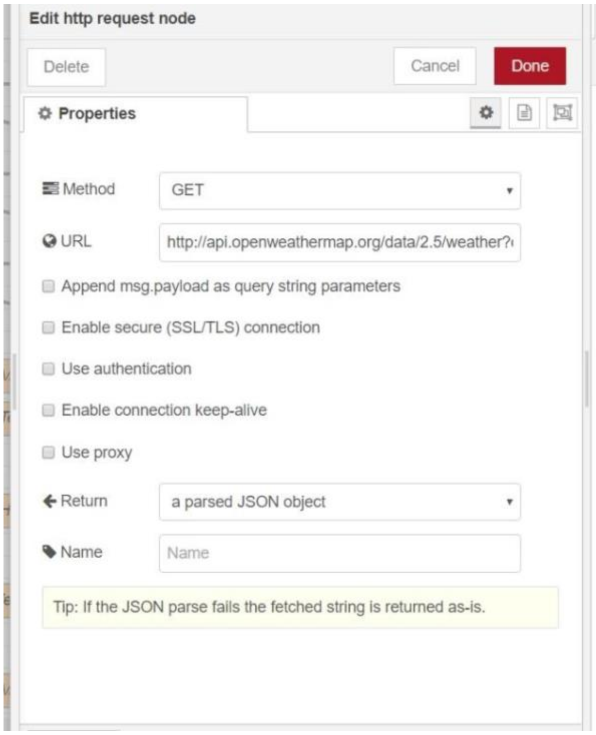
```
var temperature = msg.payload.main.temp;
temperature = temperature-273.15;
return {payload : temperature.toFixed(2)};
```

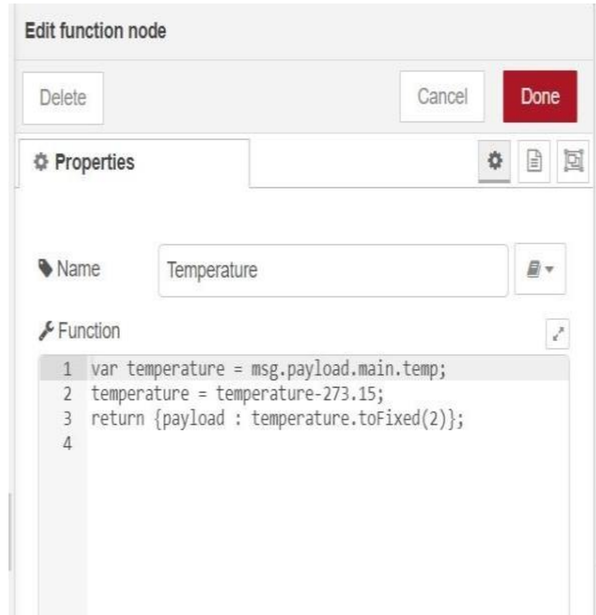
In the above Java script code we take temperature parameter into a new variable and convert it from kelvin to Celsius



Then we add Gauge and text nodes to represent data visually in UI

The above image has the program flow for receiving data from OpenWeather

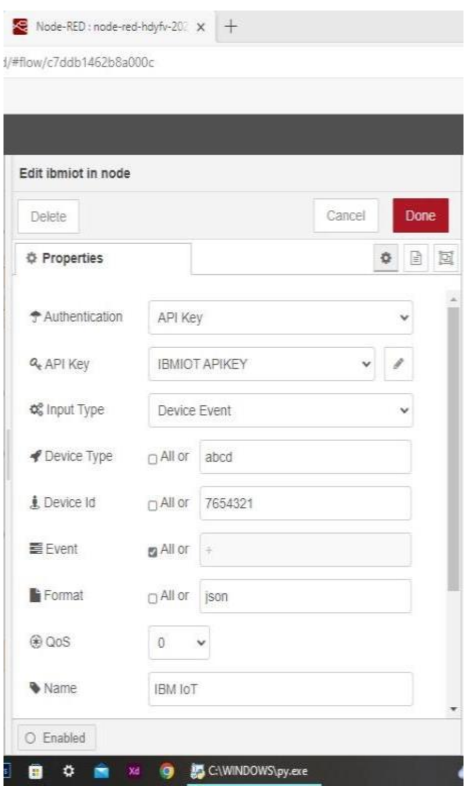




The above two images contain http request and function node data that needs to be filled.

2.4 Configuration of Node-Red to send commands to IBM cloud

ibmiot out node I used to send data from Node-Red to IBM Watson device. So, after adding it to the flow we need to configure it with credentials of our Watson device.



Here we add three buttons in UI which each sends a number 0,1 and 2.

0 -> for motor off

1 -> for motor on

2 -> for running motor continuously 30 minutes

We used a function node to analyse the data received and assign command to each number.

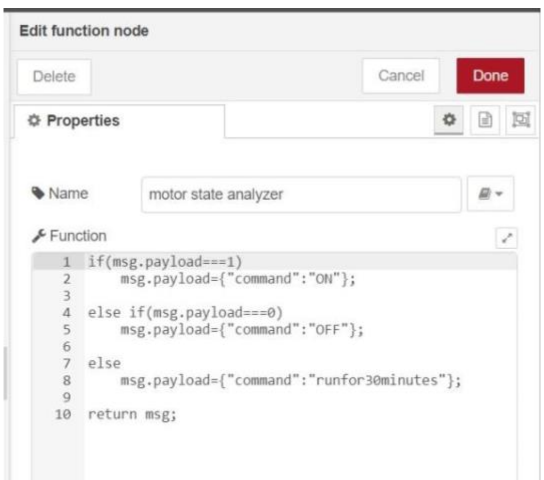
The Java script code for the analyser is:

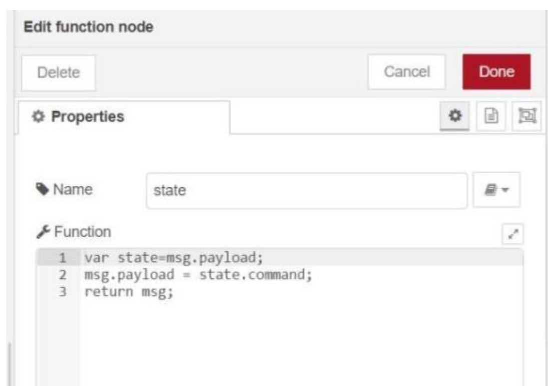
- if(msg.payload===1)
  msg.payload={"command":"ON"}; else
  if(msg.payload===0)
  msg.payload={"command":"OFF"};
  else
  msg.payload={"command":"runfor30minutes"};
- return msg;

Then we use another function node to parse the data and get the command and represent it visually with text node.

The Java script code for that function node is:

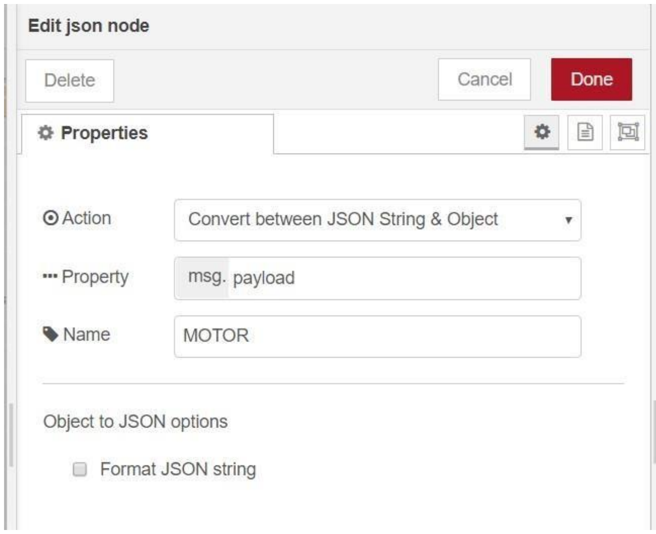
- var state=msg.payload;
  msg.payload = state.command;
- return msg;



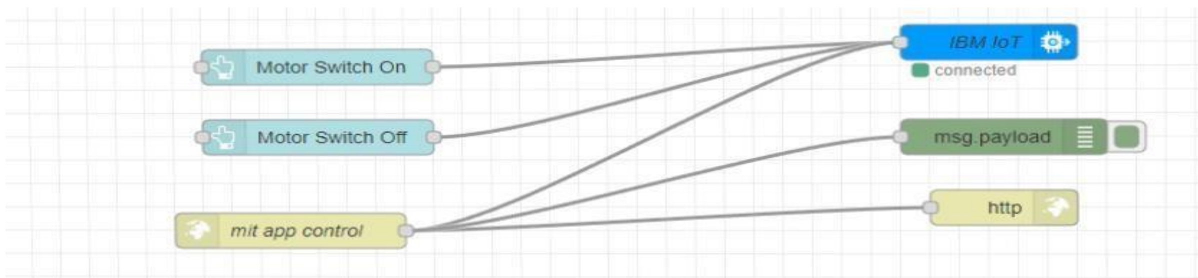


The above images show the java script codes of analyser and state function nodes.

Then we add edit Json node to the conversion between JSON string & object and finally connect it to IBM IoT Out.



Edit JSON node needs to be configured like this



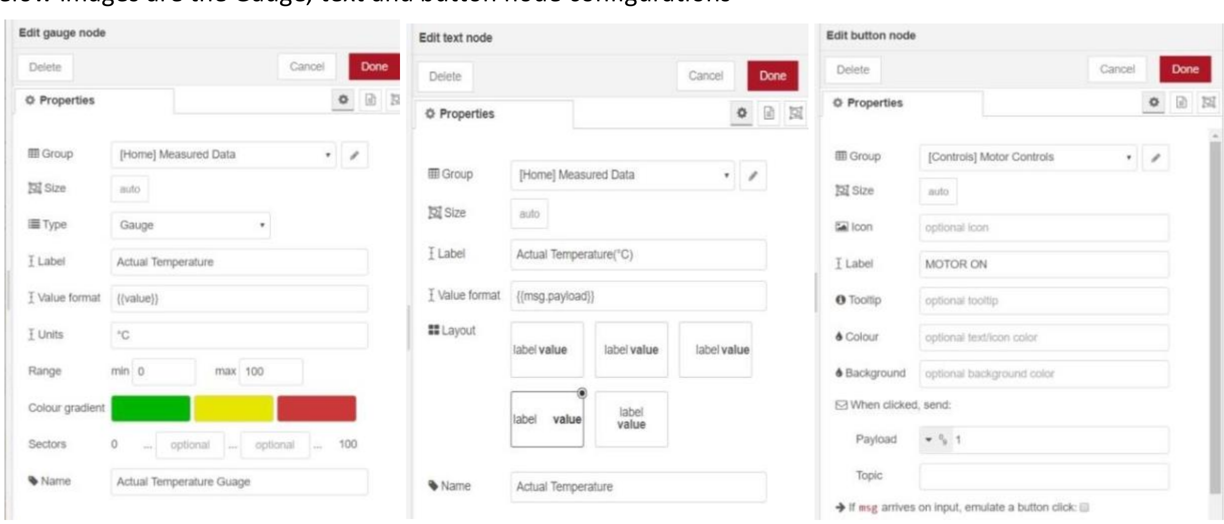
This is the program flow for sending commands to IBM cloud.

2.5 Adjusting User Interface

In order to display the parsed JSON data a Node-Red dashboard is created

Here we are using Gauges, text and button nodes to display in the UI and helps to monitor the parameters and control the farm equipment.

Below images are the Gauge, text and button node configurations



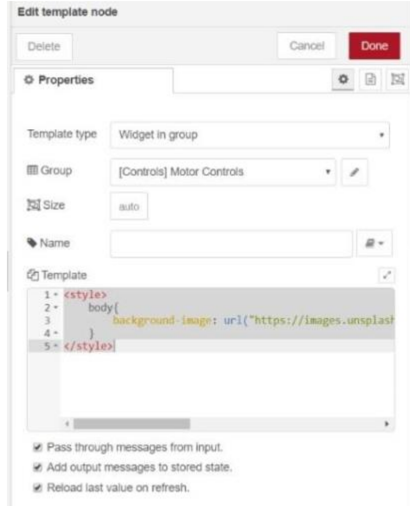
Adding Background image to the UI

To add the background image we are going to add template node and configure it with below HTML code

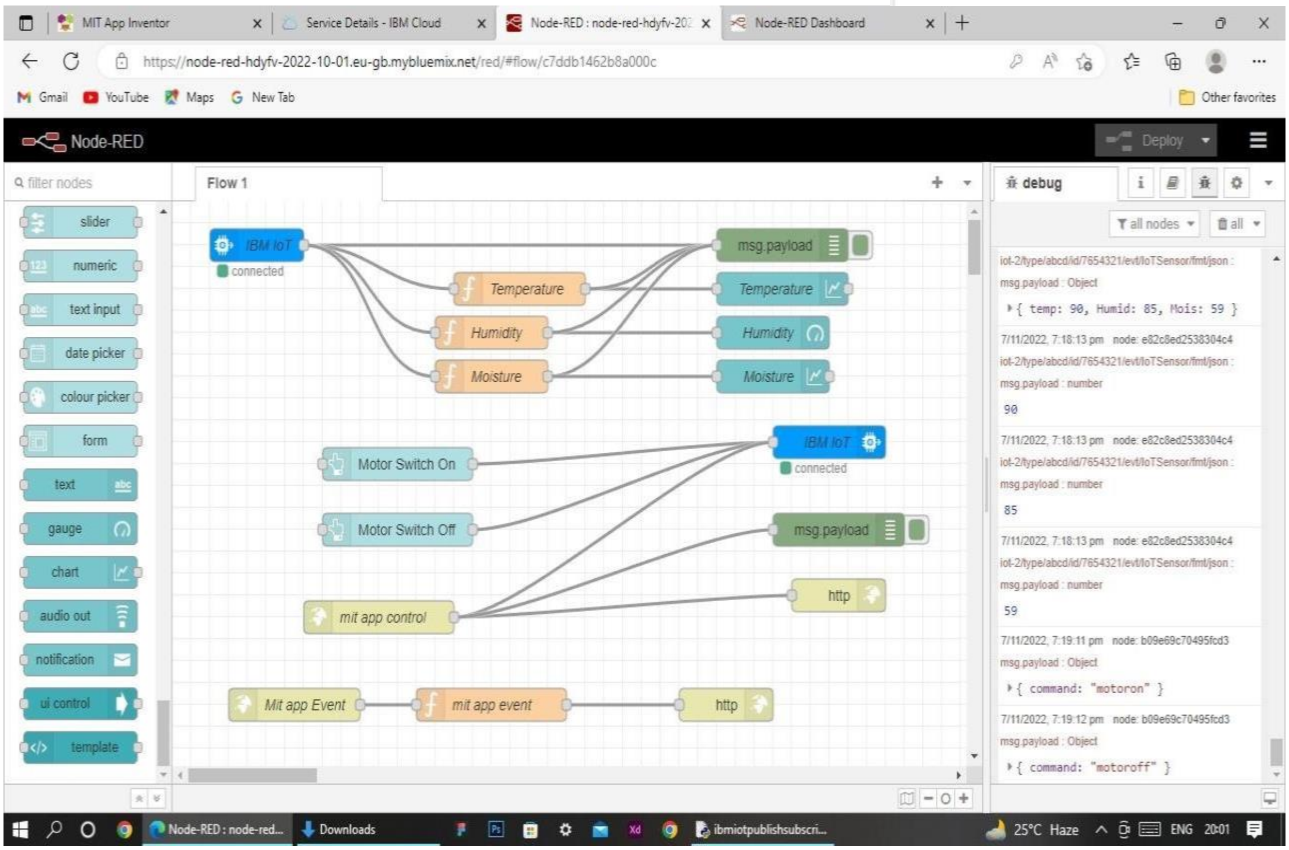
- <style> body{ background-image: url("https://images.unsplash.com/photo-1563514227147-6d2ff665 a6a0?ixlib=rb-1.2.1&auto=format&fit=crop&w=1951&q=80");
  }
- </style>

We add URL of image that need to be displayed

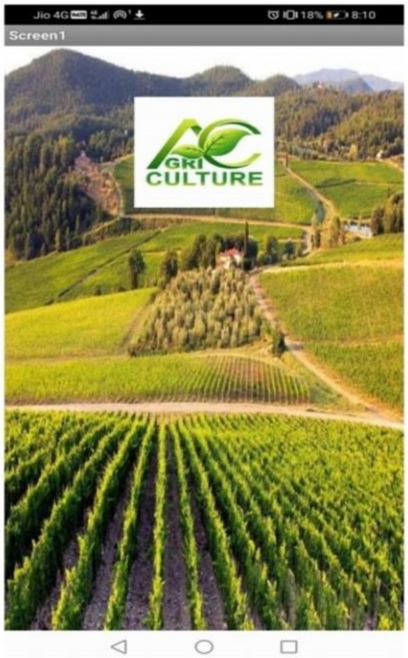
Configuration is shown in the beside image



Complete Program Flow

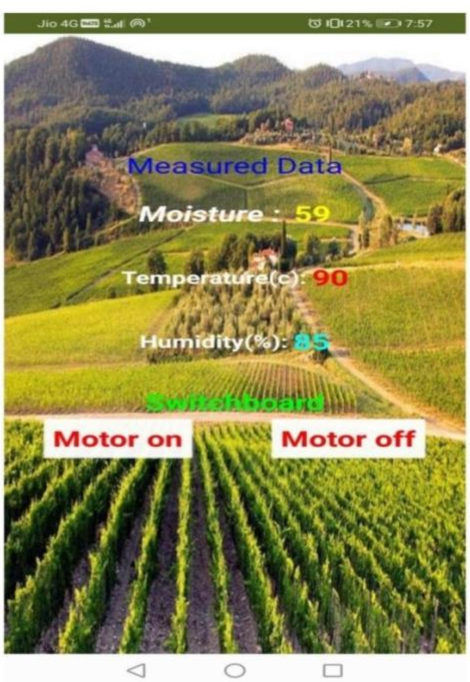


Mobile app



SCREEN – 1

SCREEN - 2

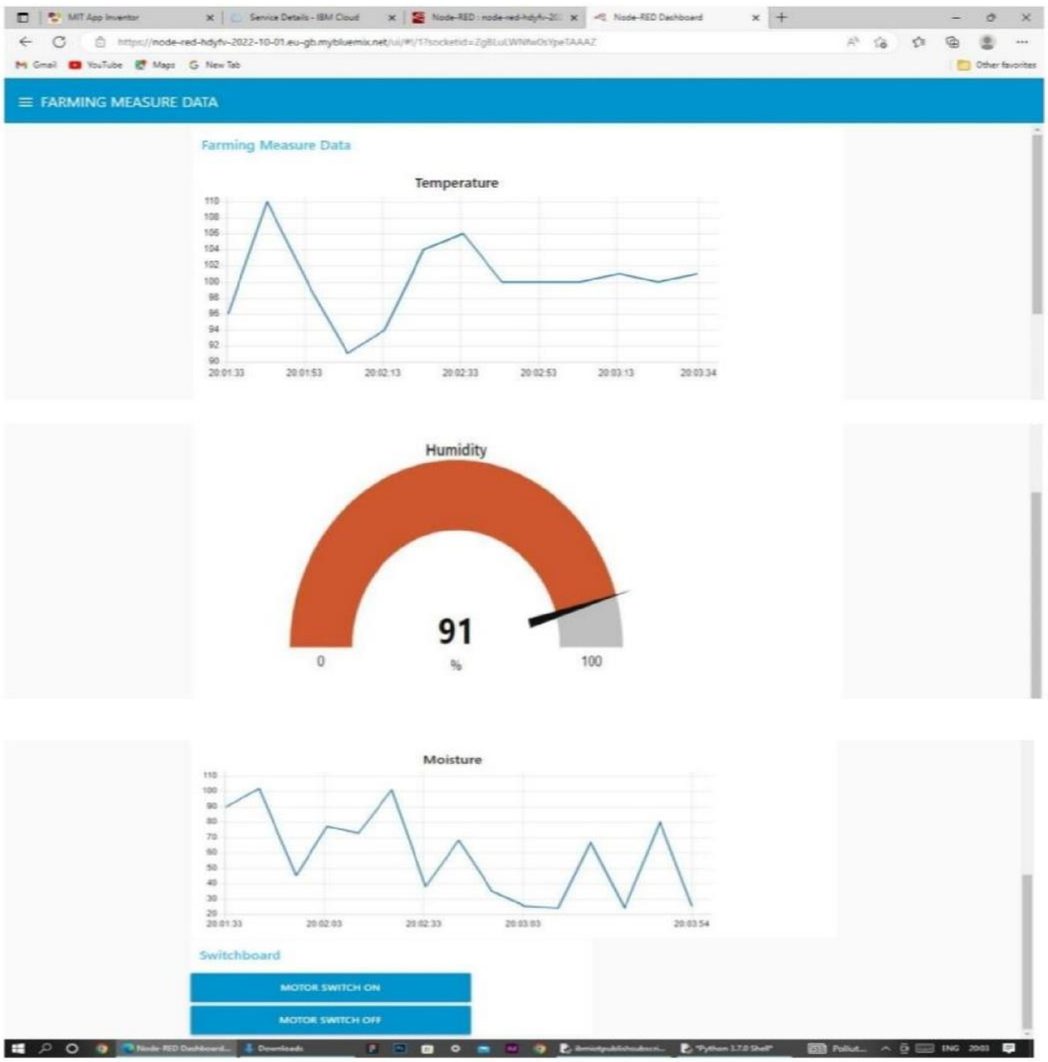


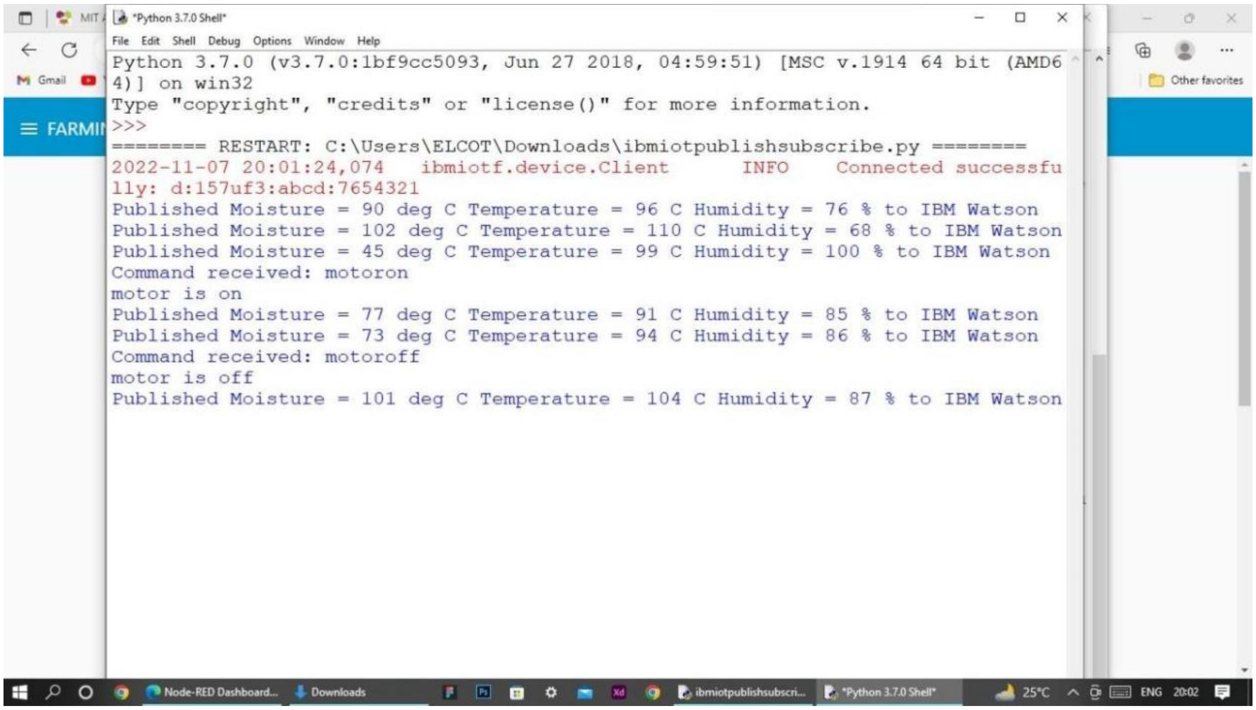
SCREEN - 3

Web APP UI

Home Tab

Web APP UI Home Tab





6.2 Sprint Delivery Planning Schedule

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my Gmail, email then you can received the OTP or Verification Code.	2	High	
Sprint-1		USN-2	As a user, I will receive confirmation Gmail or email once I have registered for the application.	1	High	
Sprint-2		USN-3	As a user, I can register for the application through Gmail and phone number.	2	Low	
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	
	Dashboard	USN-6	Once confirmation message received after login the system and Check Credentials Once check the credentials after go to the Manage modules.	2	High	
		USN-7	In this manage modules described the below functions like Manage System Admins Manage Roles of User Manage User permission and etc..	2	Medium	

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	Logout	USN-8	Then check Temperature, humidity and moisture after then logout or exist the application.	1	Medium	

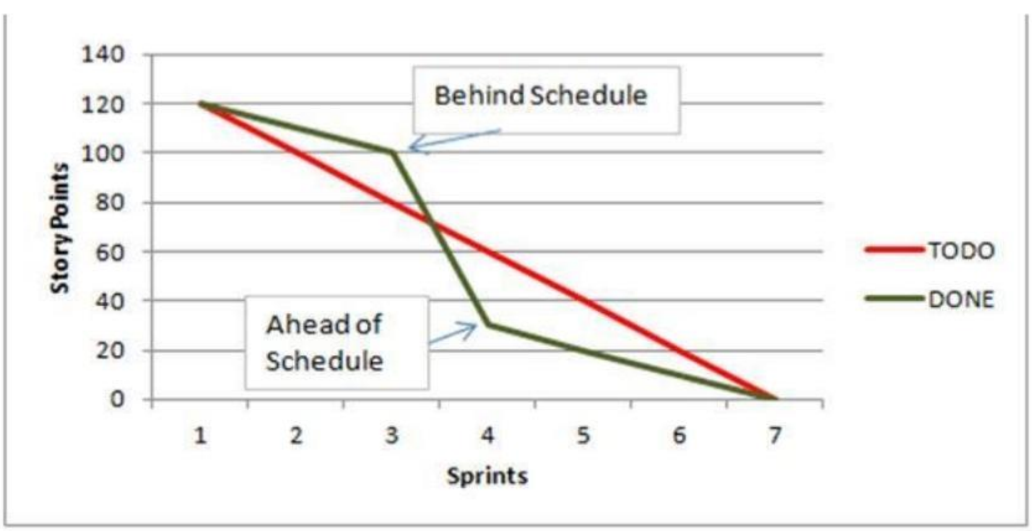
Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	35	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	45	05 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	07 Nov 2022

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

Burndown Chart:



7.Coding & Solution

7.1 Feature - 1

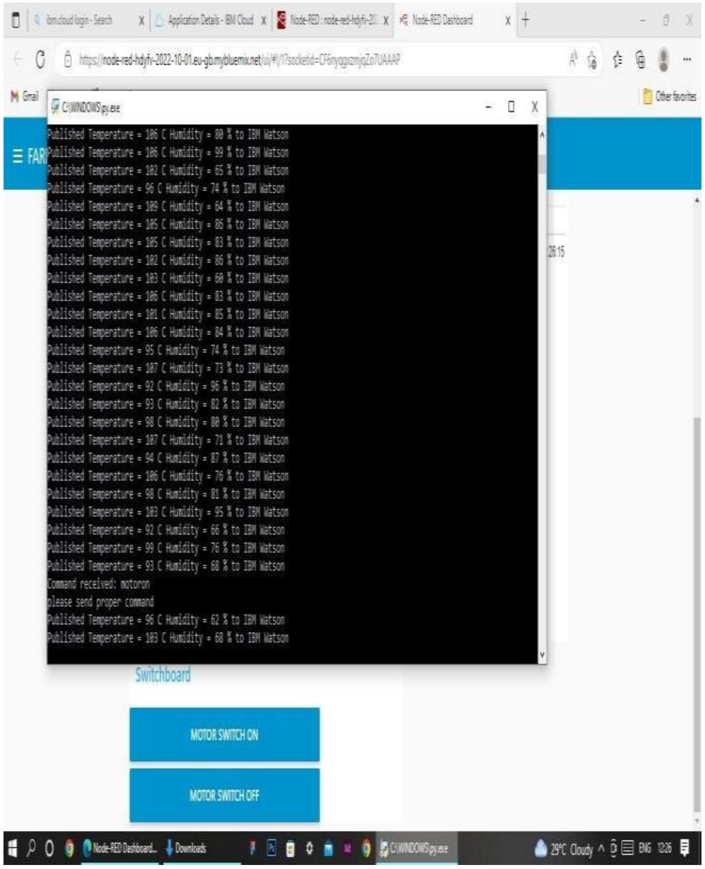
Receiving commands from IBM cloud using Python program

This is the Python code to receive commands from cloud to any device like Raspberry Pi in the farm

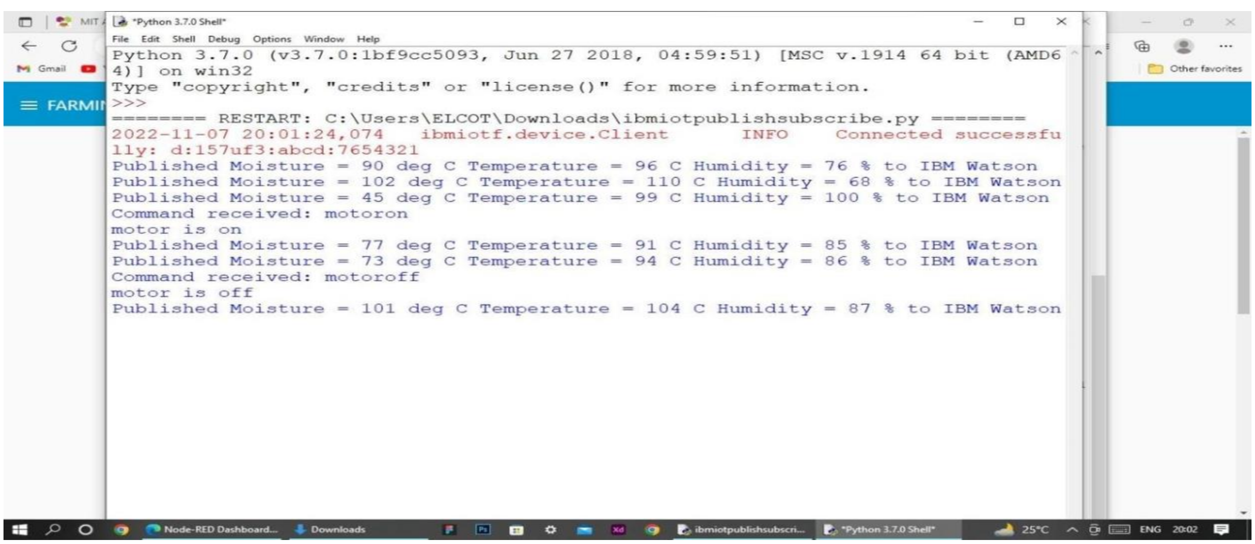
- import time import sys

```
import
ibmiotf.application
import      ibmiotf.device
import random
 #Provide your IBM Watson Device Credentials organization = "157uf3" deviceType = "abcd" deviceId
= "7654321" authMethod = "token" authToken = "87654321"

# Initialize GPIO
def myCommandCallback(cmd):
  print("Command received: %s" %
cmd.data['command'])

status=cmd.data['command']
if status=="motoron":
print ("motor is on")    elif
status == "motoroff":       print
("motor is off")
  else :
    print ("please send proper command")

try:
       deviceOptions = {"org": organization, "type":
deviceType, "id": deviceId, "auth-method": authMethod,
"authtoken": 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:
                      #Get Sensor Data from DHT11

                      temp=random.randint(90,110)
                      Humid=random.randint(60,100)

                Mois=random.randint(20,120)

                      data = { 'temp' : temp, 'Humid': Humid, 'Mois' :Mois}
                      #print data
                      def myOnPublishCallback():
                         print ("Published Temperature = %s C" % temp,
"Humidity = %s %%" % Humid, "Moisture =%s deg c" %Mois, "to IBM
Watson")

                      success = deviceCli.publishEvent("IoTSensor", "json",
data, qos=0, on_publish=myOnPublishCallback)
                      if not success:           print("Not
                connected to IoTF")
                time.sleep(10)

                      deviceCli.commandCallback = myCommandCallback

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


C:\WINDOWS\py.exe
Published Temperature = 106 C Humidity = 80 % to IBM Watson
Published Temperature = 106 C Humidity = 99 % to IBM Watson
Published Temperature = 102 C Humidity = 65 % to IBM Watson
Published Temperature = 96 C Humidity = 74 % to IBM Watson
Published Temperature = 109 C Humidity = 64 % to IBM Watson
Published Temperature = 105 C Humidity = 86 % to IBM Watson
Published Temperature = 105 C Humidity = 83 % to IBM Watson
Published Temperature = 102 C Humidity = 86 % to IBM Watson
Published Temperature = 103 C Humidity = 60 % to IBM Watson
Published Temperature = 106 C Humidity = 83 % to IBM Watson
Published Temperature = 101 C Humidity = 85 % to IBM Watson
Published Temperature = 106 C Humidity = 84 % to IBM Watson
Published Temperature = 95 C Humidity = 74 % to IBM Watson
Published Temperature = 107 C Humidity = 73 % to IBM Watson
Published Temperature = 92 C Humidity = 96 % to IBM Watson
Published Temperature = 93 C Humidity = 82 % to IBM Watson
Published Temperature = 98 C Humidity = 80 % to IBM Watson
Published Temperature = 107 C Humidity = 71 % to IBM Watson
Published Temperature = 94 C Humidity = 87 % to IBM Watson
Published Temperature = 106 C Humidity = 76 % to IBM Watson
Published Temperature = 98 C Humidity = 81 % to IBM Watson
Published Temperature = 103 C Humidity = 95 % to IBM Watson
Published Temperature = 92 C Humidity = 66 % to IBM Watson
Published Temperature = 99 C Humidity = 76 % to IBM Watson
Published Temperature = 93 C Humidity = 68 % to IBM Watson
Command received: motoron
please send proper command
Published Temperature = 96 C Humidity = 62 % to IBM Watson
Published Temperature = 103 C Humidity = 68 % to IBM Watson
Switchboard
MOTOR SWITCH ON
MOTOR SWITCH OFF

Feature – 2


Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\ELCOT\Downloads\ibmiotpublishsubscribe.py ========
2022-11-07 20:01:24,074 ibmiotf.device.Client INFO Connected successfully: d:157uf3:abcd:7654321
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 68 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson

8.Testing

8.1 Test Cases

Shopenzer Testcases Testscearnios

Date	3-Nov-22
Team ID	PNT2022TMIDxxxxxx
Project Name	Project - xxx
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
LoginPage_TC_001	Functional	Home Page	Verify user is able to see the Login/ Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	MIT App Inventor https://appinventor.mit.edu	Login popup should display		Fail	Steps not Clear to follow		Bug-1234	
LoginPage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter Smart App 2.Verify login/Singup popup with below UI elements: a.Username text box b.password text box c.Submit button d.New customer? Create account link e.Last password? Recovery password	MIT App Inventor https://appinventor.mit.edu	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass				
LoginPage_TC_003	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter MIT App Inventor URL (https://appinventor.mit.edu) Smart app and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box	Username: IBM password: IBM	User should navigate to user account homepage	Working as Expected	Pass				
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL MIT App Inventor https://appinventor.mit.edu and smart app click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box	Username: chalam@gmail password: Testing123	Application should show 'Incorrect email or password ' validation message.	Working as Expected	Pass				

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Increasing control over production leads to **better cost management and waste reduction**. The ability to trace anomalies in crop growth or livestock health, for instance, helps eliminate the risk of losing yields. Additionally, automation boosts efficiency. Smart farming **reduces the ecological footprint of farming**. Minimized or site-specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse gases.

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

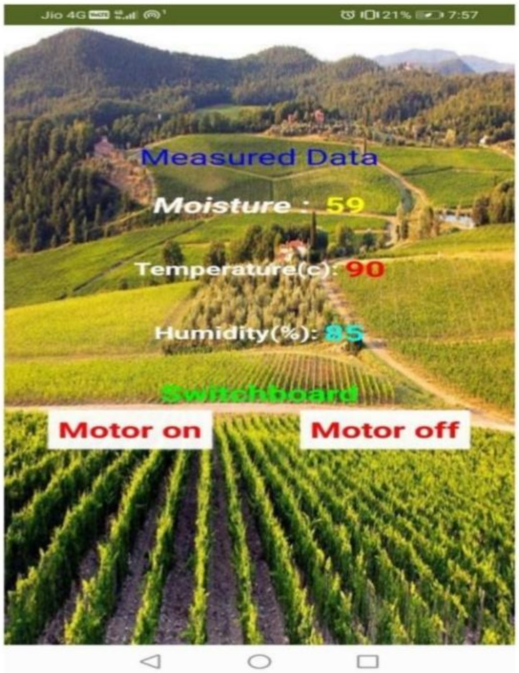
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	2	2	16
Duplicate	1	0	2	0	3
External	2	3	0	1	6
Fixed	9	2	3	17	31
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	1	4	1	1	7
Totals	21	12	9	22	66

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

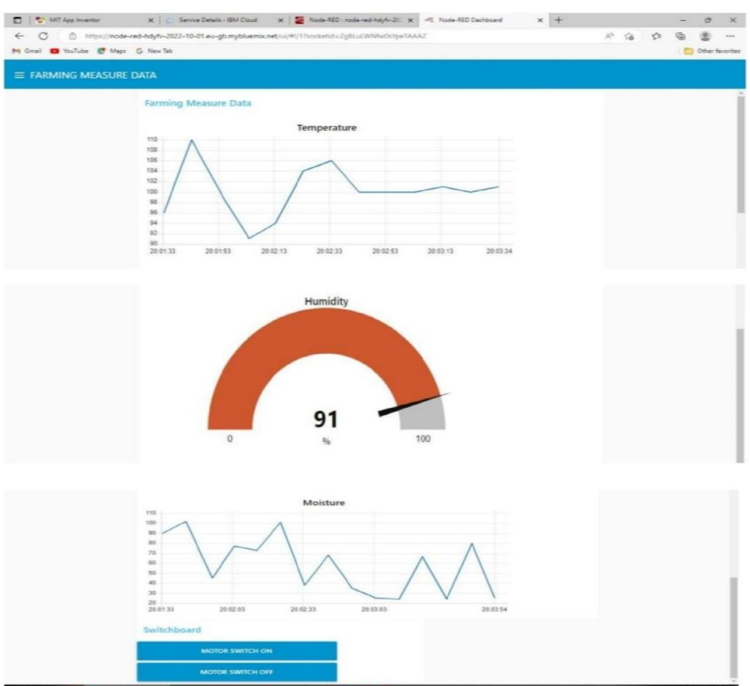
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2
Outsource Shipping	2	0	0	2
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	1	0	0	1

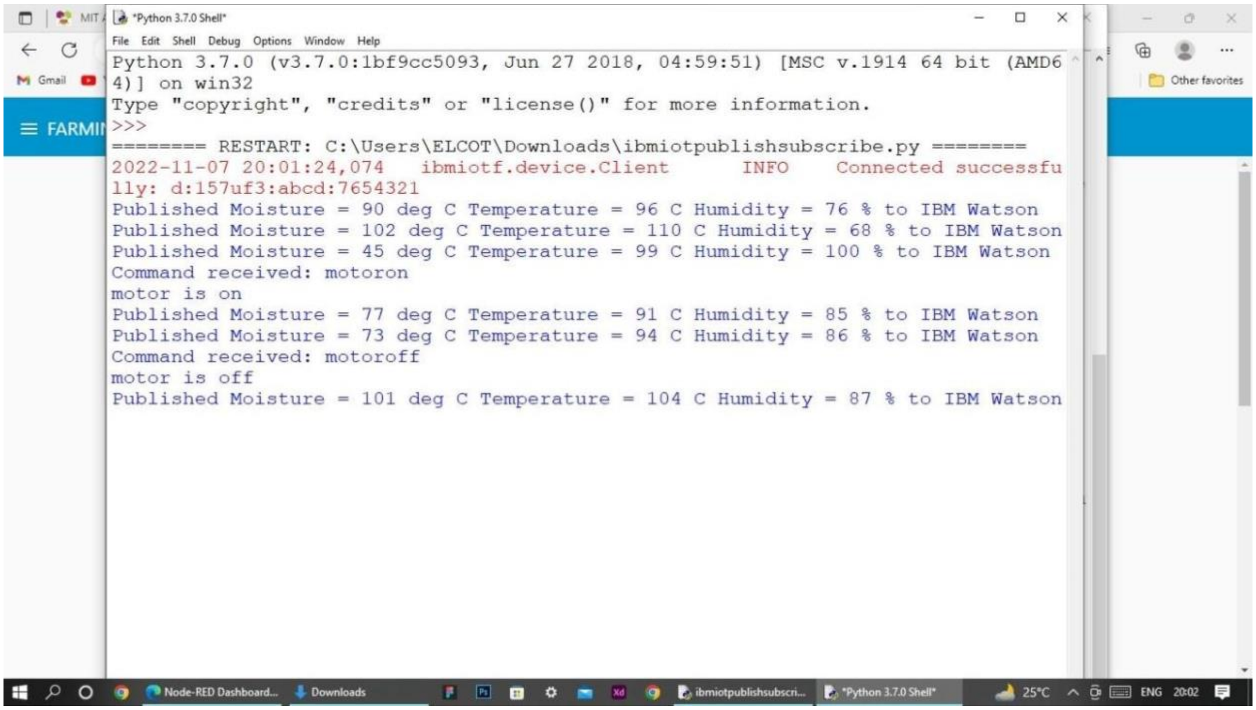
9.Results :



SCREEN - 3

Web APP UI Home Tab





10.Advantages & Disadvantages

Advantages:

- Farms can be monitored and controlled remotely.
- Increase in convenience to farmers.
- Less labour cost.
- Better standards of living.

Disadvantages:

- Lack of internet/connectivity issues.
- Added cost of internet and internet gateway infrastructure.
- Farmers wanted to adapt the use of WebApp.

11 .Conclusion

Thus the objective of the project to implement an IoT system in order to help farmers to control and monitor their farms has been implemented successfully.

12, Future Scope

Through collecting data from sensors using IoT devices, you will learn about the real-time state of your crops. The future of IoT in agriculture **allows predictive analytics to help you make better harvesting decisions**.InShot_20221118_112302897

Smart farming refers to **managing farms using modern Information and communication technologies to increase the quantity and quality of products while optimizing the human labor required**. Among the technologies available for present-day farmers are: Sensors: soil, water, light, humidity, temperature management.

IOT TECHNOLOGIES IN AGRICULTURE. IoT smart agriculture products are designed to **help monitor crop fields using sensors and by automating irrigation systems**. As a result, farmers and associated brands can easily monitor the field conditions from anywhere without any hassle.

10.Appendix :

Source code :

Link : https://1drv.ms/w/s!AoaBQFCdTpb1gRnzU4VXQ_y4nGlA

Links:

IBM cloud reference: https://cloud.ibm.com/

Github link : https://github.com/IBM-EPBL/IBM-Project-47495-1660799879

Demo Link : https://github.com/IBM-EPBL/IBM-Project-47495-1660799879/blob/main/Final%20Delivery/InShot_20221118_112302897.mp4

IOT Watson simulator : https://157uf3.internetofthings.ibmcloud.com/dashboard/devices/browse

Node-Red : https://node-red-hdyfv-2022-10-01.eu-gb.mybluemix.net/red/#flow/c7ddb1462b8a000c

MIT App Inventor : http://ai2.appinventor.mit.edu/#5147143820935168