PROJECT REPORT

Project Name:

SMART FARMER- IOT ENABLED SMART FARMING APPLICATION.

Team ID: PNT2022TMID40437

Team:

ARUN.A-TEAM LEAD

SANDHIYA.E

STEEDHAR.R

SANDHIYA.P

SMART FARMING

1.INTRODUCTION:

1.1 PROJECT OVERVIEW:

This is system that enables framers to monitor and their forms with a web based application build with Node-RED.
It uses the IBM IOT Watson cloud platform as its Backend.

1.2 PURPOSE:

Smart Farming reduce the ecological foodprint 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:

The biggest challenges faced by IoT in the agricultural
sector are lack of information, high adoption costs , and security concers , etc.
Most of the farmers are not aware of the implementation of IoT in agriculture.

2.2 REFERENCES:

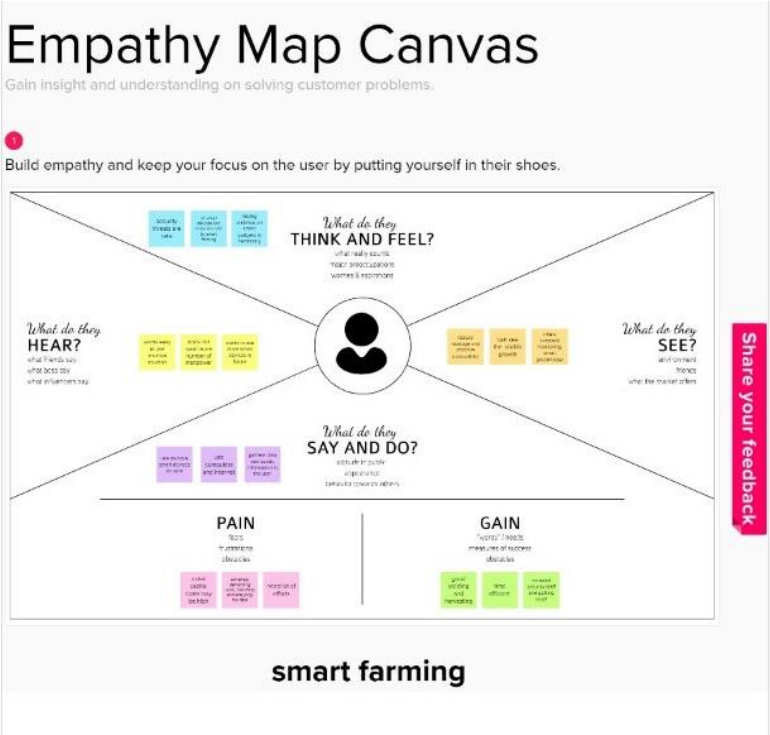
It is the application of modern ICT (Information and Communication Technologies) into agriculture. In IOT- based smart farming, a system is built for monitoring the crop field with the help of sensors (light, humidity, temperature, soil moisture, etc.). The farmers can monitor the field conditions from anywhere.

2.3 PROBLEM STATEMENT DEFINITION:

Overuse of pesticides and fertilizer in agricultural fields leads to destruction of the crop as well as reduces the efficiency of the field increasing the soil vulnerability toward pest. IoT applications may be used to update the farmer/ user about type & quantity of pesticide required by the crop.

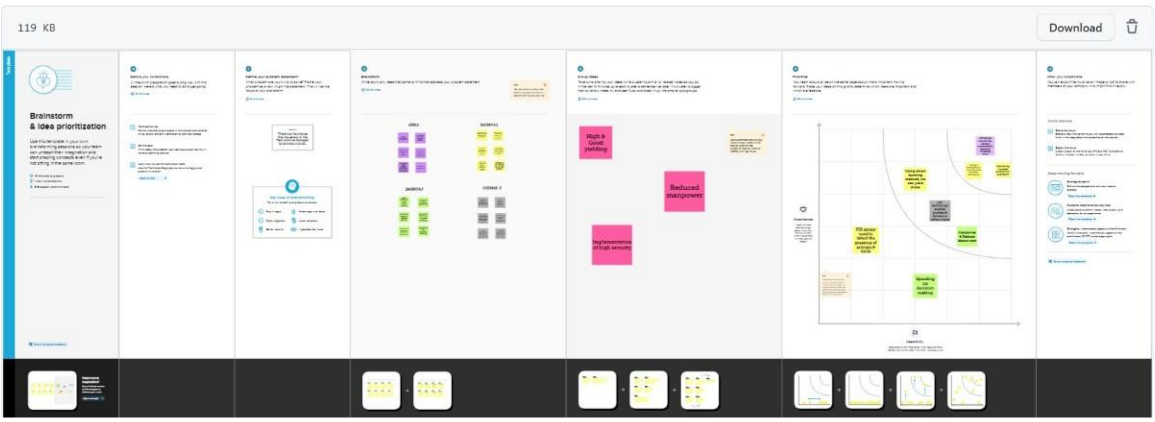
3. IDEATION & PROPOSED SOLUTION:

3.1 EMPATHY MAP CANVAS:



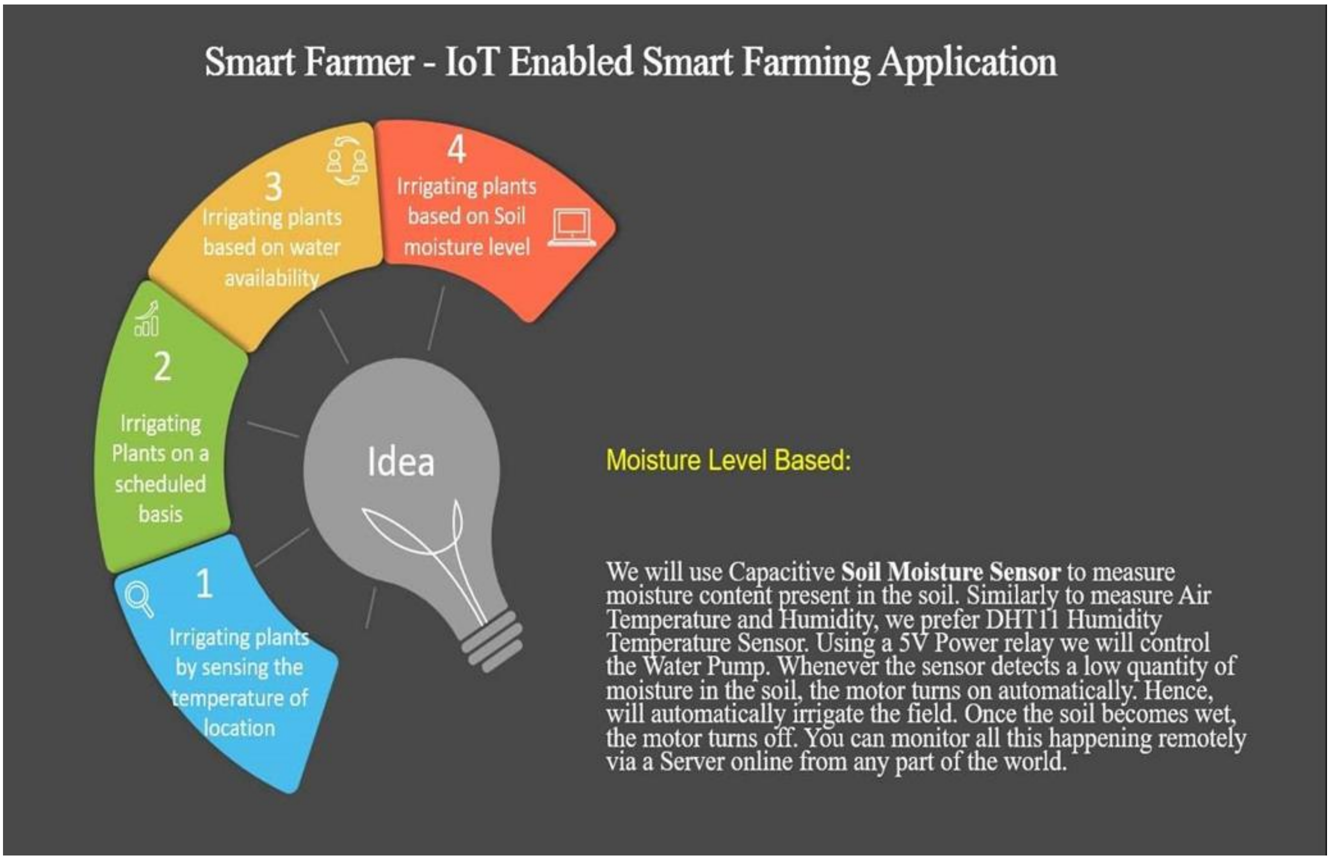
3.2 IDEATION & BRAINSTORMING:

Ideation is the create process of generating, developing, and communicating new ideas, where an is idea understood as a basic element of thought that can be either visual, concrete, or abstract.



Brainstorming is a group creative technique by which efforts are made to find a conclution for a specific problem by gatherin a list of ideas spontaneously contributed by its members.

IDEATION PROCESS



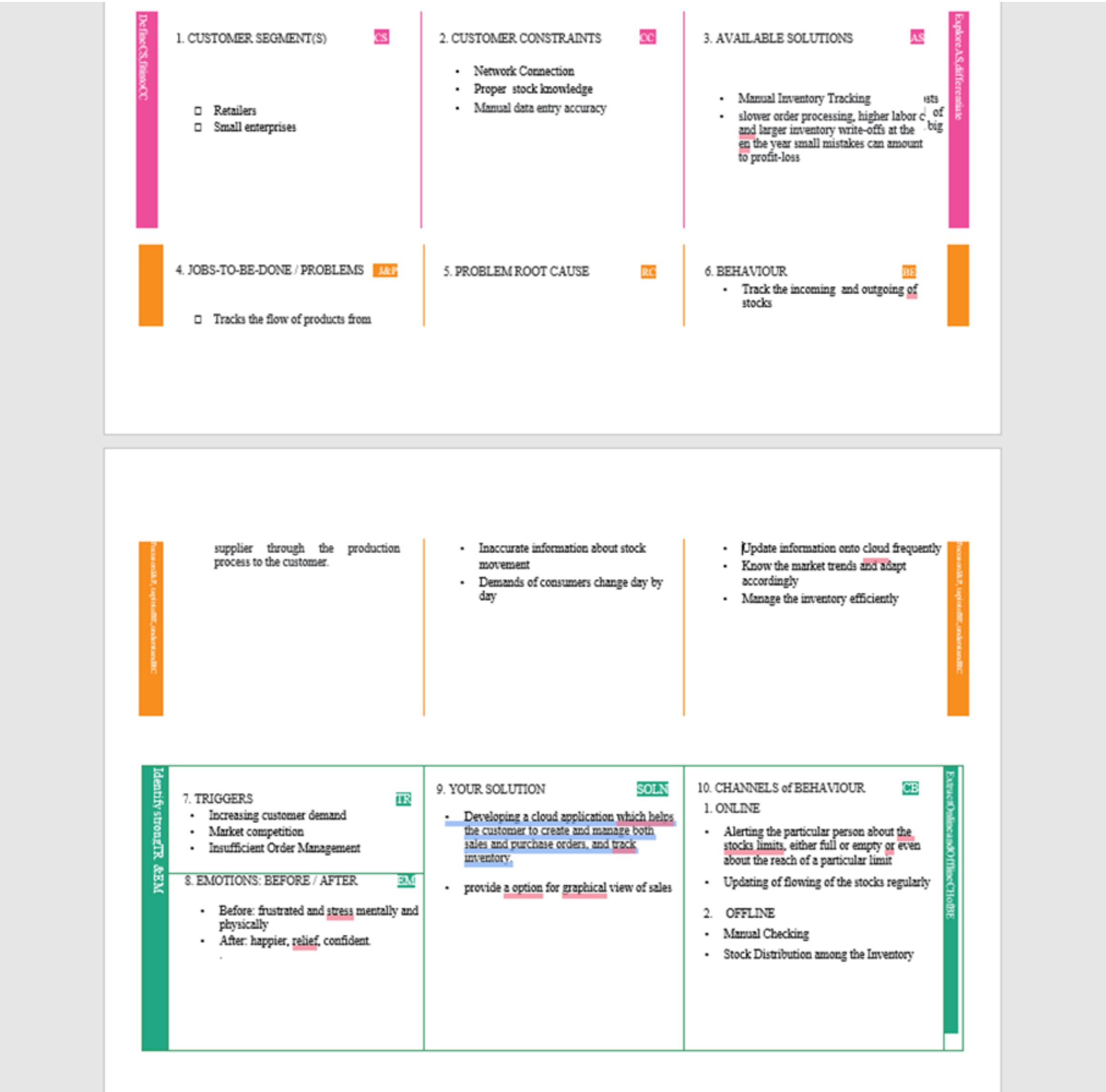
3.3 Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	To make farming easier by choosing several constraints in agriculture and to overcome those constraints, to increase production quality and quantity using IOT.
2	Idea / Solution description	Using smart techniques like monitoring farms climate, smart irrigation and soil analysis.
3	Novelty / Uniqueness	Solar power smart irrigation system which helps you to monitor temperature, moisture ,humidity using smart sensors.
4	Social Impact / Customer Satisfaction	It is better than the present modern irrigation system by using this method we can control soil erosion. There will be better production yield.

5	Business Model (Revenue Model)	As the productivity increases customer satisfaction also increases and hence need for the application also increases, which increases the revenue of the business.
6	Scalability of the Solution	It is definetly scalable we ca increase the constraints when the problem arises.

3.4 PROBLEM SOLUTIONS FIT :



4.REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL ANALYSIS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	IoT devices	Sensors and Wifi module.
FR-2	Software	Web UI, Node-red, IBM Watson, MIT app

4.2 NON FUNCTIONAL REQUIREMENTS:

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

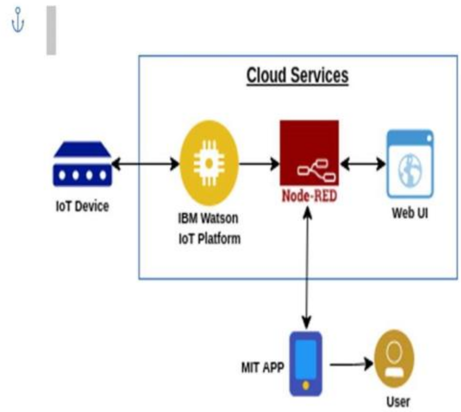
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Time consumability is less, Productivity is high.
NFR-2	Security	It has low level of security features due to integration of sensor data.
NFR-3	Reliability	Accuracy of data and hence it is Reliable.
NFR-4	Performance	Performance is high and highly productive.
NFR-5	Availability	With permitted network connectivity the application is accessible
NFR-6	Scalability	It is perfectly scalable many new constraints can be added

5.PROJECT DESIGN:

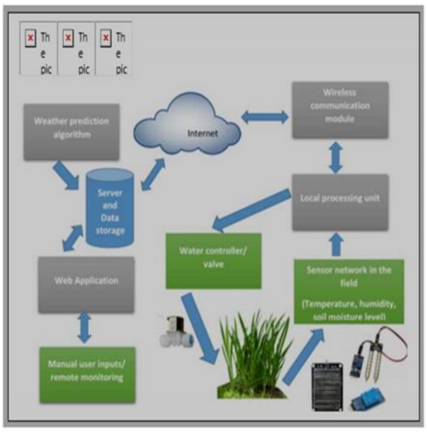
5.1 DATA FLOW DAIGRAMS AND USER STORIES:

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.

Example: (Simplified)



Example: DFD Level 0



5.2 SOLUTIONS AND TECHNICAL ARCHITECTURAL:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

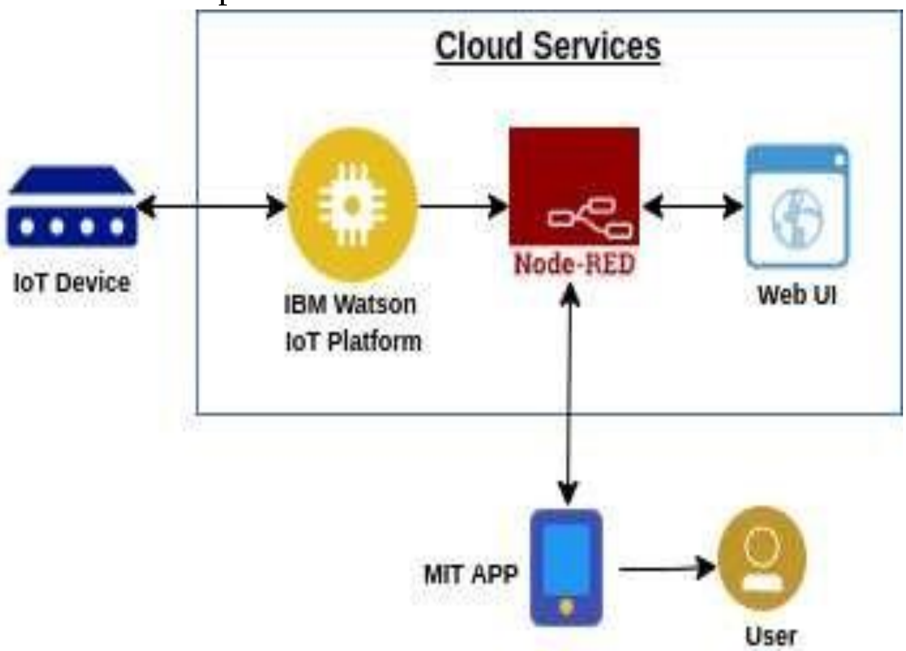


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	MIT app
2.	Application Logic-1	Logic for a process in the application	Node red/IBM Watson/MIT app
3.	Application Logic-2	Logic for a process in the application	Node red/IBM Watson/MIT app
4.	Application Logic-3	Logic for a process in the application	Node red/IBM Watson/MIT app
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM cloud.
7.	Temperature sensor	Monitors the temperature of the crop	
8.	Humidity sensor	Monitors the humidity	
9.	Soil moisture sensor (Tensiometers)	Monitors the soil temperature	
10.	Weather sensor	Monitors the weather	.
11.	Solar panel		.
12.	RTC module	Date and time configuration	
13.	Relay	To get the soil moisture data	

Table-2: Application Characteristics:

S.No.	Characteristics	Description	Technology
1	open-Source Frameworks	MIT app, Node-Red	Software
2	Scalable Architecture	Drone technology, pesticide monitoring, Mineral identification in siol	Hardware

6. PROJECT PLANNING AND SCHEDULING:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Hardware	USN-1	Sensors and wi-fi module with python code.	2	High	Arun.A, Sandhiya.E, Sandhiya.P, Steedhar
Sprint-2	Software	USN-2	IBM Watson IoT platform, Workflows for IoT scenarios using Node-red	2	High	Arun.A, Sandhiya.E, Sandhiya.P, Steedhar
Sprint-3	MIT app	USN-3	To develop an mobile application using MIT	2	High	Arun.A, Sandhiya.E, Sandhiya.P, Steedhar
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Arun.A, Sandhiya.E, Sandhiya.P, Steedhar

7. CODING & SOLUTIONS:

FEATURE :

python code.py - C:\Python\Python37\python code.py (3.7.0)

File Edit Format Run Options Window Help

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
"identity": {
"orgId": "msfsb2",
"typeId": "Arun",
"deviceId": "123456"
},
"auth": {
"token": "123456789"
} }
client = wiotp.sdk.device.DeviceClient (config=myConfig,logHandlers=None)
client.connect ()
def myCommandCallback (cmd) :
    print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
    if (m=="motoron"):
        print ("Motor is switched on")
    elif (m=="motoroff"):
        print ("Motor is switched OFF")
    print (" ")
while True:
    soil=random.randint (0,100)
    temp=random.randint (-20, 125)
    hum=random.randint (0, 100)
    myData={'soilmoisture': soil, 'temperature':temp, 'humidity':hum}
    client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0 , onPublish=None)
    print ("Published data Successfully: %s", myData)
```

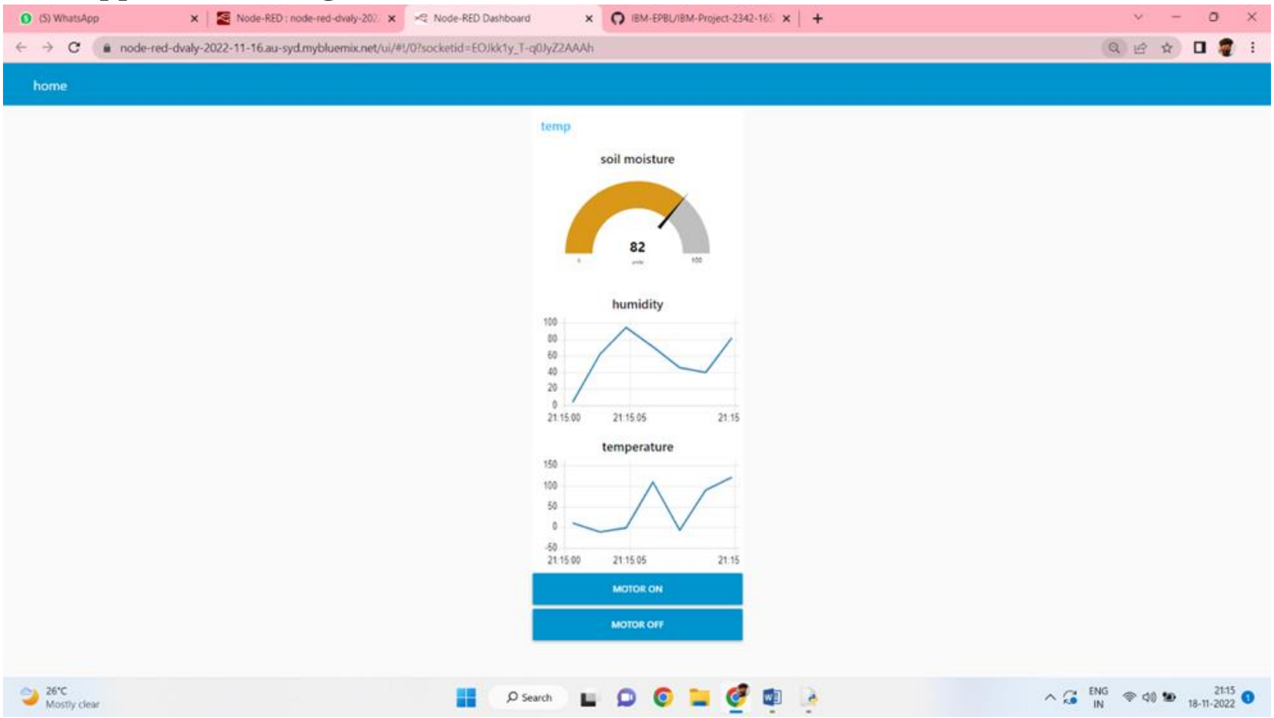
Ln: 13 Col: 0

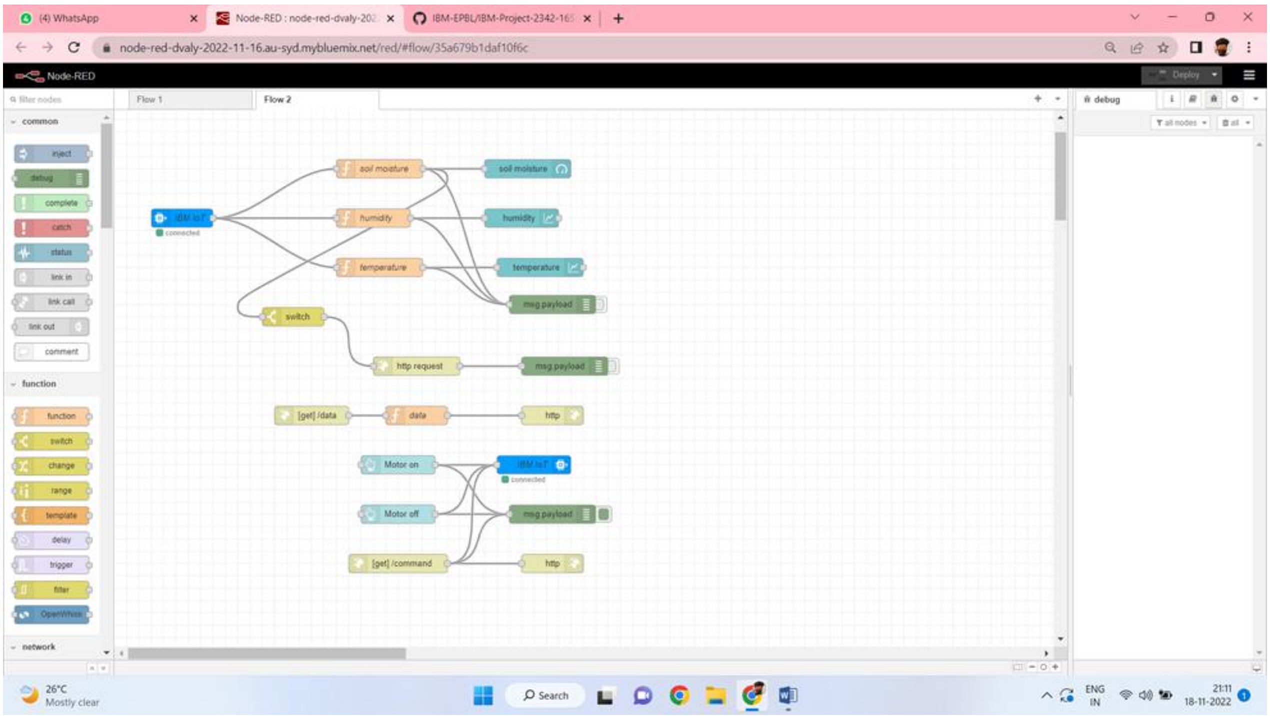
26°C Mostly clear Search ENG IN 21:52 18-11-2022

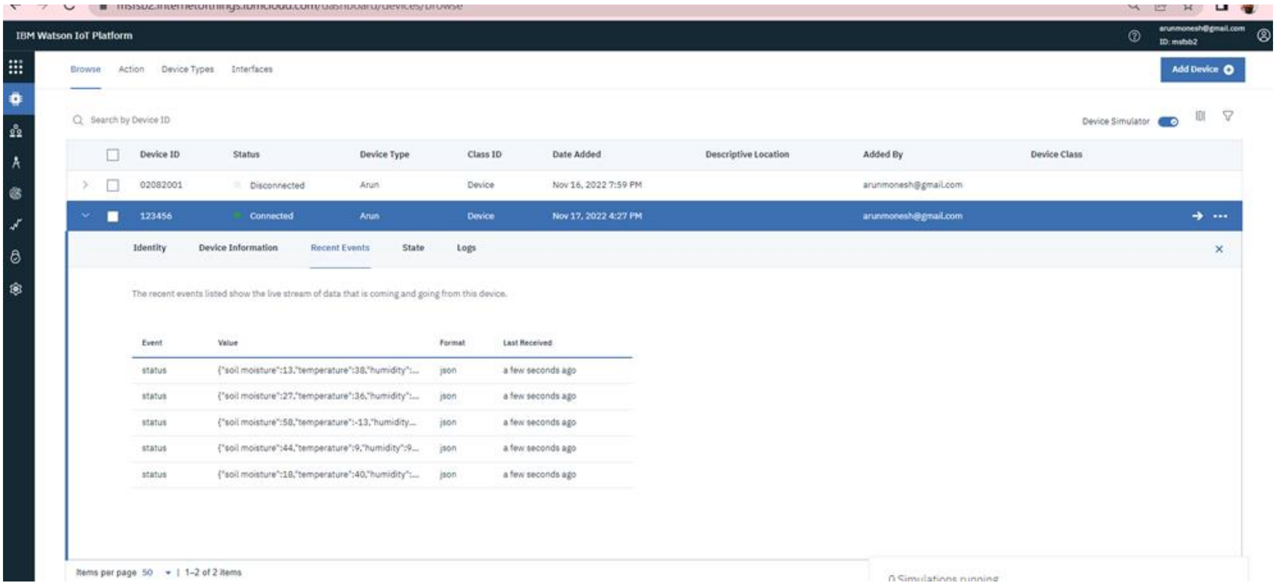
8.TESTING:

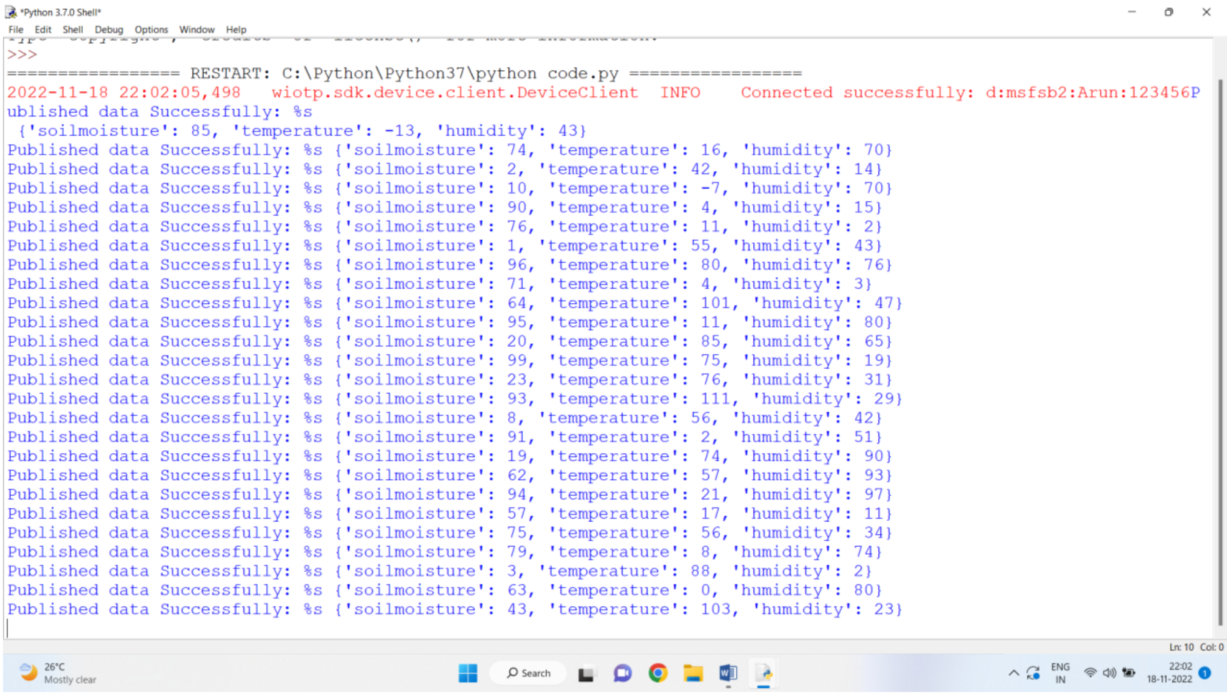
8.1 TEST CASE:

Web application using Node-RED.

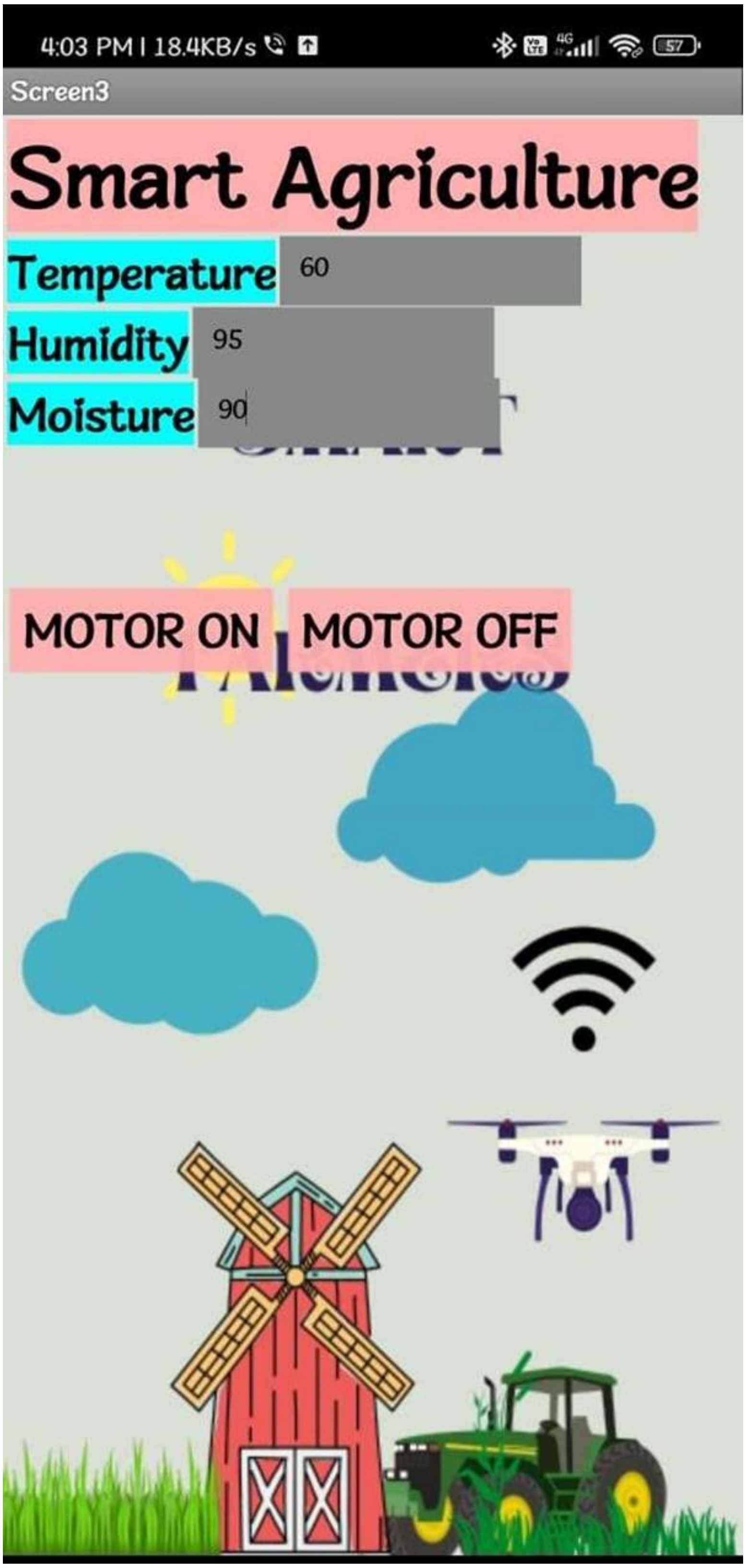





IBM Watson IoT Platform
Browse
Action
Device Types
Interfaces
Add Device
Search by Device ID
Device Simulator
Device ID | Status | Device Type | Class ID | Date Added | Descriptive Location | Added By | Device Class
02082001 | Disconnected | Arun | Device | Nov 16, 2022 7:59 PM | | arunmonesh@gmail.com
123456 | Connected | Arun | Device | Nov 17, 2022 4:27 PM | | arunmonesh@gmail.com
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event | Value | Format | Last Received
status | {"soil moisture":13,"temperature":38,"humidity":... | json | a few seconds ago
status | {"soil moisture":27,"temperature":36,"humidity":... | json | a few seconds ago
status | {"soil moisture":58,"temperature":-13,"humidity... | json | a few seconds ago
status | {"soil moisture":44,"temperature":9,"humidity":9... | json | a few seconds ago
status | {"soil moisture":18,"temperature":40,"humidity":... | json | a few seconds ago
Items per page 50 | 1–2 of 2 items

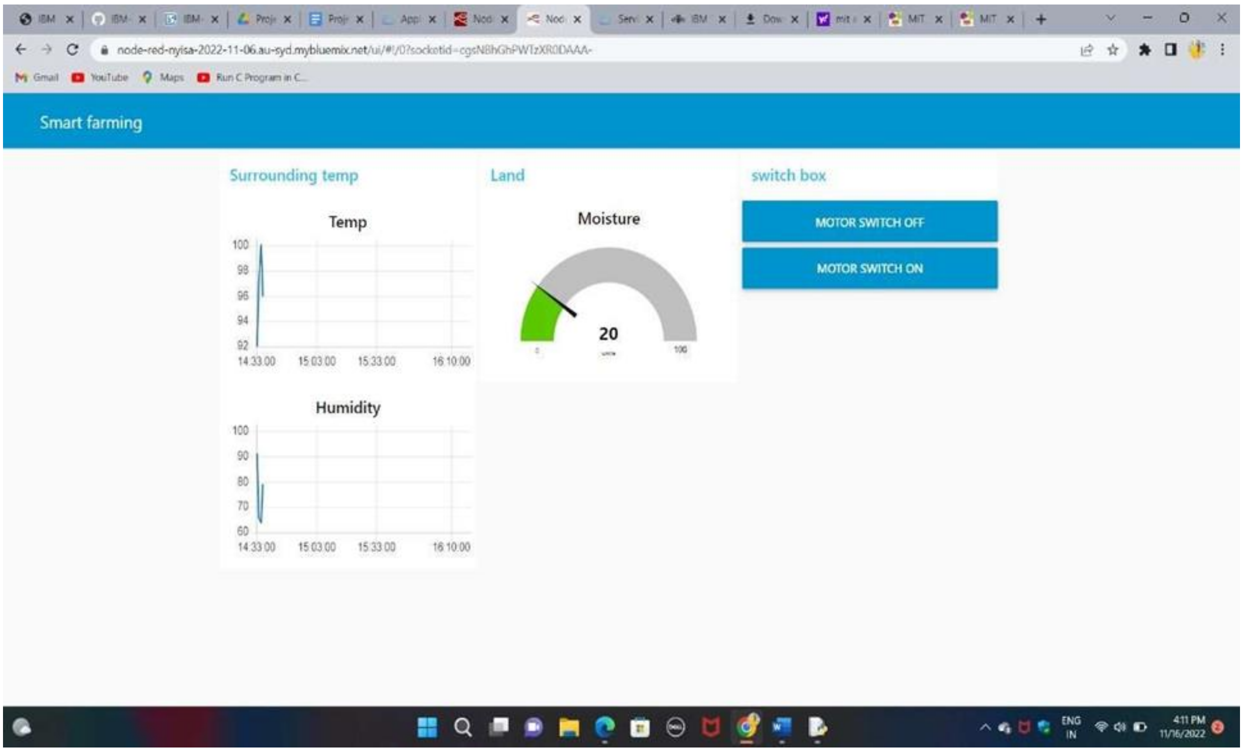

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>>
================= RESTART: C:\Python\Python37\python code.py =================
2022-11-18 22:02:05,498 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:msfsb2:Arun:123456P
ublished data Successfully: %s
{'soilmoisture': 85, 'temperature': -13, 'humidity': 43}
Published data Successfully: %s {'soilmoisture': 74, 'temperature': 16, 'humidity': 70}
Published data Successfully: %s {'soilmoisture': 2, 'temperature': 42, 'humidity': 14}
Published data Successfully: %s {'soilmoisture': 10, 'temperature': -7, 'humidity': 70}
Published data Successfully: %s {'soilmoisture': 90, 'temperature': 4, 'humidity': 15}
Published data Successfully: %s {'soilmoisture': 76, 'temperature': 11, 'humidity': 2}
Published data Successfully: %s {'soilmoisture': 1, 'temperature': 55, 'humidity': 43}
Published data Successfully: %s {'soilmoisture': 96, 'temperature': 80, 'humidity': 76}
Published data Successfully: %s {'soilmoisture': 71, 'temperature': 4, 'humidity': 3}
Published data Successfully: %s {'soilmoisture': 64, 'temperature': 101, 'humidity': 47}
Published data Successfully: %s {'soilmoisture': 95, 'temperature': 11, 'humidity': 80}
Published data Successfully: %s {'soilmoisture': 20, 'temperature': 85, 'humidity': 65}
Published data Successfully: %s {'soilmoisture': 99, 'temperature': 75, 'humidity': 19}
Published data Successfully: %s {'soilmoisture': 23, 'temperature': 76, 'humidity': 31}
Published data Successfully: %s {'soilmoisture': 93, 'temperature': 111, 'humidity': 29}
Published data Successfully: %s {'soilmoisture': 8, 'temperature': 56, 'humidity': 42}
Published data Successfully: %s {'soilmoisture': 91, 'temperature': 2, 'humidity': 51}
Published data Successfully: %s {'soilmoisture': 19, 'temperature': 74, 'humidity': 90}
Published data Successfully: %s {'soilmoisture': 62, 'temperature': 57, 'humidity': 93}
Published data Successfully: %s {'soilmoisture': 94, 'temperature': 21, 'humidity': 97}
Published data Successfully: %s {'soilmoisture': 57, 'temperature': 17, 'humidity': 11}
Published data Successfully: %s {'soilmoisture': 75, 'temperature': 56, 'humidity': 34}
Published data Successfully: %s {'soilmoisture': 79, 'temperature': 8, 'humidity': 74}
Published data Successfully: %s {'soilmoisture': 3, 'temperature': 88, 'humidity': 2}
Published data Successfully: %s {'soilmoisture': 63, 'temperature': 0, 'humidity': 80}
Published data Successfully: %s {'soilmoisture': 43, 'temperature': 103, 'humidity': 23}
Ln: 10 Col: 0
26°C
Mostly clear
Search
ENG
IN
22:02
18-11-2022

8.3 User Acceptance Testing



9. RESULT:

9.1 Performance Metrics



10.ADVANTAGES AND DISADVANTAGES:

10.1 ADVANTAGES:

- All the data like climatic conditions and changes in them, soil or crop conditions everything can be easily monitored.
- Risk of crop damage can be lowered to a greater extent.
- Many difficult challenges can be avoided making the process automated and the quality of crops can be maintained.
- The process included in farming can be controlled using the web applications from anywhere, anytime.

10.2 DISADVANTAGES:

- Smart Agriculture requires internet connectivity continuously, but rural parts cannot fulfil this requirement.
- Any faults in the sensors can cause great loss in the agriculture, due to wrong records and the actions of automated processes. IOT devices need much money to implement.

11.CONCLUSION:

An IOT based smart agriculture system using Watson IOT platform, Watson simulator, IBM cloud and Node-RED.

12.FUTURE SCOPE:

In future due to more demand of good and more farming in less time, for betterment of the crops and reducing the usage of extravagant resources like electricity and water IOT can be implemented in most of the places.

13.APPENDIX:

SOURCE CODE:

```
import wiotp.sdk.device import time import sys import ibmiotf.application
import ibmiotf.device import random

#Provide your IBM Watson Device organization = "msfsb2" deviceType =
"Arun" deviceId = "123456" authMethod = "token" authToken ="123456789"
# Initialize GPIO def myCommandCallback(cmd):     print("Commandreceived:
%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, "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
aneventof type "greeting" 10 times deviceCli.connect() while True:
#Get Sensor Data fromDHT11     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")
```

Github link : https://github.com/IBM-EPBL/IBM-Project-2342-1658470095

Project Demo link : https://photos.app.goo.gl/dinp7LBaa29jhn9k9

THANK YOU…..