IBM – NAALAIYA THIRAN

SmartFarmer - IoTEnabled Smart Farming Application

FINAL REPORT

Team ID:
PNT2022TMID15028

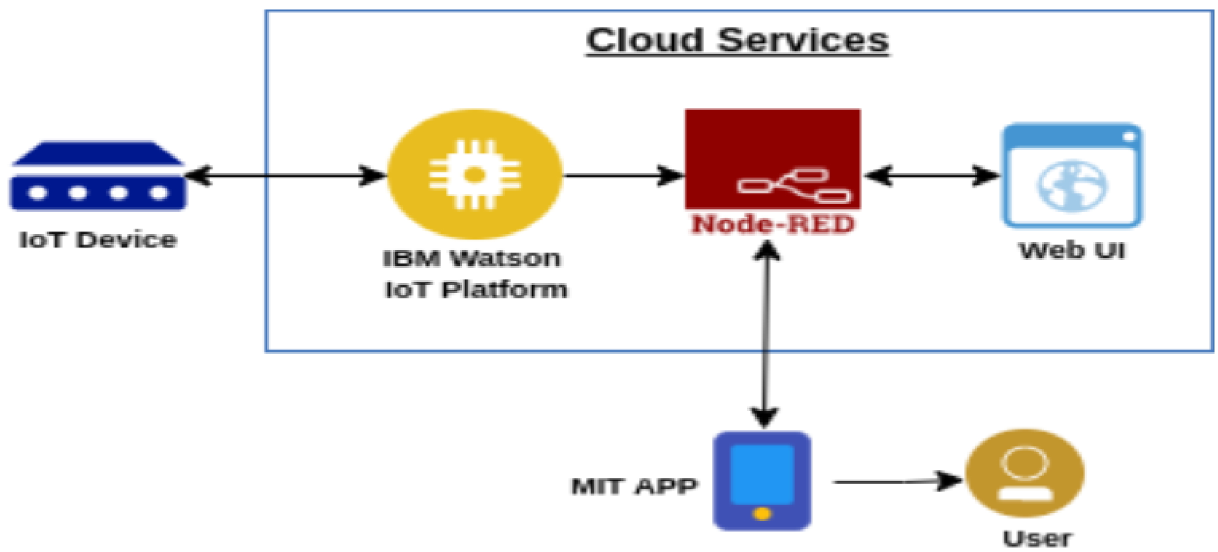
Team Leader:
VIJEY VARSHAN – 19BEC115

Team Members:
KOODALARASAN M 19BEC114
NARESH V H - 19BEC116
VISVABARATHI S -19BEC117

1. INTRODUCTION

1.1 Project Overview:

The IoT-based agriculture system allows farmers to monitor different parameters of their fields, including soil moisture, temperature, and humidity. Farmers can monitor these parameters by using a web or mobile application even if the farmer is not near his field. Watering the crop is one of the most important tasks for farmers; they can make decisions about whether or not to water their crops by monitoring sensor parameters and controlling motor pumps from their mobile device.



1.2 Purpose:

The main aim of this smart agriculture model is to avoid water wastage in the irrigation process. It is low cost and efficient system Is shown below. It includes NodeMCU, Arduino Nano, sensors like soil moisture and Dht11, solenoid valves, relays.

2. LITERATURE SURVEY

2.1 Existing problem:

Smart agriculture systems require an extensive infrastructure of sensors to collect data from the environment, which then need to be processed by analytics software and automated responses implemented. To successfully deploy a smart agriculture system, consider setting up a communications network that can integrate a limited number of sensors across a large area of farmland.

This will require third-party network provisioning or setting up a private network consisting of access points and uplinks to a private backhaul network, which channels all the data traffic to centralized monitoring software or an analytics head-end system.

- The system is not secure
- It lacks motion detection and automation, which are essential for protection of agricultural fields.
- Automation is not available.

2.2 References:

[1]Rohith, M., Sainivedhana, R., & Fatima, N. S. (2021, May). IoT Enabled Smart Farming and Irrigation System. In *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 434-439). IEEE.

[2] Doshi, J., Patel, T., &kumar Bharti, S. (2019). Smart Farming using IoT, a solution for optimally monitoring farming conditions. *Procedia Computer Science*, *160*, 746-751.

[3]Quy, V. K., Hau, N. V., Anh, D. V., Quy, N. M., Ban, N. T., Lanza, S., ... &Muzirafuti, A. (2022). IoT-Enabled Smart Agriculture: Architecture, Applications, and Challenges. Applied Sciences, 12(7), 3396.

[4]Idoje, G., Dagiuklas, T., & Iqbal, M. (2021). Survey for smart farming technologies: Challenges and issues. Computers & Electrical Engineering, 92, 107104.

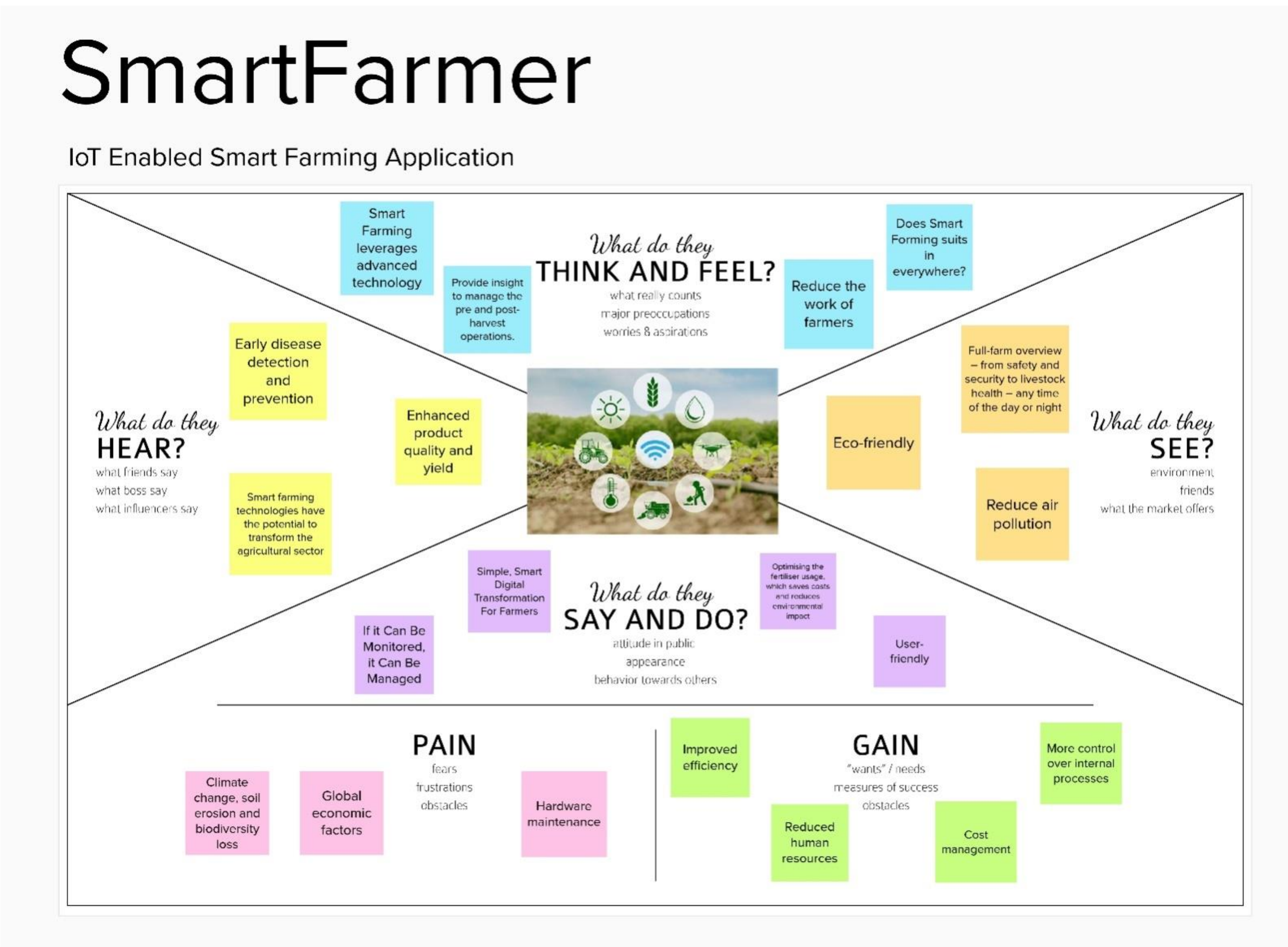
[5]Keerthana, B., Nivetha, P., Boomika, M., Mathivathani, M., & Niranjana, A. (2018). IoT based smart security and monitoring devices for agriculture. Int. J. Inf. Res. Rev, 5(04), 5415-5419.

2.3 Problem Statement Definition:

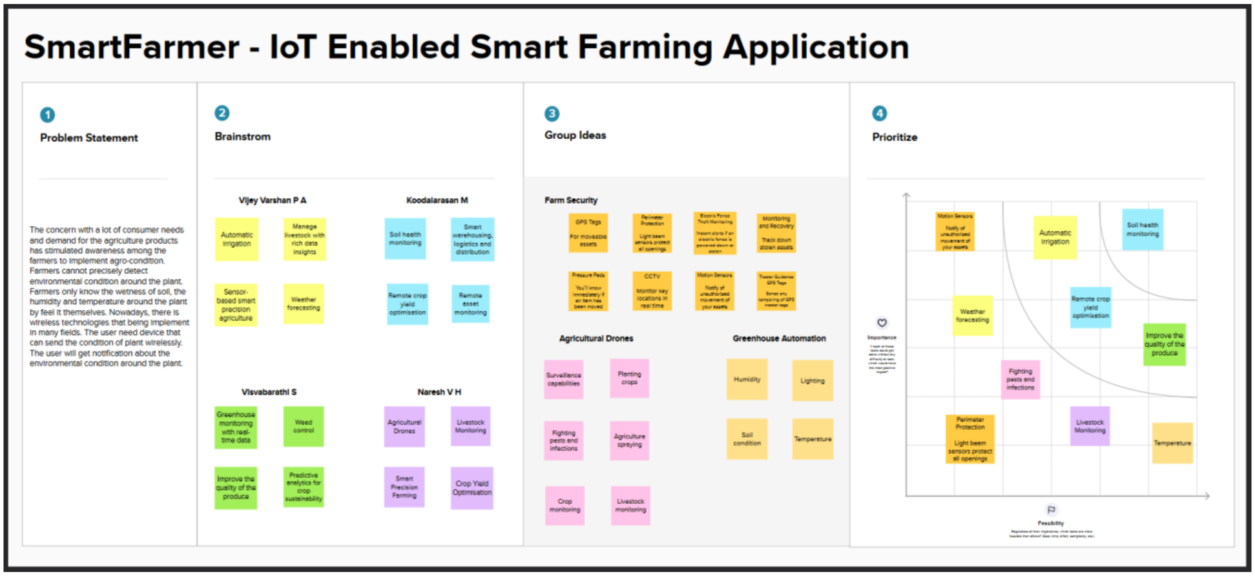
To provide a effective decision support system farming with a wireless sensor network that manages various agricultural activities and provides pertinent farm information like temperature,humidity, soil moisture content,motion detection and pH level detector.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation and Brainstorming:



3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To provide a effective decision support system farming with a wireless sensor network that manages various agricultural activities and provides pertinent farm information like temperature, humidity, soil moisture content, motion detection and pH level detector. There are many factors which are bad for agriculture.
2.	Idea / Solution description	An integrated IoT platform for agriculture is provided by the SmartFarmer - IoT Enabled Smart Farming Application. This platform enables farmers to use sensors, smart gateways, and monitoring systems to gather data, manage numerous agricultural factors, and analyse real-time data to make optimal decisions that yields high produce.

3.	Novelty / Uniqueness	The use of IoT principles in agriculture has been the focus of many renowned scientists working toward smart agriculture. But there are still a variety of issues that haven't yet found an appropriate solution. This study makes an effort to explain prior research and unsettled issues in IoT-based agriculture.
4.	Social Impact / Customer Satisfaction	By reducing the quantity of fertiliser used on the field, which lowers costs considerably, makes it easier to monitor the field, and lowers pay for agricultural labourers. It helps you save lots of time. By boosting the client experience overall, Our application can assist in enhancing customer connections.
5.	Business Model (Revenue Model)	Farmers will be required to pay a monthly subscription for the service of irrigation timing recommendation and forecast based on sensor characteristics such as temperature, humidity, and soil moisture even when an intruder is present in the fields. Through the promotion of the custom manufacturing of raw materials for the desired condition, which will be described in terms of easy home recipes or they can order it if it in mass production.
6.	Scalability of the Solution	It is the ability to adapt to a systematic increase in the capacity of a large market, for instance, the number of technology devices such as sensors and actuators, while enabling timely analysis will yield organic or high end product in mass production that can also be sold with the affiliation of our brand which brings marketing reach and also the need of people for good produces, which in turn returns huge profit .

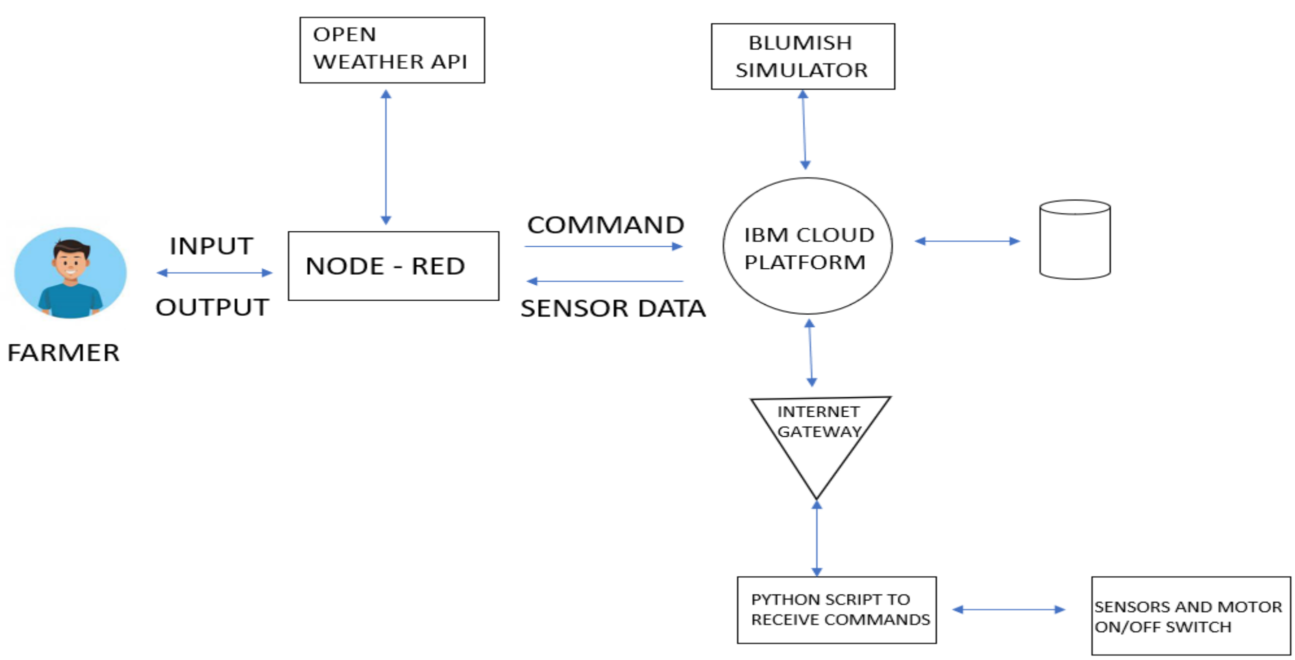
4. REQUIREMENT ANALYSIS

4.1 Functional Requirement:

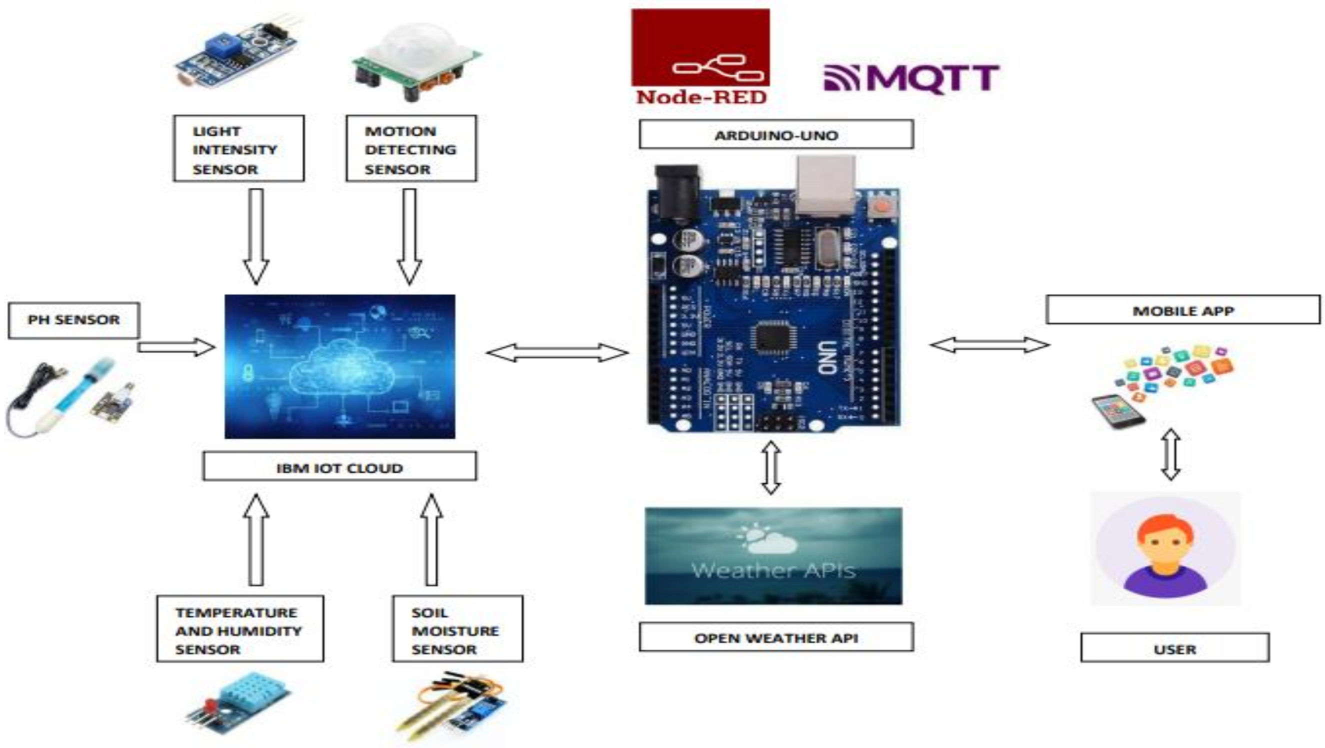
FR No.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	User Registration	As a user Registration through Gmail.
FR-2	User Confirmation	As a user Confirmation via Email then generate the Confirmation via OTP.
FR-3	Log in to system	Once confirmation message received after login the system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

5. PRODUCT DESIGN

5.1 Data flow diagrams:



5.2 Solution and Technical Architecture:



Guidelines:

- The different soil parameters (Temperature, Humidity, Light Intensity, Ph level, Soil Moisture and Motion) are sensed using different sensors and the obtained value is stored in IBM cloud.
- Arduino UNO is used as a processing unit which processes the data obtained from sensors and weather data from weather API.
- Node red is used as a programming tool to wire the hardware, software and APIs. The MQTT protocol is followed for communication.
- All the collected data are provided to the user through a mobile application which was developed using MIT app inventor. The user could make decision through an app, whether to water the crop or not depending upon the sensor values.
- Increased control over production allows for better cost management. In addition, being alerted early to any anomalies in crop growth or livestock health allows farmers to mitigate costly risks.
- With more control (even remotely), farmers can reduce risks and plan better. For example, If you know exactly how large your yield is going to be, you can ensure that you find enough buyers and your product won't lie around unsold.

6. PROJECT PLANNING AND SCHEDULING

Sprint Schedule, and Estimation:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	creation		Arduino with python code.			Vijey Varshan
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red.	2	High	Koodalarasan, Vijey Varshan
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using MIT App Inventor.	2	High	Koodalarasan, Naresh
Sprint-3	Dashboard	USN-3	Design the Modules and test the app.	2	High	Koodalarasan, Naresh
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Visvabarathi, Vijey Varshan

7. CODING AND SOLUTIONING

Code:

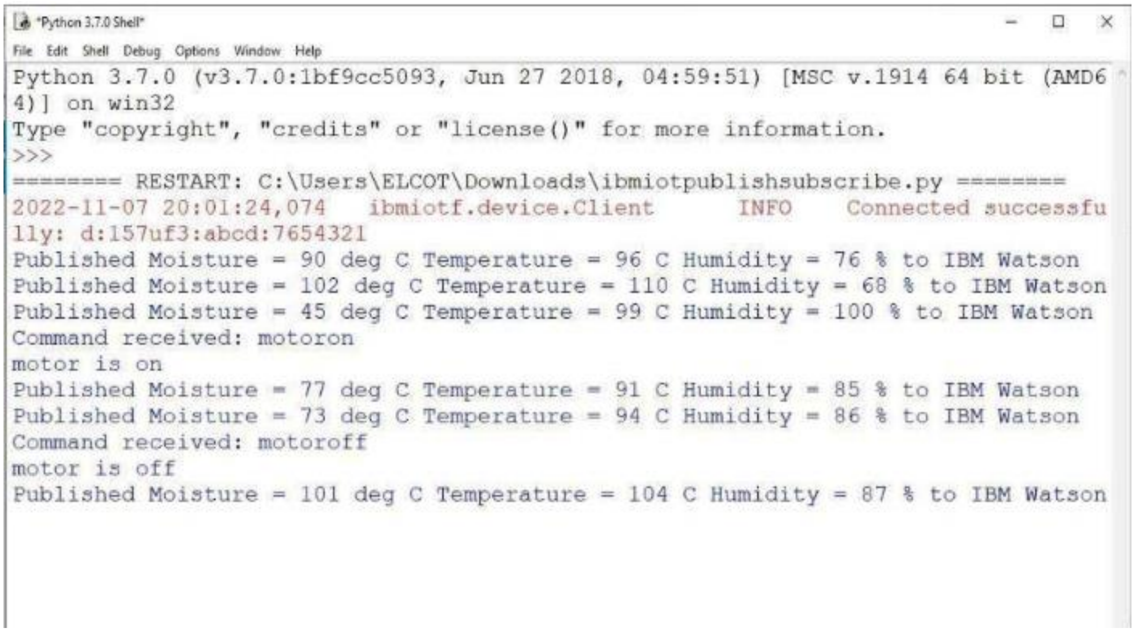
```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
  "identity": {
    "orgId" :"vo6jfg",
    "typeId":"Arduino",
    "deviceId":"2266"
    },
    "auth": {
      "token":"12345678"}}
client=wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect()
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(90,125)
    hum=random.randint(0,100)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}
    client.publishEvent(eventId="status",msgFormat="json" ,data=myData
,qos=0,onPublish=None)
    print("published data successfully: %s",myData)
    time.sleep(5)
    client.commandCallback=myCommandCallback
client.disconnect()
```

code1.py - C:\Users\Asus\Documents\AEIC\code1.py (3.7.4)

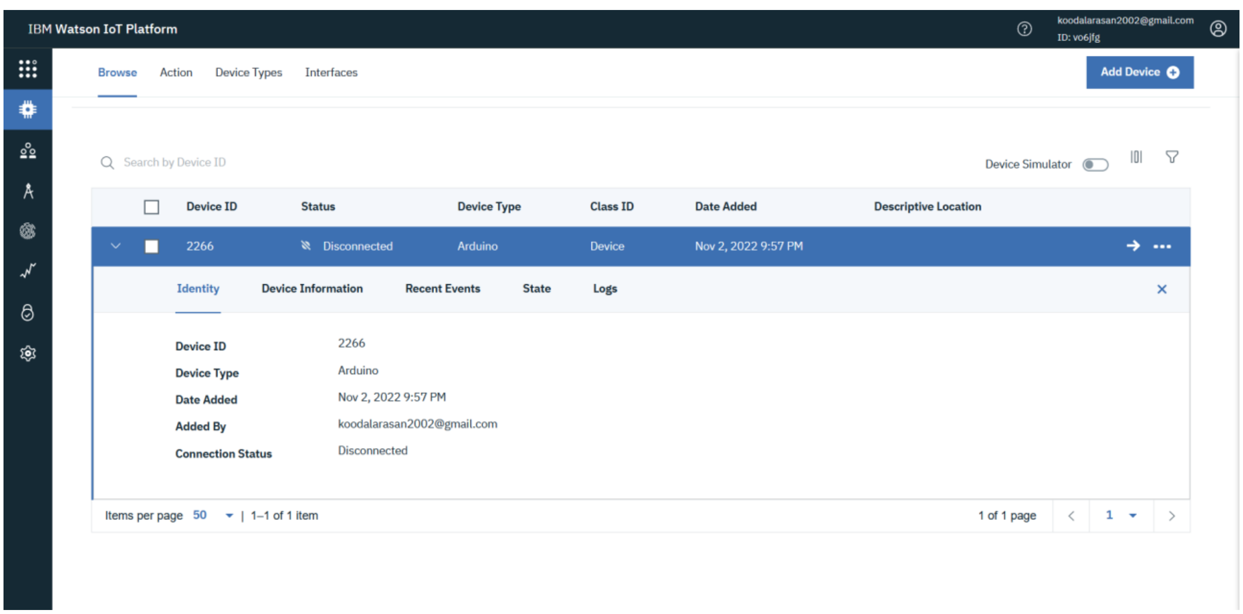
File Edit Format Run Options Window Help

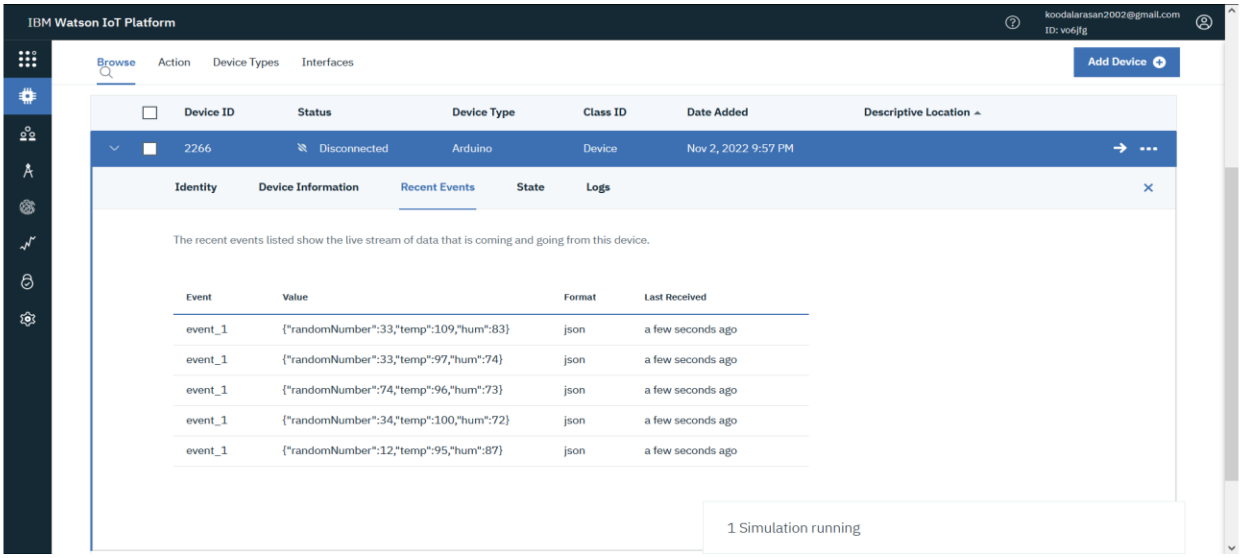
```
import wiotp.sdk.device
import time
import os
import datetime
import random

myConfig = {
    "identity": {
        "orgId" :"vo6jfg",
        "typeId":"Arduino",
        "deviceId":"2266"
        },
        "auth": {
            "token":"12345678"
            }
}
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(90,125)
    hum=random.randint(0,100)
    myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}
    client.publishEvent(eventId="status",msgFormat="json" ,data=myData ,qos=0,onPublish=None)
    print("published data successfully: %s",myData)
    time.sleep(5)
    client.commandCallback=myCommandCallback
client.disconnect()
```

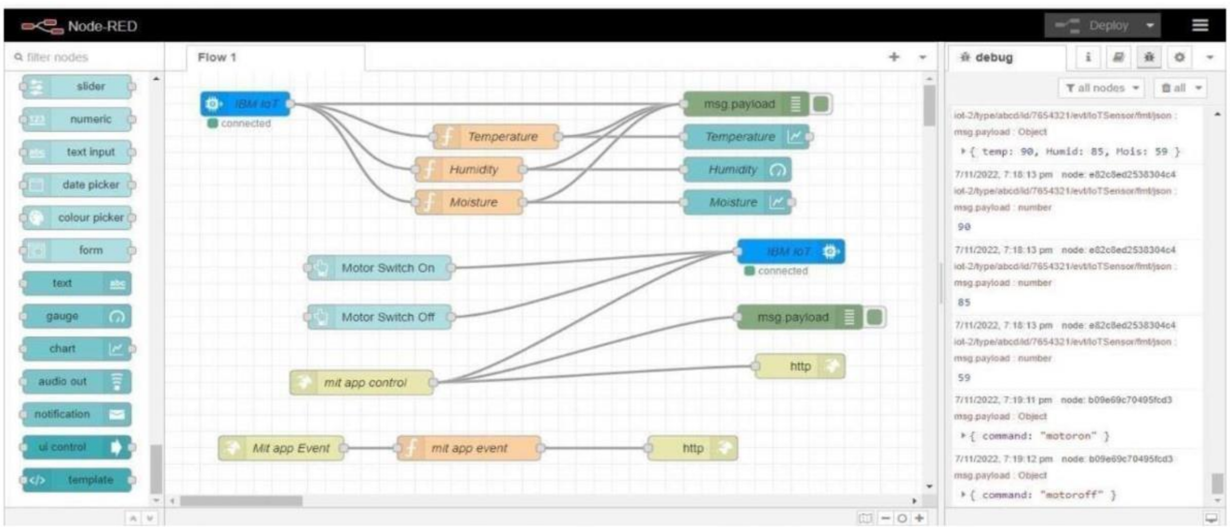


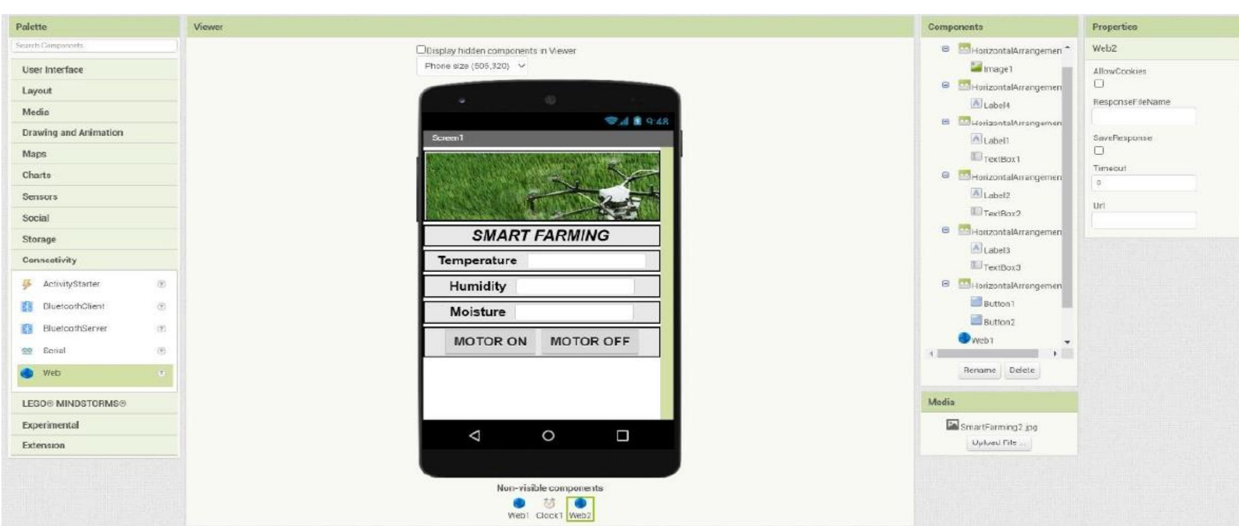
8. TESTING

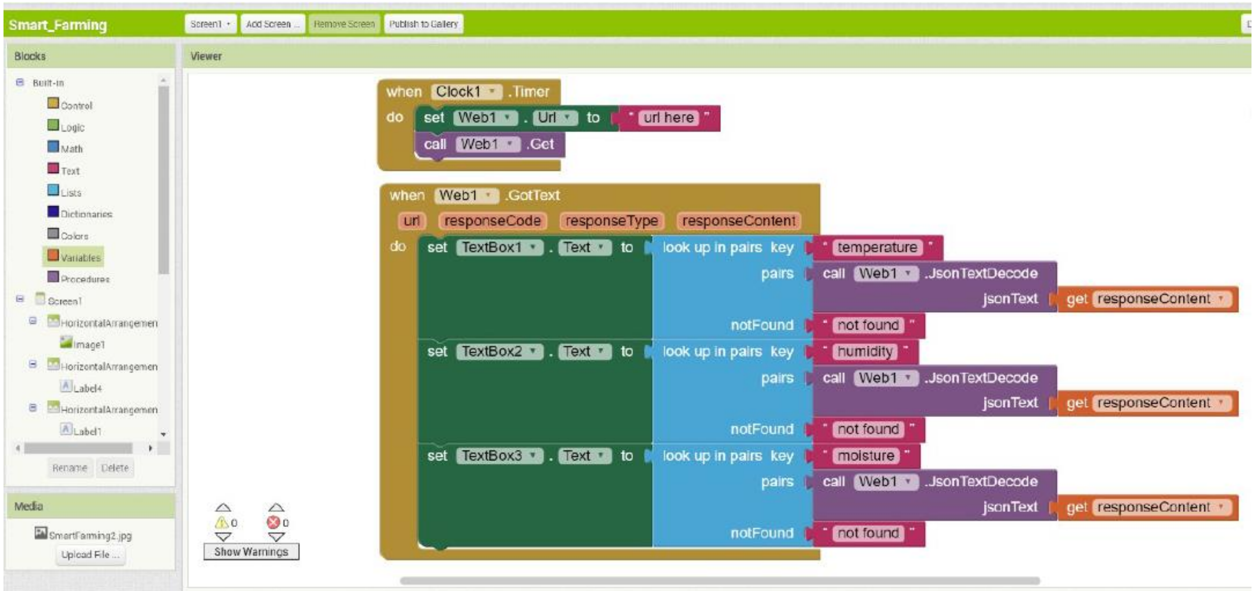
8.1 Test Cases:



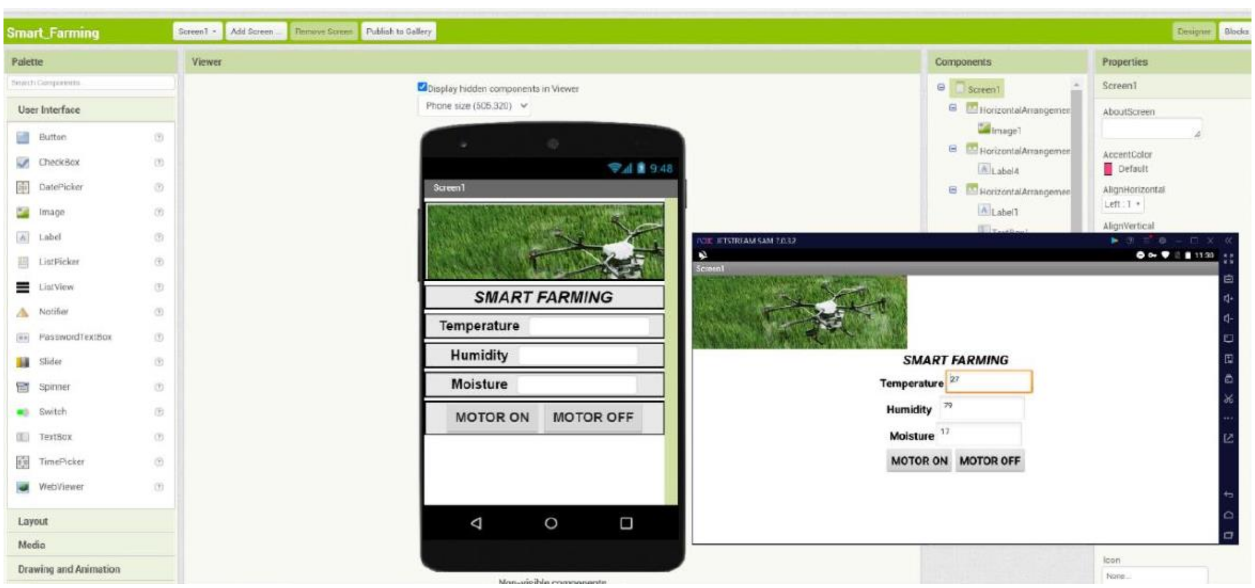

IBM Watson IoT Platform
koodalarasan2002@gmail.com
ID: vo6jfg
Browse
Action
Device Types
Interfaces
Add Device
Device ID
Status
Device Type
Class ID
Date Added
Descriptive Location
2266
Disconnected
Arduino
Device
Nov 2, 2022 9:57 PM
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
event_1 {"randomNumber":33,"temp":109,"hum":83} json a few seconds ago
event_1 {"randomNumber":33,"temp":97,"hum":74} json a few seconds ago
event_1 {"randomNumber":74,"temp":96,"hum":73} json a few seconds ago
event_1 {"randomNumber":34,"temp":100,"hum":72} json a few seconds ago
event_1 {"randomNumber":12,"temp":95,"hum":87} json a few seconds ago
1 Simulation running


Node-RED
Deploy
Flow 1
debug
slider
numeric
text input
date picker
colour picker
form
text
gauge
chart
audio out
notification
ui control
template
msg.payload
Temperature
Humidity
Moisture
Motor Switch On
Motor Switch Off
mit app control
http
Mit app Event
mit app event


Palette
Viewer
Components
Properties
User Interface
Layout
Media
Drawing and Animation
Maps
Charts
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Experimental
Extension
Display hidden components in Viewer
Screen1
SMART FARMING
Temperature
Humidity
Moisture
MOTOR ON
MOTOR OFF
Non-visible components
Web2
Media
SmartFarming1.jpg

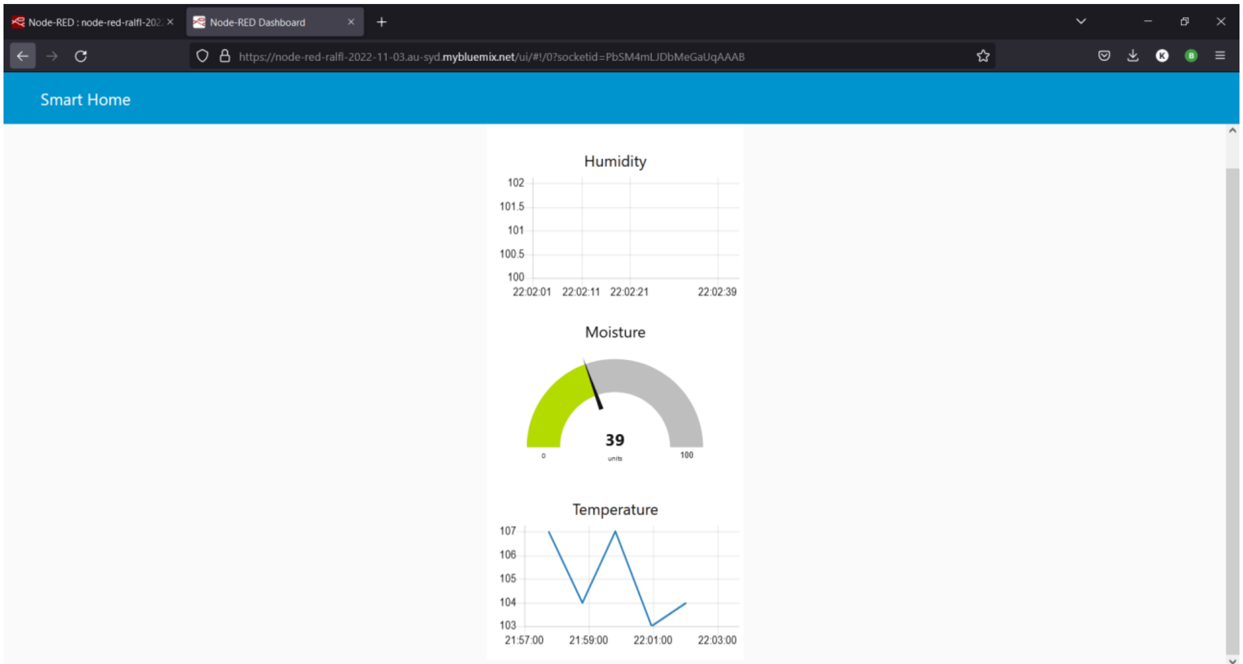


8.2 User Acceptance Testing:



9. RESULTS

Performance Metrics:



10. ADVANTAGES & DISADVANTAGES

Advantages:

➢ A remote control system can help in working irrigation system valves dependent on schedule. Irrigating remote farm properties can be exceptionally troublesome and labor-intensive. It gets hard to comprehend when the valves were started and whether the ideal measure of water was distributed.

➢ For situations where a quick reaction is required, manual valve actuation may not be conceivable constantly. Thus, remote observing and control of irrigation systems, generators or wind machines or some other motor-driven hardware become the next logical step.

➢ Various solutions are available to monitor engine statistics and starting or stopping the engine. When the client chooses to begin or stop the motor, the program transmits a sign to the unit within seconds by means of a mobile phone system.

➢ Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.

Disadvantages:

➢ The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.

➢ The smart farming based equipment require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.

11. CONCLUSION

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted dependent on the local environmental variables. This technology also detects animal invasions, which are a major cause of crop loss. This technology aids in the scheduling of irrigation based on present data from the field and records from a climate source. It helps in deciding the farmer to whether to do irrigation or not to do. Continuous internet connectivity is required for continuous monitoring of data from sensors. This also can be overcome by using GSM unit as an alternative of mobile app. By GSM, SMS can be sent to farmer's phone.

12. FUTURE SCOPE

In the current project we have implemented the project that can protect and maintain the crop. In this project the farmer monitor and control the field remotely. In future we can add or update few more things to this project.

• We can create few more models of the same project ,so that the farmer can have information of a entire.

• We can update the this project by using solar power mechanism. So that the power supply from electric poles can be replaced with solar panels. It reduces the power line cost. It will be a one time investment. We can add solar fencing technology to this project.

• We can use GSM technology to this project so that the farmers can get the information directly to his home through SMS. This helps the farmer to get information if there is a internet issues.

• We can add camera feature so that the farmer can monitor his field in real time. This helps in avoiding thefts.

13. APPENDIX

Source Code:

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
  "identity": {
    "orgId" :"vo6jfg",
    "typeId":"Arduino",
    "deviceId":"2266"
    },
    "auth": {
      "token":"12345678"
     }
}
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(90,125)
  hum=random.randint(0,100)
  myData={'soil_moisture':soil,'temperature':temp,'humidity':hum}
  client.publishEvent(eventId="status",msgFormat="json" ,data=myData
,qos=0,onPublish=None)
  print("published data successfully: %s",myData)
  time.sleep(5)
  client.commandCallback=myCommandCallback
client.disconnect()
```

Github link: https://github.com/IBM-EPBL/IBM-Project-12118-1659373791

Project Demo link:
https://drive.google.com/file/d/1uJtfPRwbdLQ0AZwbFaVtEs5CR7D7vYOd/view?usp=share_link