SMART FARMER - IOT ENABLED SMART FARMING APPLICATION

A PROJECT REPORT

Submitted by

TEAM ID : PNT2022TMID07205

BAVANI V (130719205011)

KISHORE V (130719205025)

JEEVANANDHAM V (130719205022)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

JERUSALEM COLLEGE OF ENGINEERING,

(An Autonomous Institution, Affiliated to Anna University, Chennai)

PALLIKARANI, CHENNAI-100

ANNA UNIVERSITY: CHENNAI 600 025

NOVEMBER 2022

SMART FARMER - IOT ENABLED SMART FARMING APPLICATION

1. INTRODUCTION

The main aim of this project is to help farmers automate their farms by providing them with a Web App through which they can monitor the parameters of the field like Temperature, soil moisture, humidity and etc and control the equipment like water motor and other devices remotely via internet without their actual presence in the field.

1.1 PROJECT OVERVIEW

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 will be 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 parameter, farmer can water his crop by controlling the motor using the mobile application. Thus even if the farmer is not present near his crop he can water his crop by controlling the motors using the application from anywhere.

1.2 PURPOSE

By making farming more connected and intelligent, precision agriculture helps reduce overall costs and improve the quality and quantity of products, the sustainability of agriculture and the experience for the consumer. 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. With smart devices, multiple processes can be activated at the same time, and automated services enhance product quality and volume by better controlling production processes

2. LITERATURE SURVEY

2.1 EXISTING PROBLEMS AND REFERENCES

SMART FARMING (IoT ENABLED SMART FARMING APPLICATION) - - LITERATURE SURVEY				
S.NO	**TITLE OF THE JOURNAL**	**AUTHOR NAME**	**JOURNAL NAME**	**DESCRIPTION**
1.	Agri-IoT: A semantic framework for internet of Things-enabled smart farming applications	Andreas Kamilaris; Feng Gao; Francesc X. Prenafeta-Boldu;	IEEE	Agri-IoT, a semantic framework for IoT-based smart farming applications, which supports reasoning over various heterogeneous sensor data streams in real-time.
2.	A Low-Cost Information Monitoring System for Smart Farming Applications	Muhamamd saqib; Tarik Adnan Almohamad; Raja Majid Mehmood	MDPI	A low-cost, low-power, and low data-rate solution is proposed to fulfill the requirements of information monitoring for actual large-scale agricultural farms. a tree-based communication mechanism is deployed to extend the communication range by adding intermediate nodes. Each sensor node consists of a solar panel, a rechargeable cell, a microcontroller, a moisture sensor, and a communication unit.
3.	Smart farming for improving agricultural management	ElsayedSaid Mohamed; Mohamed BZahran[a];	The Egyptian Journal of Remote Sensing and Space Science	The smart irrigation system included those sensors for monitoring water level, irrigation efficiency, climate, etc. Smart irrigation is based on smart controllers and sensors as well as some mathematical relations.
4.	An Architecture model for Smart Farming	Anna Triantafyllou; Dimosthenis C.Tsouros; Panagiotis Sarigiannidis; Stamatia Bibi	IEEE	To guide the process of designing and implementing Smart farming monitoring systems, in this paper propose a generic reference architecture model, taking also into consideration a very important non-functional

				requirement, the energy consumption restriction.
5.	Smart Farming – IoT in Agriculture	Rahul Dagar; Subhranil Som; Sunil Kumar Khatri	IEEE	In this paper they surveyed typical agriculture methods used by farmers these days and what are the problems they face, they visited poly houses for further more information about new technologies in farming. The proposed model is a simple architecture of IoT sensors that collect information and send it over the Wi-Fi network to the server, there server can take actions depending on the information.
6.	Design and implementation of a connected farm for smart farming system	Minwoo Ryu; Jaeseok Yun; Ting Miao; Il-Yeup Ahn; Sung-Chan Choi;	IEEE	In this paper, we present a connected farm based on IoT systems, which aims to provide smart farming systems for end users.
7.	Big Data in Smart Farming – A review	SjaakWolfert; Marc-JeroenBogaardta	ELSEVIER	This review aims to gain insight into the state-of-the-art of Big Data applications in Smart Farming and identify the related socio-economic challenges to be addressed.
8.	A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming	Muhammad Shoaib Farooq; Kamran Abid; Muhammad Azhar Naeem	IEEE	The article presents many aspects of technologies involved in the domain of IoT in agriculture. It explains the major components of IoT based smart farming. A rigorous discussion on network technologies used in IoT based agriculture has been presented, that involves network architecture and layers, network topologies used, and protocols.
9.	The Digitisation of Agriculture: a	ManlioBacco;	ELSEVIER	In this work, they provide a survey of the most recent

	Survey of Research Activities on Smart Farming	Massimiliano Ruggerib		research activities, in the form of both research projects and scientific literature, with the objective of showing the already achieved results, the current investigations, and the still open challenges, both technical and non technical.
10.	Experimental validation of a wireless system for the irrigation management in smart farming applications	Federico Viani	Microwave and Optical Technology Letters	The proposed system has been prototyped and experimentally validated in an apple orchard, close to the city of Trento, in the north of Italy.

2.2 PROBLEM DEFINITION STATEMENT

- ✓ Farmer are to be present at form for its maintenance irrespective of the weather conditions.
- ✓ They have to ensure that the crops are well irrigated and the farm status is monitored by them physically.
- ✓ Farmer have to stay most of the time in field in order to get a good yield.
- ✓ In difficult times like in the presence of pandemic also they have to work hard in their fields risking their lives to provide food for the country.

CUSTOMER PROBLEM STATEMENT

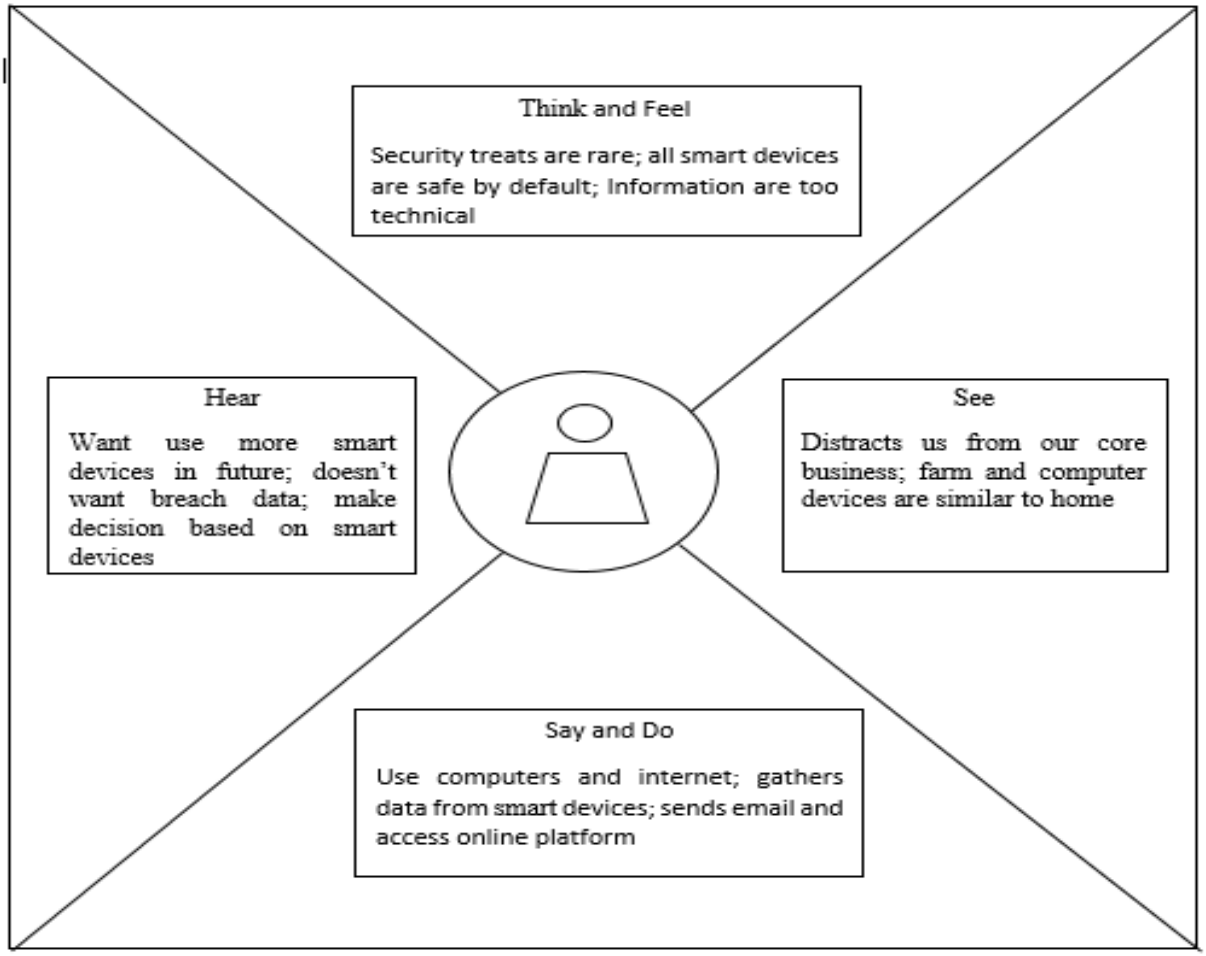
Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Farmer	Control the farm from anywhere of the world	It doesn't have any way for that	Internet bandwidth is low in villages	Frustrated
PS-2	Farmer	Visualize the farm from home	It doesn't have any way for that	It is expensive	wealthless
PS-3	Farmer	Visualize, control and operate the farm from home or anywhere	It doesn't have any way for that	That facilities are not available in village and also for big farms	Restless and unhealthy

		from the world			

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Pain:	Goal:
✓ Hard to get a good internet speed ✓ Heath issues ✓ Uncomfortable for paying cash ✓ Looking for application that matches interest	✓ Maintain a good work-life balance ✓ Provide support to farmers ✓ Comfortable ✓ Accuracy

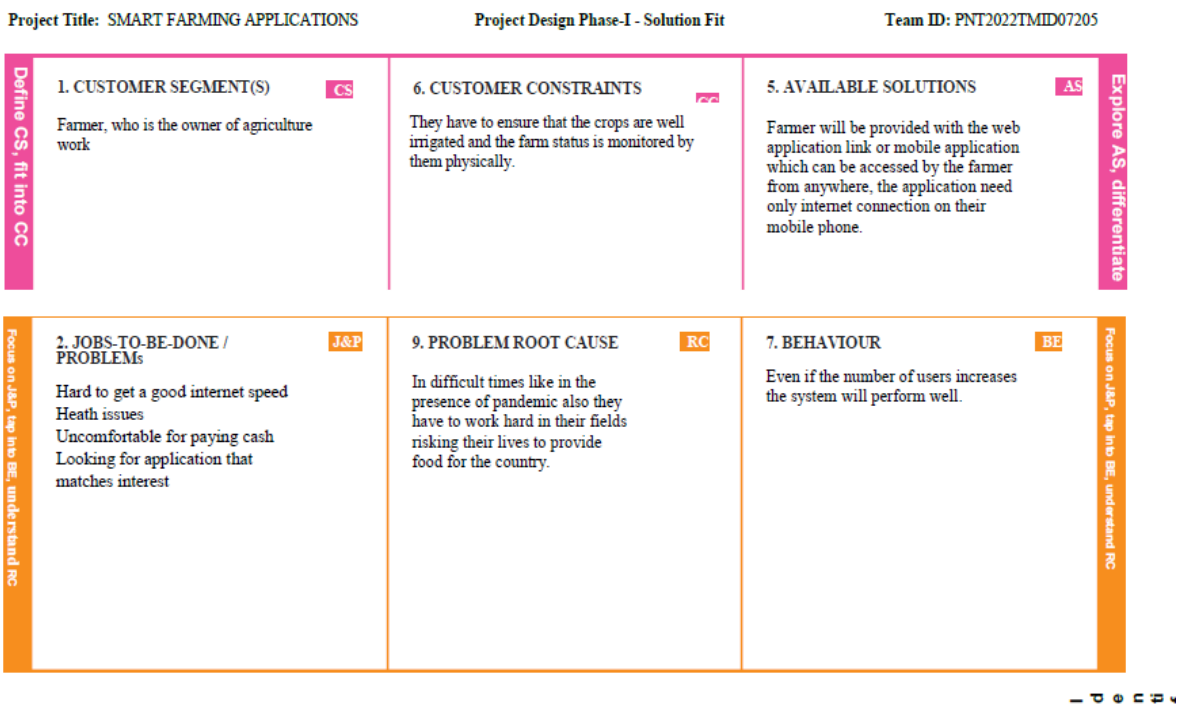


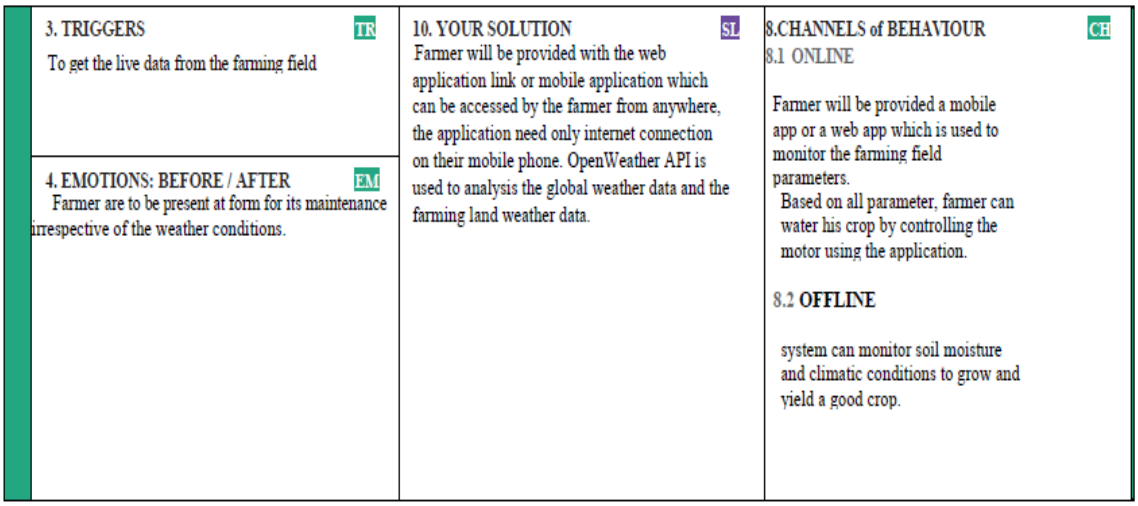
3.2 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	✓ Farmer are to be present at form for its maintenance irrespective of the weather conditions. ✓ They have to ensure that the crops are well irrigated and the farm status is monitored by them physically. ✓ Farmer have to stay most of the time in field in order to get a good yield. ✓ In difficult times like in the presence of pandemic also they have to work hard in their fields risking their lives to provide food for the country. ✓ Hence there is a need of **smart farming application**
2.	Idea / Solution description	✓ This system can monitor soil moisture and climatic conditions to grow and yield a good crop. ✓ This can also get the real time weather forecasting data by using external platforms like Open Weather API. ✓ Farmer will be provided a mobile app or a web app which is used to monitor the farming field parameters. ✓ Based on all parameter, farmer can water his crop by controlling the motor using the application. ✓ Even in the absence of farmer near the field, Farmer can able to monitor the field using the application from anywhere.
3.	Novelty / Uniqueness	✓ Farmer will be provided with the web application link or mobile application which can be accessed by the farmer from anywhere, the application need only internet connection on their mobile phone. ✓ OpenWeather API is used to analysis the global weather data and the farming land weather data.

4.	Social Impact / Customer Satisfaction	✓ The main aim of the project is to help farmers by providing them with a web or mobile application. ✓ By using the application farmer can monitor all the parameters of the field such as temperature, humidity, soil moisture, etc… ✓ And even they can be able to control the equipments the fields like water motors and other devices remotely via internet.
5.	Scalability of the Solution	✓ Even if the number of users increases , the system will perform well.

3.4 PROBLEM SOLUTION FIT





4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT AND USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation	I can receive confirmation	High	Sprint-1

			email once I have registered for the application	email & click confirm		
		USN-3	As a user, I can register for the application through Mobile number	I can register & access the dashboard with mobile number Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user , I can know the parameters values through dashboard	, I can know the parameters values through dashboard	High	Sprint-1

	Access Motor Control	USN-7	As a user , I can manipulate motor from anywhere	I can manipulate motor from anywhere	High	Sprint-1
Customer (Web user)	Registration	USN-8	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-9	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-10	As a user, I can register for the application through Mobile number	I can register & access the dashboard with mobile number Login	Low	Sprint-2
		USN-11	As a user, I can register	I can register & access the	Medium	Sprint-1

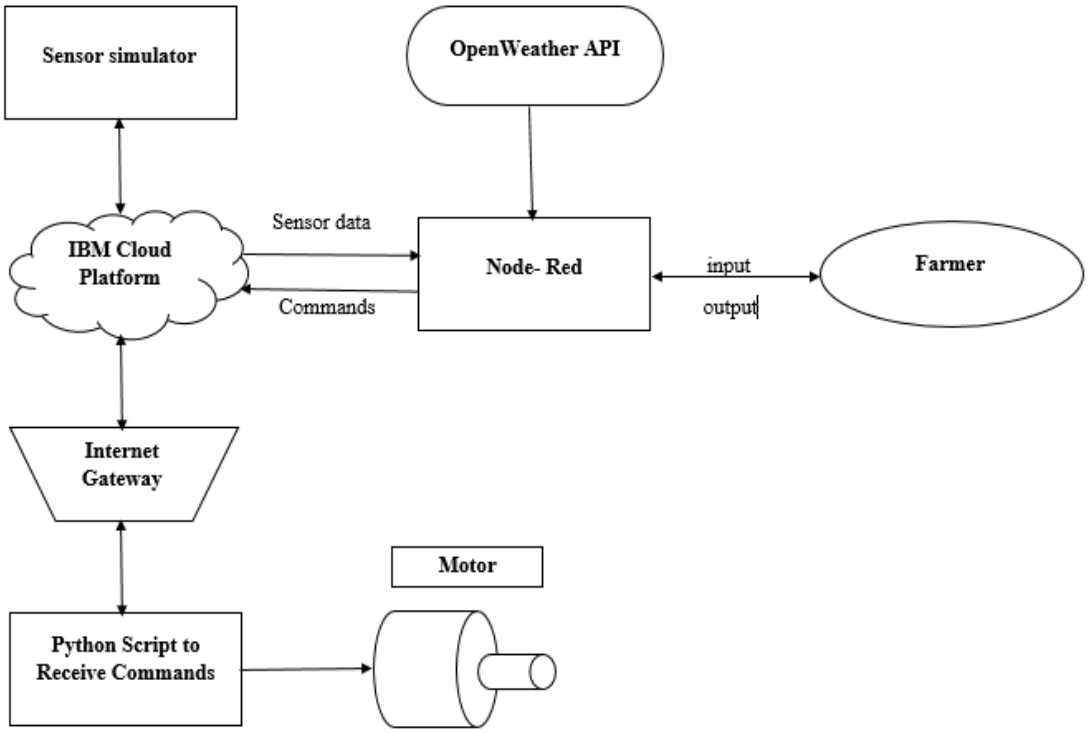
			for the application through Gmail	dashboard with Gmail		
	Login	USN-12	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-13	As a user , I can know the parameters values through dashboard	, I can know the parameters values through dashboard	High	Sprint-1
	Access Motor Control	USN-14	As a user , I can manipulate motor from anywhere	I can manipulate motor from anywhere	High	Sprint-1

4.2 NON- FUNCTIONAL REQUIREMENTS

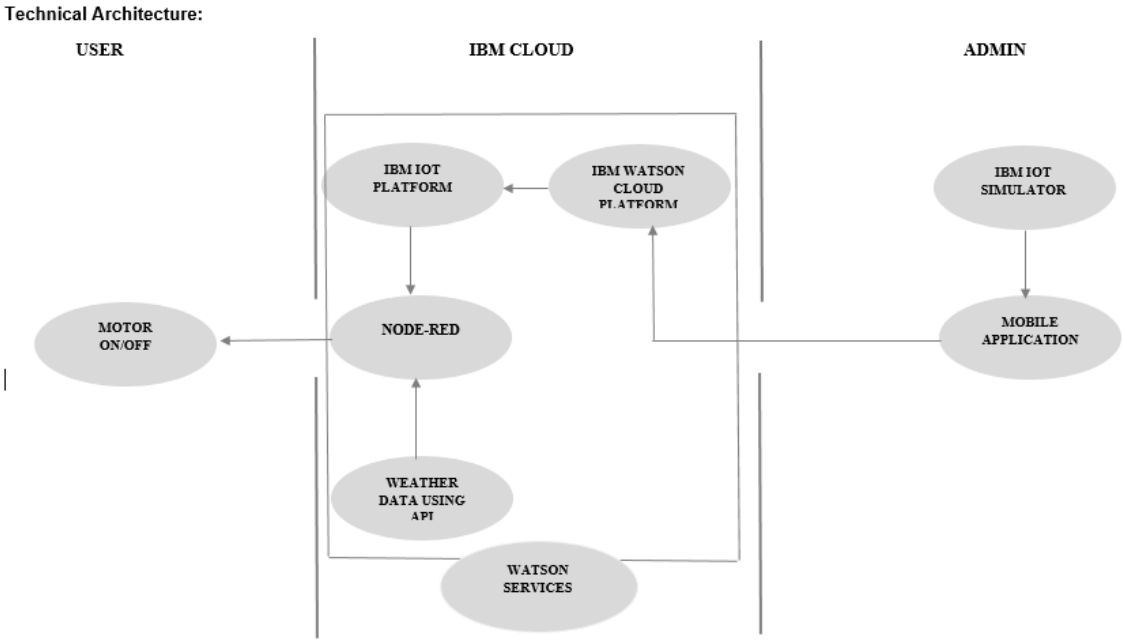
- ✓ Scalability and Multiplicity
- ✓ Security
- ✓ Privacy
- ✓ Interoperability
- ✓ Timelines or Real time
- ✓ Availability
- ✓ Ease of deployment, maintenance and use
- ✓ Spontaneous interaction,, Adaptability and flexibility

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE



6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

User Type	Functional Requirement (Epic)	Milestone	Activity List	Activity Acceptance criteria	Activity Priority
Customer (Mobile user)	Registration	Milestone -1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High
		Milestone -2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High
		Milestone -3	As a user, I can register for the application through Mobile number	I can register & access the dashboard with mobile number Login	Low
		Milestone -4	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium
	Login	Milestone -5	As a user, I can log into the application by entering email & password		High

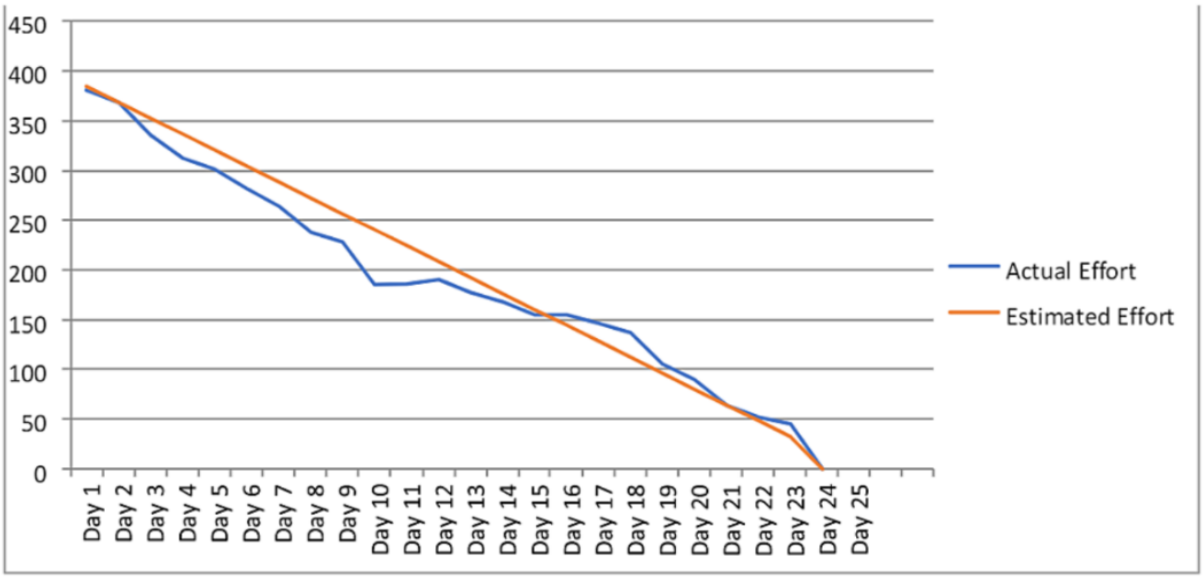
User Type	Functional Requirement (Epic)	Milestone	Activity List	Activity Acceptance criteria	Activity Priority
	Dashboard	Milestone -6	As a user , I can know the parameters values through dashboard	, I can know the parameters values through dashboard	High
	Access Motor Control	Milestone -7	As a user , I can manipulate motor from anywhere	I can manipulate motor from anywhere	High
Customer (Web user)	Registration	Milestone -8	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High
		Milestone -9	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High
		Milestone -10	As a user, I can register for the application through Mobile number	I can register & access the dashboard with mobile number Login	Low
		Milestone -11	As a user, I can register for the application through Gmail	I can register & access the dashboard with Gmail	Medium

User Type	Functional Requirement (Epic)	Milestone	Activity List	Activity Acceptance criteria	Activity Priority
	Login	Milestone -12	As a user, I can log into the application by entering email & password		High
	Dashboard	Milestone -13	As a user , I can know the parameters values through dashboard	, I can know the parameters values through dashboard	High
	Access Motor Control	Milestone -14	As a user , I can manipulate motor from anywhere	I can manipulate motor from anywhere	High

6.2 SPRINT DELIVERY SCHEDULE

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	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 REPORTS FROM JIRA



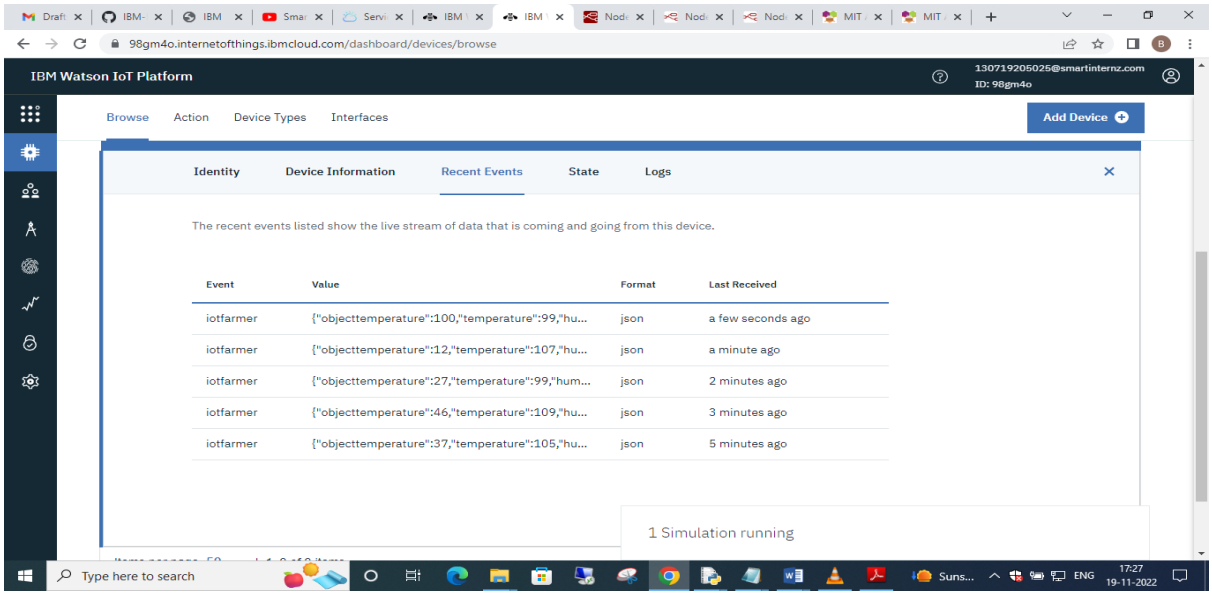
7. CODING & SOLUTIONING (Explain the features added in the project along with code)

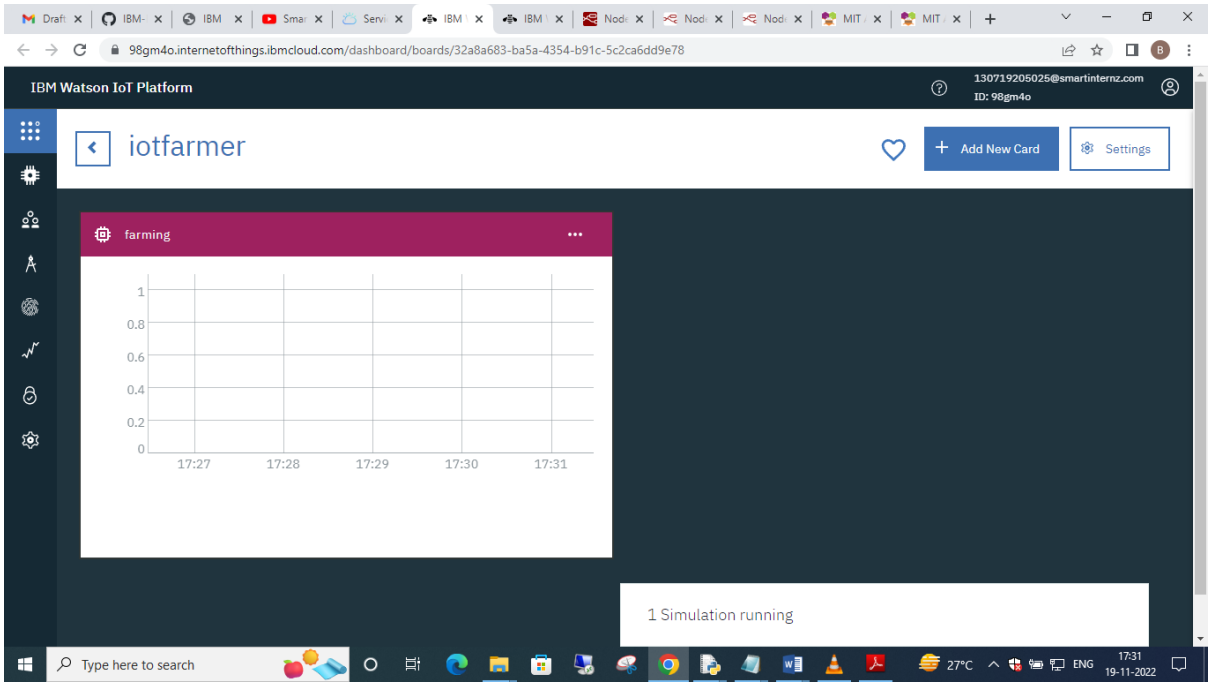
7.1 CODE

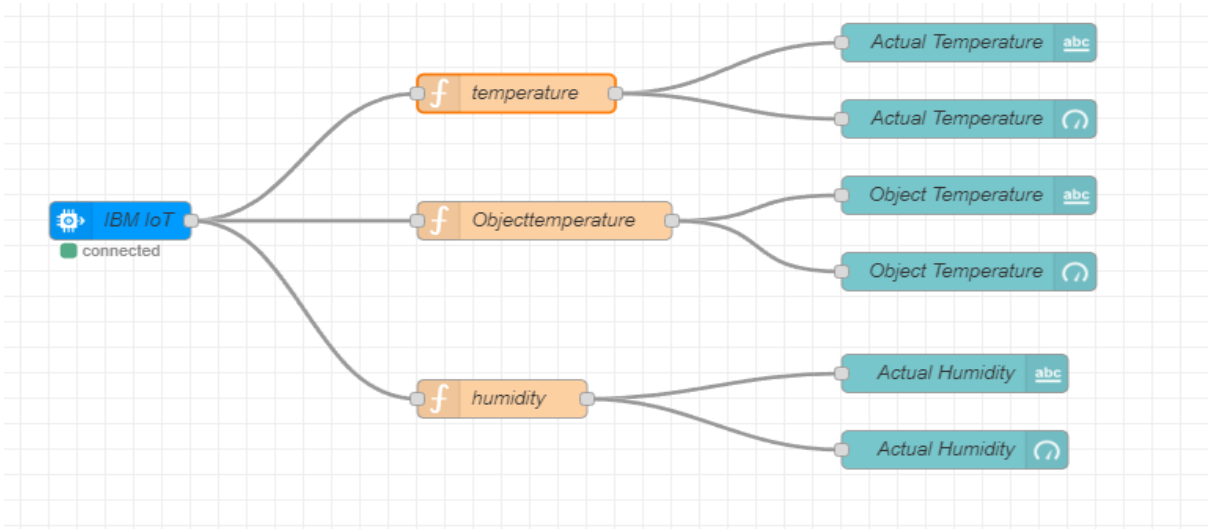
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
organization = "98gm4o"
deviceType = "iotdevice"
deviceId = "qwerty321"
authMethod = "token"
authToken = "qwerty123"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='ON':
        print("MOTOR ON IS RECEIVED")
        time.sleep(1)
        print("MOTOR STARTED")
    elif cmd.data['command']=='OFF':
        print("MOTOR OFF IS RECEIVED")
```

```
        time.sleep(1)
        print("MOTOR STOPPED")
    elif cmd.data['command']=='runfor30minutes':
        print("MOTOR RUNS FOR 30 MINUTES")
        print("MOTOR STARTED")
    for i in range(1,31):
        print("%d minutes to stop"%(30-i)) # use time.sleep(60) for delay of one minute
        time.sleep(2)
    print("MOTOR STOPPED")
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
deviceCli.disconnect()
```

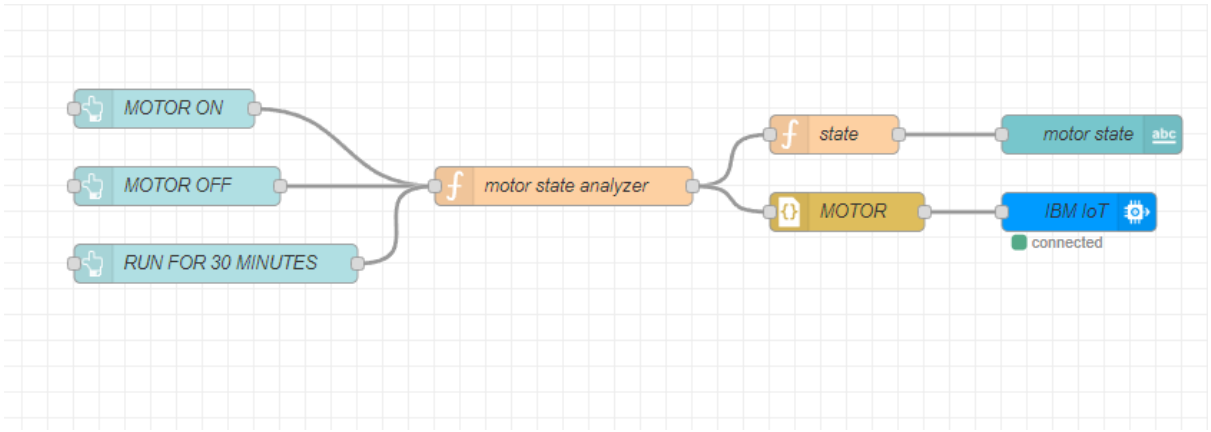
8. TESTING





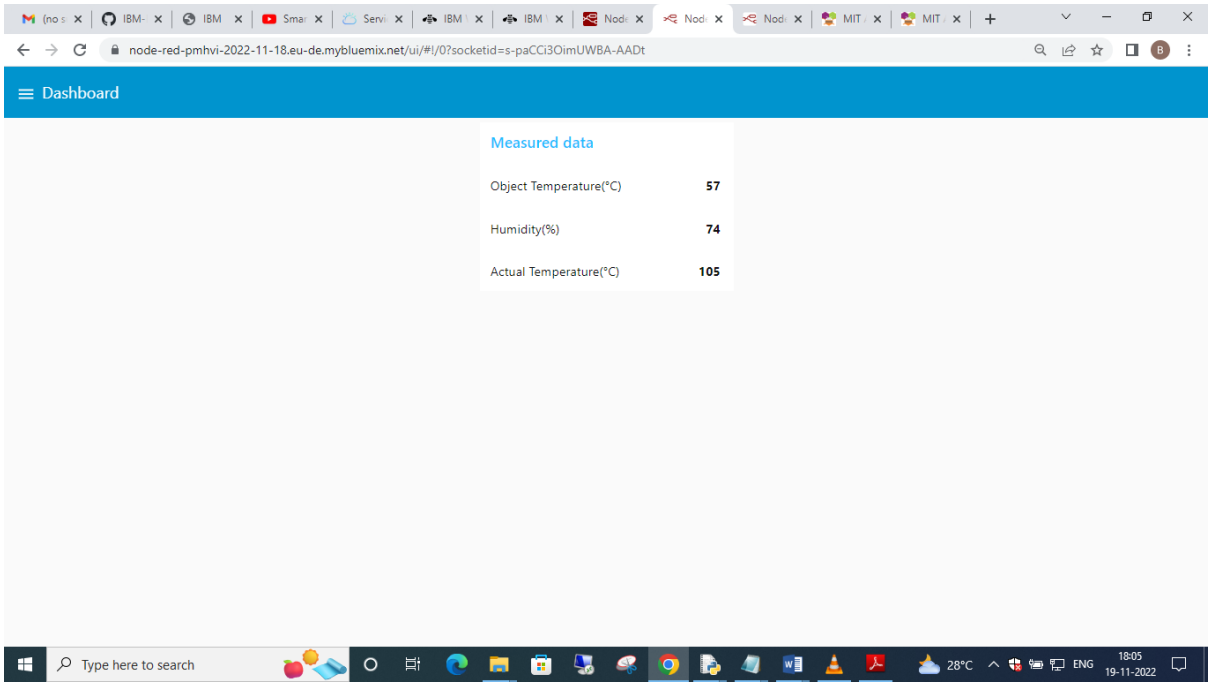


Nodes connected in following manner to get each reading separately

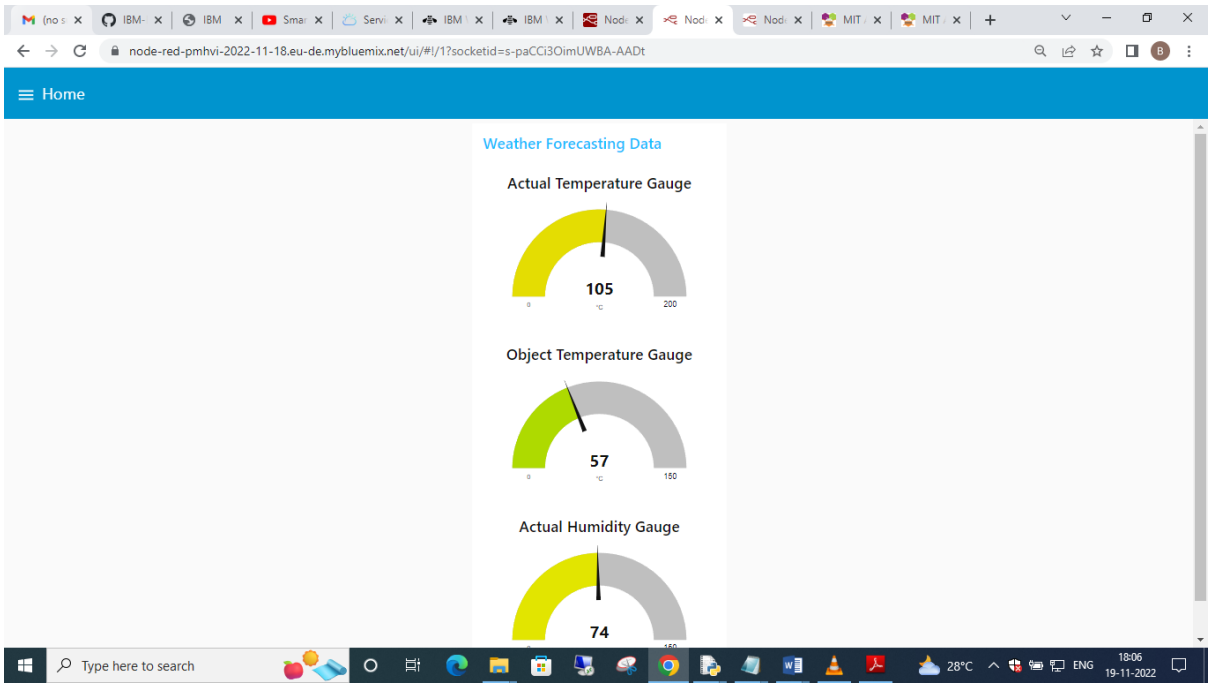


This program flow for sending commands to cloud

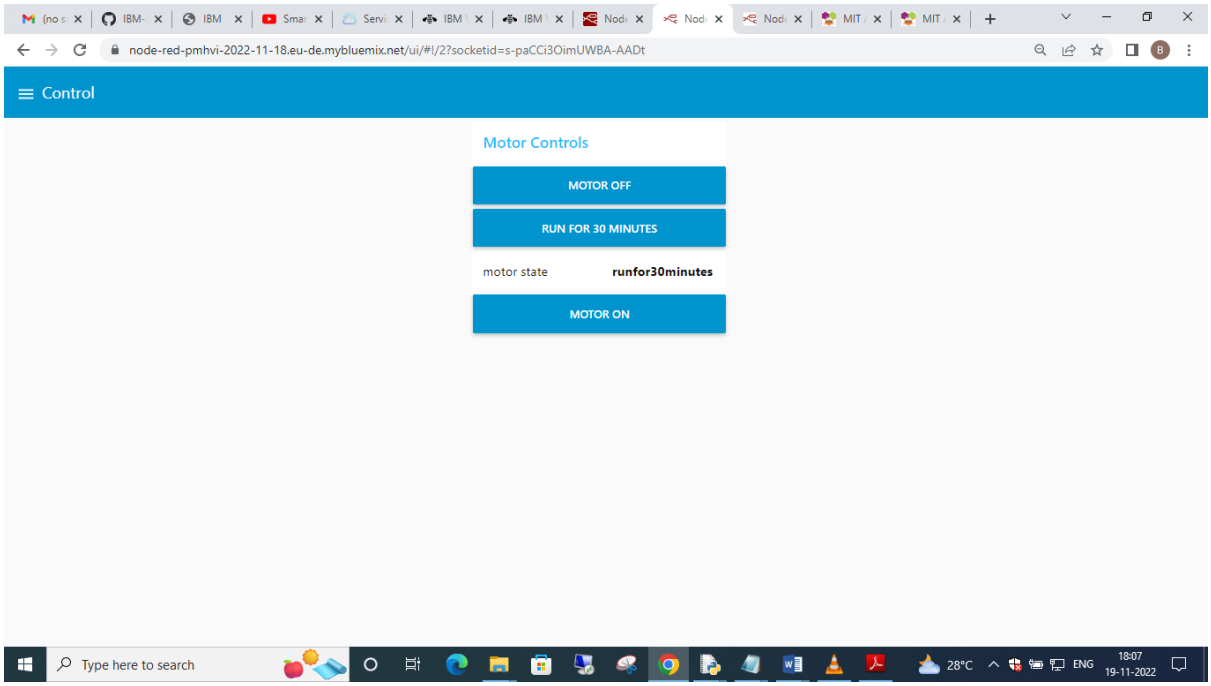
9. RESULTS



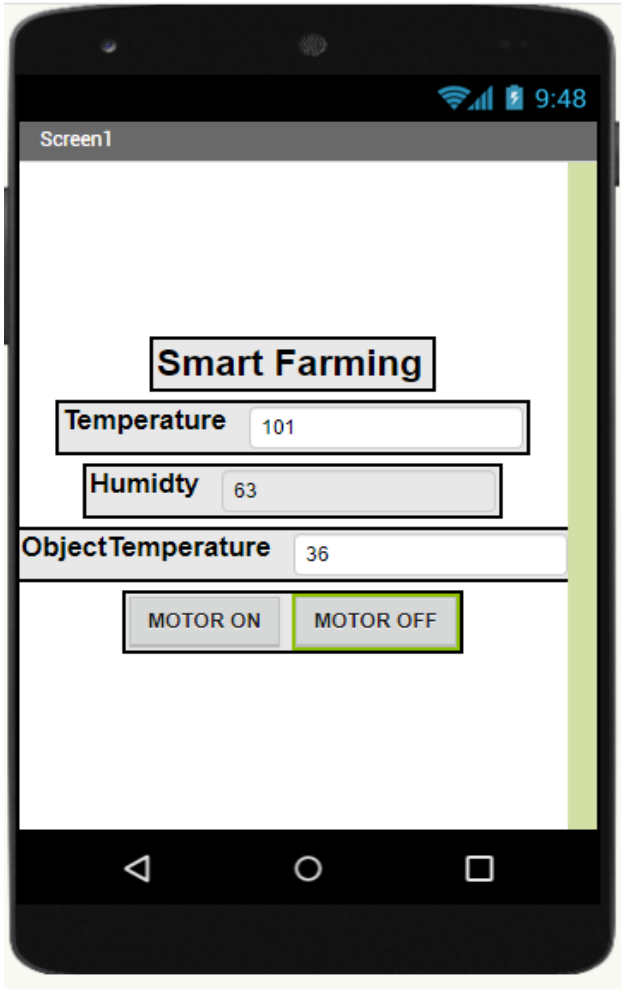
This page will show the measured value of the simulator



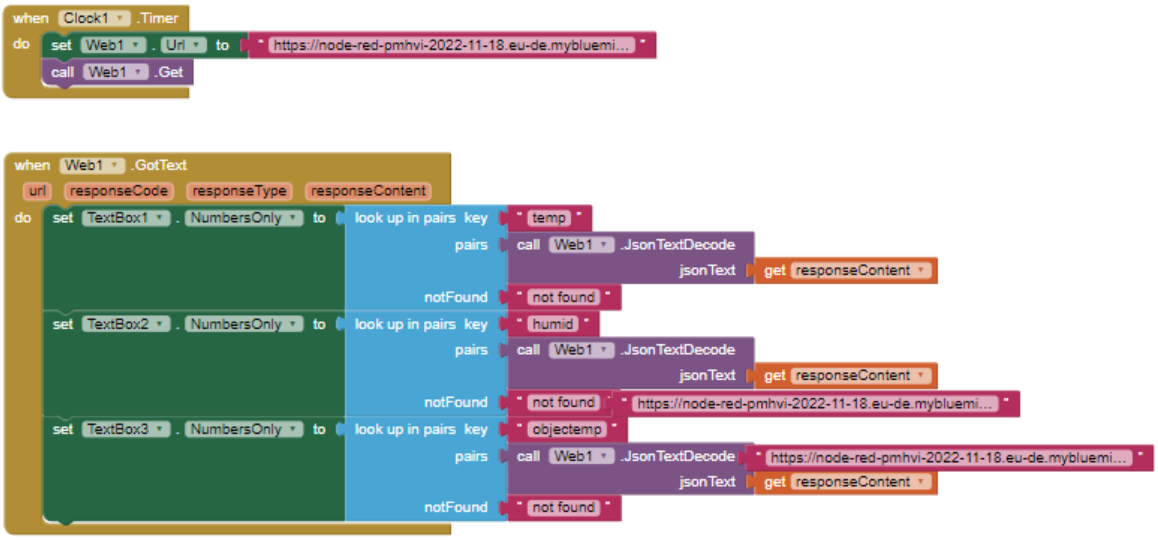
This page is for general weather forecasting data



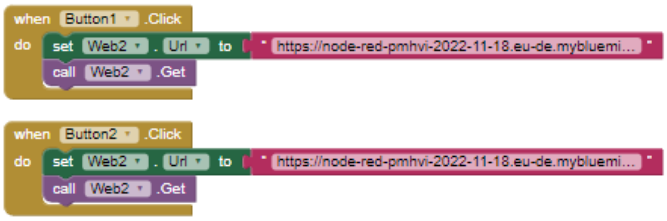
This page is for controlling the motor



This is the complete layout of the mobile app



This is the functions runs behind the textbox



This is the functions runs behind the button

The Actual Output of the smart farming in python console

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

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

10.2 DISADVANTAGES

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

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

Smart farming is certainly a leading enabler in producing more food with less for an increasing world population. In particular, smart farming enables increased yield through more efficient use of natural resources and inputs, and improved land and environmental management.

13. APPENDIX

SOURCE CODE

```
import time

import sys

import ibmiotf.application

import ibmiotf.device

organization = "98gm4o"

deviceType = "iotdevice"

deviceId = "qwerty321"
```

```
authMethod = "token"
authToken = "qwerty123"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    if cmd.data['command']=='ON':
        print("MOTOR ON IS RECEIVED")
        time.sleep(1)
        print("MOTOR STARTED")
    elif cmd.data['command']=='OFF':
        print("MOTOR OFF IS RECEIVED")
        time.sleep(1)
        print("MOTOR STOPPED")
    elif cmd.data['command']=='runfor30minutes':
        print("MOTOR RUNS FOR 30 MINUTES")
        print("MOTOR STARTED")
    for i in range(1,31):
        print("%d minutes to stop"%(30-i)) # use time.sleep(60) for delay of one minute
        time.sleep(2)
    print("MOTOR STOPPED")
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
deviceCli.disconnect()
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-15419-1659598497

DEMO LINK:

https://drive.google.com/drive/folders/1uyaUM-OPfj_7PVaMP1pZARMBkfMwj8Pe