KONGUNADU COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HX8001-PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTEREPRENEURSHIP

SMARTFARMER – IOT ENABLED SMART FARMING APPLICATION

NALAIYATHIRAN PROJECT REPORT 2022

SUBMITTED BY:

MENAGHA P	**621319106055**
NISHA P	**621319106062**
RAMYA P	**621319106074**
RUKMANI R	**621319106313**

TEAM ID: PNT2022TMID13531

SI.NO.	TITLE	PG.NO.

CHAPTER 1

INTRODUCTION

Agriculture is the root to country's economic development. In recent times, huge scientific advancement has been implemented in various agricultural fields for the betterment of the future. Despite of various researches, proper assessment and productivity couldn't be reached. The Agriculture Parameters are utilizing an IOT Technology and system availability that draw in these objects to assemble and deal information. The Internet of Things (IOT) enables things to be selected, recognised, or potentially forced remotely across existing configuration, create open gateways for all the additional obvious merging of the vast earth into PC-based frameworks, and acknowledge improved capacity, precision, and financial interconnected preferred stance. In fact, when IOT is expanded with sensors and actuators, the advancement changes into an instance of the broader category of electronic physical structures, which also includes innovations like smart grids, beautiful residences, intelligent transportation, and smart urban communities.

1.1 PROJECT OVERVIEW

In this project we have developed a mobile application using which a farmer can monitor the temperature, humidity, pressure and soil moisture parameters along with weather forecasting details. Based on these details he can water the crops by controlling the motors through the app and the app gives an alert message if temperature or humidity goes beyond a threshold value.

1.2 PURPOSE

Agriculture has a significant impact on an economy's health. The cornerstone of our economic system requires that production methods and product quality be improved. Now available is the smart agriculture system. The capacity of smart agriculture to help farmers grow high-quality crops is facilitated by automated farming, data gathering from the field, and data analysis.

Through real-time field monitoring, IoT-based smart farming enhances the entire agricultural system. The Internet of Things in Agriculture has reduced wasteful resource consumption, including the excessive use of water and power, and has saved farmers' time through the use of sensors and connection.

CHAPTER 2

LITERATURE SURVEY:

AUTHOR: Zuraida Muhammad, Muhammad Azri Asyraf Mohd Hafez, Nor Adni MatLeh, Zakiah Mohd Yusoff , Shabinar Abd Hamid

The term "Internet of Things" describes the process of attaching equipment, and other items to a network in order to share data (IoT). The Internet of Things (IoT) is being used more often to link things and gather data. Therefore, the usage of the Internet of Things in agriculture is essential. The project's goal is to build a smart agriculture network that is integrated with the internet of things. To deal with Malaysia's changing weather, technology is coupled with an irrigation system. The Raspberry Pi 4 Model B serves as the system's microcontroller. The DHT22 and soil moisture sensor are used to keep track of the local climate's temperature and humidity as well as the soil's level of moisture. The information will be accessible on a computer and a smartphone. Therefore, smart agriculture systems based on Raspberry Pi and the Internet of Things (IoT) have a big impact on how farmers work. Additionally, it will have a positive effect on agricultural productivity. Comparing IoT-based irrigation systems to traditional irrigation systems in Malaysia results in annual savings of about 24.44 percent. This will avoid water waste from daily demands while also saving money on labour costs.

AUTHOR: Divya J, Divya M, Janani V

India's economy and population both depend heavily on agriculture. This project aims to develop an embedded-based soil monitoring and irrigation system that will decrease manual field monitoring and deliver data via a mobile app. Using this technique, farmers should be able to produce more food. The instruments used to investigate the soil include pH sensors, temperature sensors, and humidity sensors. Farmers could choose to sow the most suitable crop for the area based on the findings. Through Wi-Fi, the field manager receives the sensor data, and the mobile app creates the crop recommendations. An automatic irrigation system is employed when the soil temperature is high.

AUTHOR: H.G.C.R. Laksiri, H.A.C. Dharmagunawardhana, J.V. Wijayakulasooriya

Farmers in the agricultural industry have a critical need for the creation of an efficient loT-based smart irrigation system. This study creates a low-cost, weather-based smart irrigation system. In order to get started, a system that can automatically adjust water flow to plants based on soil moisture levels must be developed for drip irrigation. Then, to increase the effectiveness of this water-saving irrigation system even further, an IoT-based communication feature is included. This feature enables a remote user to monitor the soil moisture levels and manually control the water flow. Additionally, the system has updated temperature, humidity, and rain drop sensors that enable online remote monitoring of these

variables. These field meteorological parameters are continuously kept in a distant database. Finally, a weather prediction algorithm is used to govern water distribution based on the current weather conditions. The suggested smart irrigation system would enable farmers to 88888O'irrigate their crops more effectively.

AUTHOR: Dweepayan Mishra, Arzeena Khan, Rajeev Tiwari, Shuchi Upadhaye

Indians' primary source of income is agriculture, which also has a significant impact on the country's economy. Crop development is crucial for increased yield and better product delivery. Crop beds with the best conditions and the right amount of moisture can therefore significantly affect yield. Streams that run from one end to the other are typical traditional irrigation methods. The moisture content of the fields may change as a result of this delivery. A well-planned watering system can improve the way the water system is managed. This study suggests a terrain-specific programmable water system that can increase agricultural productivity and reduce human labour costs. A moisture sensor, an Arduino kit, and a Wi-Fi module make up the system. By integrating our test system with a cloud infrastructure, data is obtained. Next, cloud services analyse the data and take the appropriate measures.

AUTHOR: Shweta B. Saraf, Dhanashri H. Gawali

A significant number of gadgets are connected to the internet via the "Internet of Things" (IoT) (IoT). Each item is identified uniquely, enabling data transmission without human intervention. It enables the creation of plans for better management of natural resources. The Internet of Things (IoT) idea states that smart devices with sensors enable communication with both the physical and logical worlds. The Internet of Things is the foundation for the system that this study proposes, and real-time input data is used. A smart farm irrigation system uses an Android phone to remotely monitor and control drips over a wireless sensor network. Zigbee is used to communicate between sensor nodes and base stations. The real-time observed data from the server is processed and displayed using a web-based java graphical user interface. Using an Android phone for remote monitoring and control, field irrigation system wireless monitoring eliminates the need for human intervention.

2.1 EXISTING PROBLEM

The climate has a big impact on agriculture. Depending on the environment, rising temperatures and carbon dioxide can improve some crops' yields; however, other factors like humidity, pressure, and water accessibility must also present. Even though moderate warming and an increase in atmospheric carbon dioxide would help some plants grow more quickly, extreme temperatures, floods, and drought would decrease crop production. These require a lot of time from farmers to maintain.

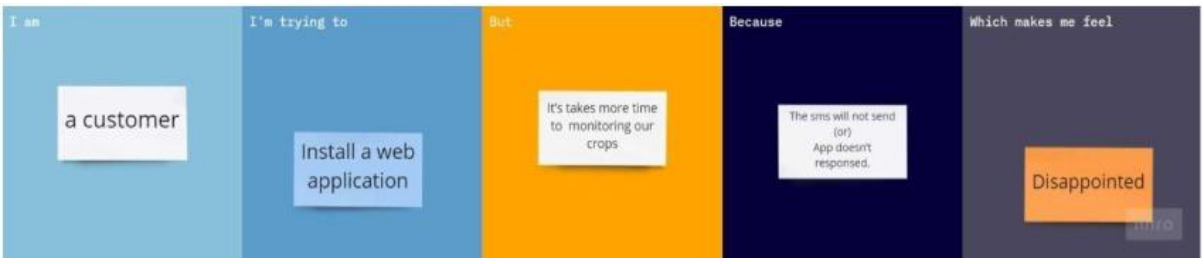
Extreme weather does not only include heat. Farmers can profit from extreme cold by freezing the deep-seated soil. Frost depths can reach more than 40 inches in some areas of the upper Midwest. Farmers can benefit in a variety of ways from a deep frost depth. The cold prevents nitrogen that is applied in the fall from vaporising during the winter. The repeated freezing and thawing of water aids in the soil's eventual softening. Some insects have a worse chance of surviving in extremely cold temperatures and frozen soils.

Severe weather other than heat and cold can cause loss and devastation to a farm. Most farmers can't avoid the results of extreme weather. Diverse extreme weather can affect farms in different ways. Because of this, it's important that farmers have a proper system and need a mobile application to monitor the weather changes and to control the motor.

2.2 REFERENCES

- Zuraida Muhammad,Muhammad Azri Asyraf Mohd Hafez,Nor Adni Mat "Smart Agriculture Using Internet of Things with Raspberry Pi." 2020.
- Divya J., Divya M.,Janani V."IoT based Smart Soil Monitoring System for Agricultural Production" 2017.
- H.G.C.R.Laksiri, H.A.C.Dharmagunawardhana, J.V.Wijayakulasooriya "Design and Optimization of loT Based Smart Irrigation System in Sri Lanka"2019.
- Anushree Math, Layak Ali, Pruthviraj U "Development of Smart Drip Irrigation System Using IoT"2018.
- Shrihari M, "A Smart Wireless System to Automate Production of Crops and Stop Intrusion Using Deep Learning" 2020.
- G. Sushanth1, and S. Sujatha, "IOT Based Smart Agriculture System"2018.
- Dweepayan Mishra1, Arzeena Khan2 Rajeev Tiwari3 , Shuchi Upadhay, "Automated Irrigation System-IoT Based Approach",2018.

2.3 PROBLEM STATEMENT DEFINITION



Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Need crop quality.	Water level checking.	The water pipeline cause some issues.	The motor does not work it.	The customer becomes very angry .
PS-2	Monitoring.	Analyze the temperature.	It leads to high level temperature.	It is a summer season.	They are disappointed.

Fig.2.3.1

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

A collaborative tool called an empathy map can help teams understand their clients more thoroughly. An empathy map can serve as a representation for a group of users, such as a consumer segment, much like a user persona. The empathy map was invented by Dave Gray in the beginning and has become quite well-liked in the agile community. You can utilise empathy maps if you feel the urge to fully immerse yourself in a user's environment.

Everyone would add at least one sticky to every section. You might ask questions, such as:

- What would the user be thinking and/or feeling? What are some of their worries and aspirations?
- What would their friends, colleagues, and boss be likely to say while the user is using our product? What would the user hear in these scenarios?
- What would the user see while using our product in their environment?
- What might the user be saying and/or doing while using our product? How would that change in a public or private setting?
- What are some of the user's pain points or fears when using our product?
- What gains might the user experience when using our product?

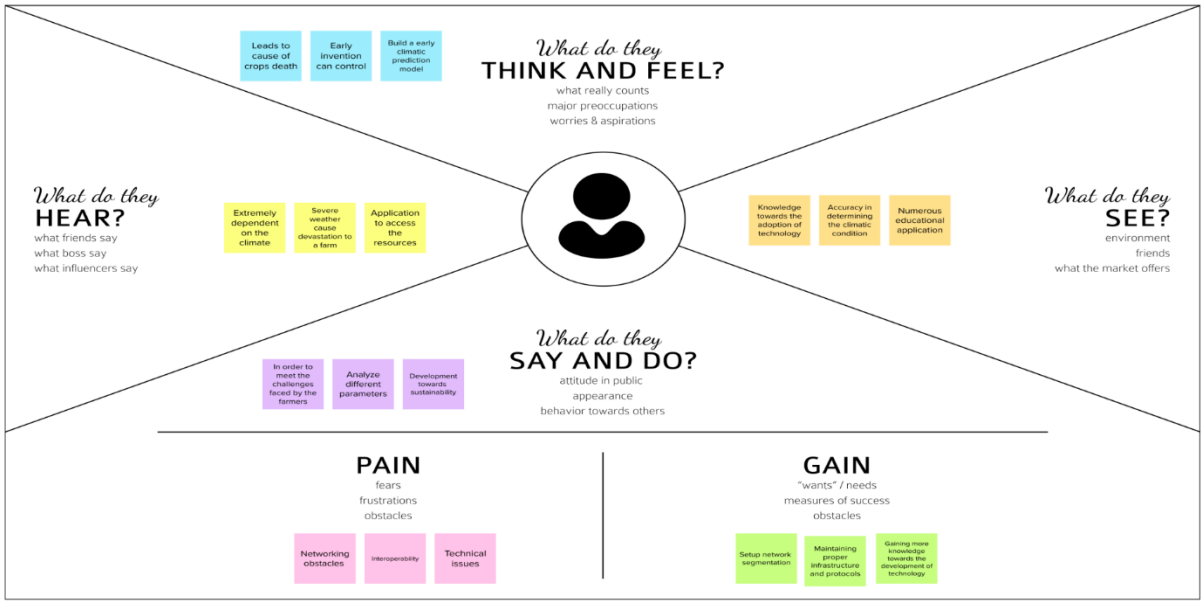


Fig. 3.1.1

3.2 IDEATION & BRAINSTORMING

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. The main distinction between ideation and brainstorming is that whereas brainstorming is nearly often done in groups, ideation is typically seen as being more of a solitary endeavour. A group of people are frequently gathered for a brainstorming session to generate either fresh, general ideas or solutions to specific problems or circumstances.

For example, a major corporation that recently learned it is the object of a major law suit may want to gather together top executives for a brainstorming session on how to publicly respond to the lawsuit being filed.

Participants in a brainstorming session are encouraged to freely toss out whatever ideas may occur to them. According to the theory, by coming up with a lot of ideas, the brainstorming group is more likely to find a workable solution to the problem they are trying to solve.

With the creation of various brainstorming software tools, including Idea wake and Bright Idea, the distinction between ideation and brainstorming has gotten a little bit more hazy. These software tools are created to inspire staff members to come up with fresh suggestions for enhancing business operations and, eventually, bottom-line profitability.

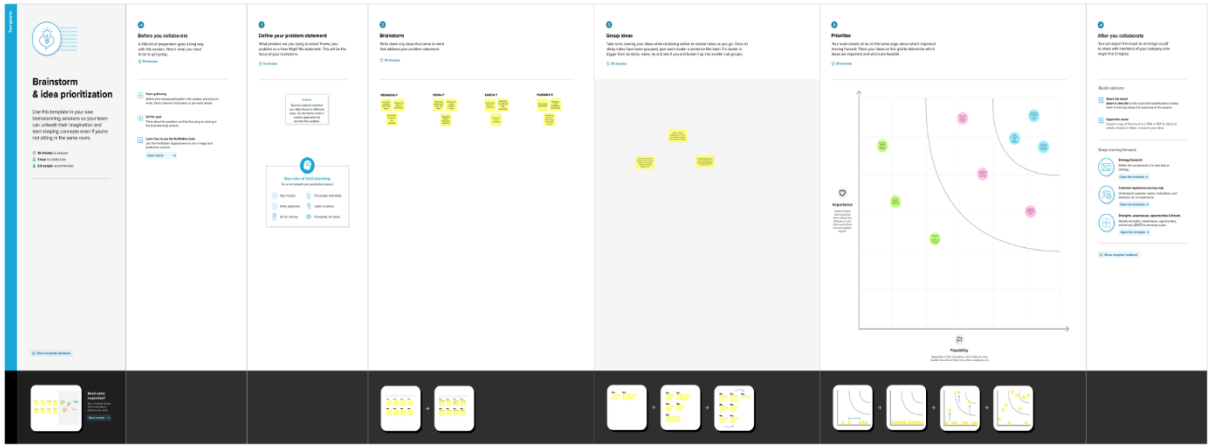


Fig. 3.2.1

3.3 PROPOSED SOLUTION

As the climates are changing rapidly and weather is unpredictable, so farmers are facing difficulties so they need a system to tackle this, here we use “open weather API” to get weather information such as temperature, pressure, humidity and weather description at their current location.

Based on which they can decide whether to turn on the motors or turn off the motor if needed temperature and moisture sensors from IBM simulator is displayed on UI for

monitoring the weather. An algorithm developed with threshold values of temperature, pressure, humidity is programmed to intimate the farmer if weather conditions go bad. He can control motors remotely from any place through IoT. Internet interface that allow data inspection and irrigation scheduling to be programmed through mobile application or Node-RED UI. The technological development in software and hardware make it easy to develop this which can make better monitoring and wireless network made it possible to use in monitoring and control of greenhouse parameter in precision agriculture.

3.4 PROBLEM SOLUTION FIT

The Problem-Solution is a tool for entrepreneurs, marketers, and corporate innovators that helps to find ideas with higher odds of solution adoption, minimise time spent on solution testing, and gain a better understanding of the existing situation. Such information is generally acquired "on the fly," following rounds of revisions and consumer interviews, but it is critical to your success. This canvas contains everything you need to find patterns and realise what would work and why, based on the ideas of learn startup, and user experience design. Simply be where your consumers are and address a genuine need, whether it's the same problem done differently or something new presented in a familiar way.

In this project these are the needs for that.

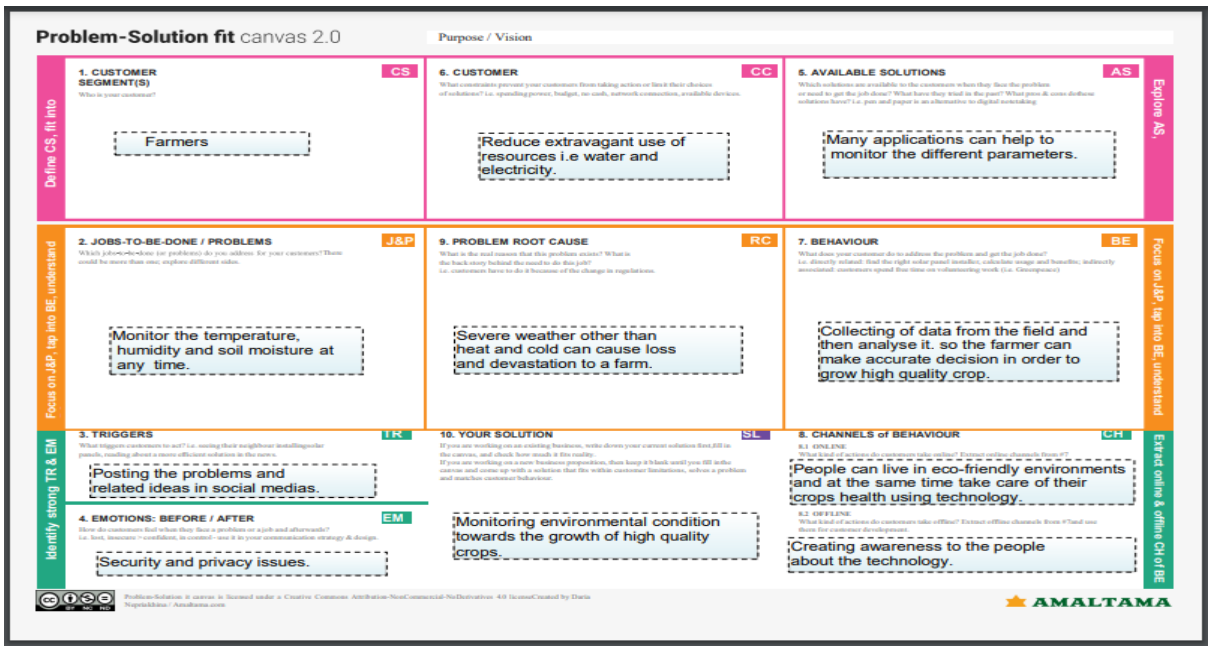


Fig. 3.4.1

CHAPTER 4

REQUIREMENT ANALYSIS

Determining user expectations for a new or modified product is the process known as requirements analysis, sometimes known as requirements engineering. These characteristics, also known as criteria, must be precise, pertinent, and quantitative. Such specifications are sometimes referred to as functional requirements in software engineering.

Project management includes requirements analysis as a key component. In order to resolve disagreement or ambiguity in requirements as needed by different users or groups of users, eliminate feature creep, and document every step of the project development process from beginning to end, requirements analysis requires continuous communication with system users. Instead of attempting to shape user expectations to match the requirements, effort should be focused on ensuring that the end system or product adheres to client needs.

Teamwork is essential for requirements analysis, which calls for knowledge of hardware, software, and human factors engineering as well as interpersonal skills.

The Requirements Analysis Phase's goal is to turn the needs and high-level requirements defined in prior phases into requirements that are clear, complete, consistent, traceable, and approved by all relevant stakeholders.

4.1 FUNCTIONAL REQUIREMENTS

Functional requirements specify what a system is expected to do and can include computations, technical details, data processing and manipulation, among other functions. All of the situations in which the system employs the functional requirements are described in use cases, which are known as behavioural requirements.

Following are the functional requirements of the proposed solution.

S.NO.	FUNCTIONAL REQUIREMENT DES
1.	Farmer must be able to receive the weather forecast every hour.
2.	The mobile app must be user friendly to the farmer.
3.	Farmer must be able to monitor the temperature, humidity and soil moisture parameters along with weather forecasting details.
4.	Based on the forecast parameters he must be able to control the motor if needed.

4.2 NON-FUNCTIONAL REQUIREMENTS

A Non-functional requirement (NFR) is a requirement in systems engineering and requirements engineering that describes criteria rather than specific behaviours that can be used to evaluate how a system performs. They stand in contrast to functional requirements, which specify certain behaviours or activities. The system design includes a thorough implementation strategy for functional needs.

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

S.NO.	NON-FUNCTIONAL REQUIREMENT	DESCRIPTION
1.	Usability	The mobile application can monitor the temperature, humidity, pressure and soil moisture parameters along with weather forecasting details. Based on these details he can water the crops by controlling the motors through the app and the app gives an alert message if temperature or humidity goes beyond a threshold value.
2.	Security	The system needs the patient to recognize herself or himself using the phone. Any users who make use of the system.
3.	Reliability	It specifies how likely the system or its element would run without a failure.
4.	Performance	The system provides acknowledgment in just one second. The user interface acknowledges within five seconds.
5.	Availability	The system is available all the time.
6.	Scalability	Threshold values are set any anomalies will be reported to the farmer. User friendly and efficient. Low cost.

CHAPTER 5

PROJECT DESIGN

The classic visual representation of how information moves through a system is a data flow diagram (DFD). A tidy and understandable DFD can graphically represent the appropriate quantity of the system demand. It can be done manually, automatically, or both. It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

A DFD's goal is to outline the boundaries and scope of a system as a whole. It can be utilised as a communication tool between a system analyst and any participant in the sequence that serves as the foundation for system redesign. The DFD is also known as a bubble chart or data flow graph.

A location for the collecting of data items is indicated by a series of parallel lines. A data store denotes the storage of data that can be used later or by various processes in a different order. The data storage may contain one or more components. Any level of abstraction for a system or piece of software can be performed using the DFD. Levels that correspond to increasing information flow and functional detail may be partitioned into DFDs. The system is then broken down and represented as a DFD with several bubbles. The system components that each of these bubbles represents are then broken down and documented as ever-more-detailed DFDs.

5.1 DATA FLOW DIAGRAMS

The classic visual representation of how information moves through a system is a data flow diagram (DFD). A tidy and understandable DFD can graphically represent the appropriate quantity of the system demand. It can be done manually, automatically, or both.

It demonstrates how information enters and exits the system, what modifies the data, and where information is kept.

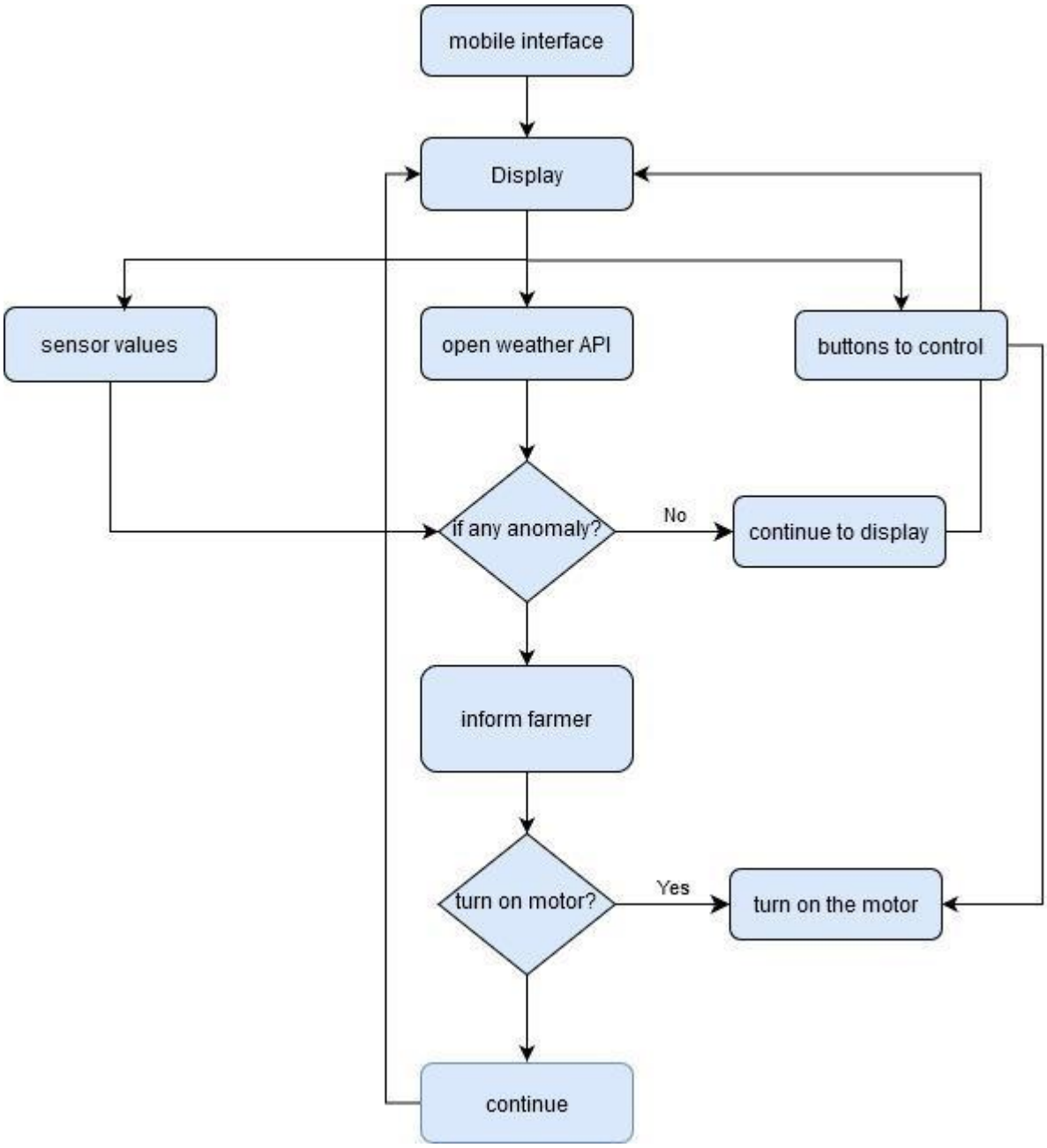


Fig. 5.1.1

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Solution architects are most like project managers in that they make sure that all parties, including stakeholders, are on the same page and going in the right direction at all times. A

new application's successful implementation is overseen by technical architects. They suggest a set of components that together offer the optimum solution. The technological architecture and enterprise architecture are connected by this method, which is particularly detail-oriented. It also requires a depth of understanding of the technological and business operations of the corporation.

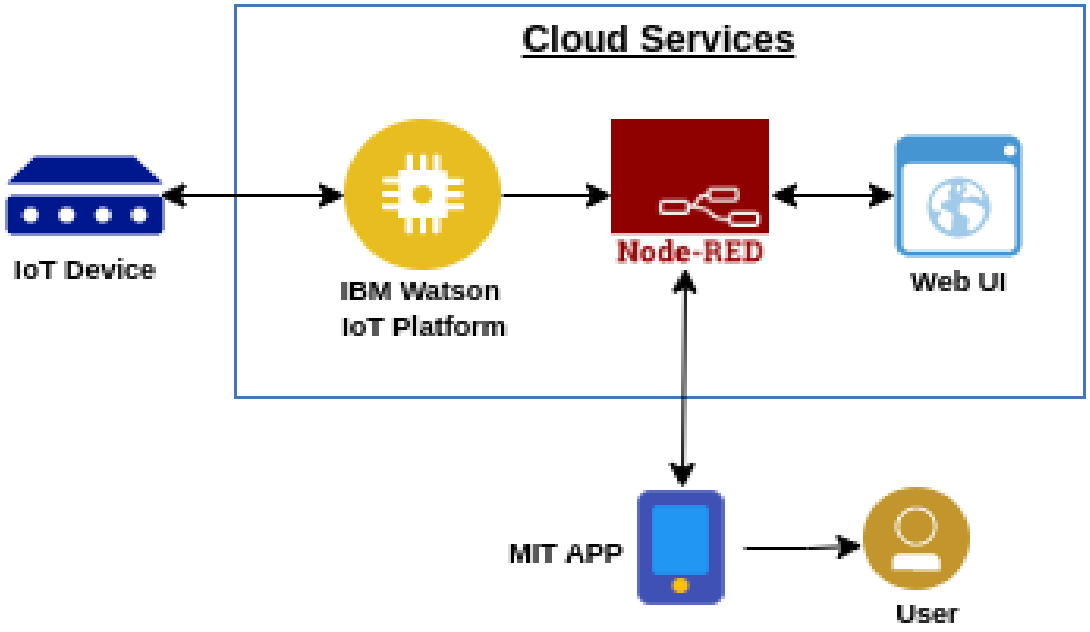


Fig. 5.2.1

5.3 USER STORIES

The smallest piece of work in an agile system is a user story. It is a final objective, not a feature, as seen through the eyes of a software user.

A user story is a casual, all-inclusive description of a software feature written from the viewpoint of the client or end user.

A user story's objective is to describe how a piece of work will provide the customer with a specific value. Keep in mind that "customers" don't always have to be end users on the outside in the conventional sense; they might also be colleagues or internal customers within your company who depend on your team.

User stories are short, straightforward statements that describe the desired result. They don't get specific. Later requirements are added.

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	I can access my account / dashboard	High	Sprint-1

			email, password, and confirming my password.			
		USN-2	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-3	As a user, I can register for the application through Gmail.	I can receive confirmation email & click confirm to login.	Medium	Sprint-1
	Login	USN-4	As a user, I can log into the application by entering email& password.		High	Sprint-1
		USN-5	If I forgot my password or username, I can reset it again through my email.	I can receive reset mail to the registered Email Id.	Medium	Sprint-2
Customer (Web user)	Registration	USN-6	As a user, I can register by entering my email, password, and confirming my password	I can access my account / dashboard.	High	Sprint-2
		USN-7	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm.	High	Sprint-2
		USN-8	As a user, I can register for the application through Gmail.	I can receive confirmation email & click confirm to login.	Medium	Sprint-3

	Login	USN-9	As a user, I can log into the application by entering email & password.		High	Sprint-4

CHAPTER 6

PROJECT PLANNING & SCHEDULING

Planning - Planning is the process of determining what supplies and resources will be needed to meet incoming and anticipated demand. This stage is essential to making sure you have the necessary supplies and resource capacity on hand to fulfil your orders on time. This element relates to the "what" and "how" of every project: precisely what must be performed and how it will be done.

Scheduling - The time of the utilisation of specific organisational resources is determined by scheduling. In manufacturing, scheduling entails creating schedules for personnel, machinery, and supplies. By allocating the proper resources to finish the production plan within a certain time frame, it addresses the "when" of a project. Your facility will be able to cut expenses, boost productivity, and deliver goods on time if you create efficient production schedules.

6.1 SPRINT PLANNING AND ESTIMATION

Planning: The team decides what it will develop and how it will build it during the sprint planning phase. After breaking user stories down into tasks and performing task-level estimation, the team commits to the Sprint target. The Product Owner, Scrum Master, and Team coordinate sprint planning. Each project in Scrum is divided into sprints, which are time chunks that are typically 2-4 weeks long. The Scrum Team, Scrum Product Manager, and Scrum Master gather for a sprint planning meeting to decide which backlog items will be tackled during the following sprint.

Estimation: During the Sprint Planning Meeting, the entire team estimates in Scrum projects. The goal of the estimation would be to prioritise the User Stories for the Sprint and assess the team's capacity to complete them inside the Sprint's Time Box.

The prioritised User Stories are moved to the top of the Product Backlog by the Product Owner, who also makes sure they are clear and can be estimated.

The Scrum Team will take care to choose the User Stories for the Sprint based on the size of the Product Increment and the effort necessary for the same, as the Scrum Team as a whole is accountable for the delivery of the product increment.

User Story Points are used to estimate the size of the product increment. Once the size has been established, the effort is approximated using historical data, or effort per User Story Point, otherwise known as Productivity.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	High	Menagha
Sprint-1		USN-1	As a user, I will receive confirmation email once I have registered for the application.	Medium	Menagha
Sprint-1		USN-1	As a user, I can register for the application through Gmail.	High	Menagha
Sprint-2	Login	USN-2	As a user, I can log into the application by entering email& password.	High	Nisha
Sprint-3	Dashboard	USN-2	As a user, I can access my dashboard through the url provided.		Ramya
Sprint-4	Scheduling appointments	USN-4	During this interaction, the farmer collects basic information about the field and the climatic condition. With this information, the farmer can cultivate the crop in the field.	High	Rukmani

6.2 SPRINT DELIVERY SCHEDULE

Since sprints take place over a fixed period of time, it's critical to avoid wasting time during planning and development. And this is precisely where sprint scheduling enters the equation.

In case you're unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

Teams often run into trouble when they create more than few schedules. This can create conflict and derail projects midway through their cycles. To ensure things stay on track, one schedule makes sense.

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

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

6.3 REPORTS FROM JIRA FILES

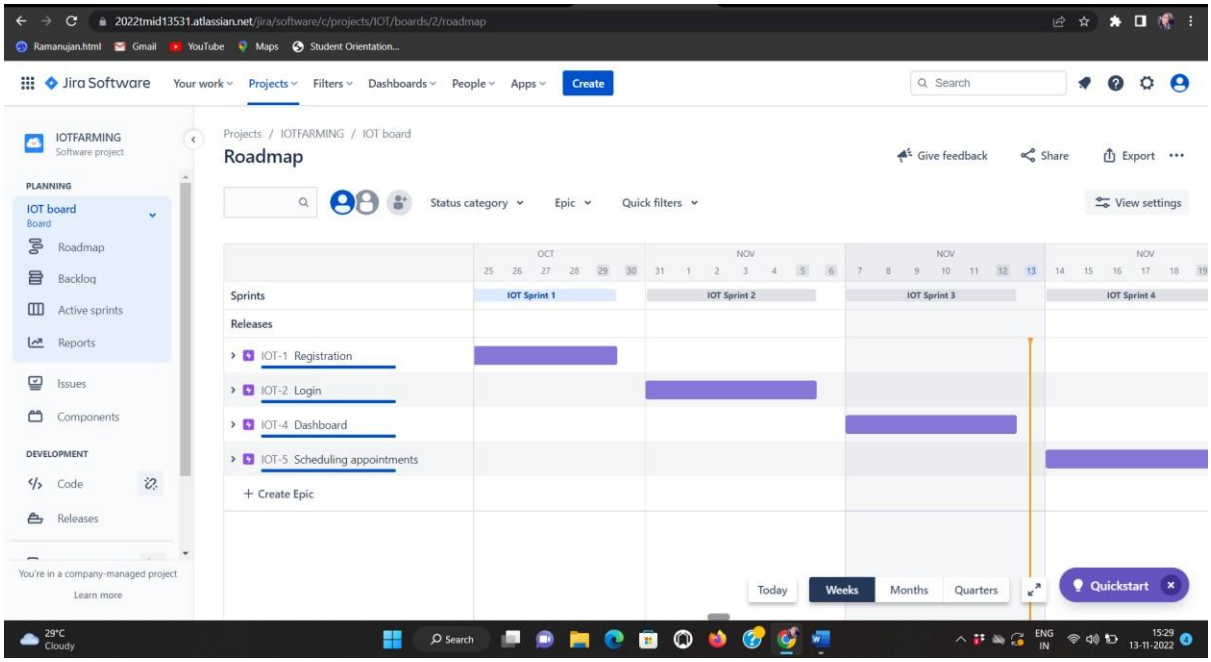


Fig. 6.3.1

CHAPTER 7

CODING AND SOLUTIONING

7.1 FEATURE 1

To present your insights and analysis, IBM cloud offers dashboards and stories. A view that includes graph, chart, plot, table, map, or any other type of visual representation of data can be put together.

CODE:

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
    "identity":{
        "orgId":"nqhzg5",
        "typeId":"Node-red",
        "deviceId":"1234"
        },
    "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(-20,125)
    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(2)
    client.commandCallback = myCommandCallback
client.disconnect()
```

7.2 FEATURE 2

IBM CLOUD SERVICES:

Created the IBM Watson IoT Platform and the device and create the node-red services.

PYTHON SCRIPT:

Develop the python code and connect the IBM Watson device with python code.

NODE-RED APPLICATION:

Build the node-red connection for the project and display the web application using the dashboard module.

MIT APP INVENTOR:

Develop the mobile application using the MIT App inventor and add the links from the node-red web application to operate the motor switches, according to the data obtained.

CHAPTER 8

TESTING

8.1 TEST CASES

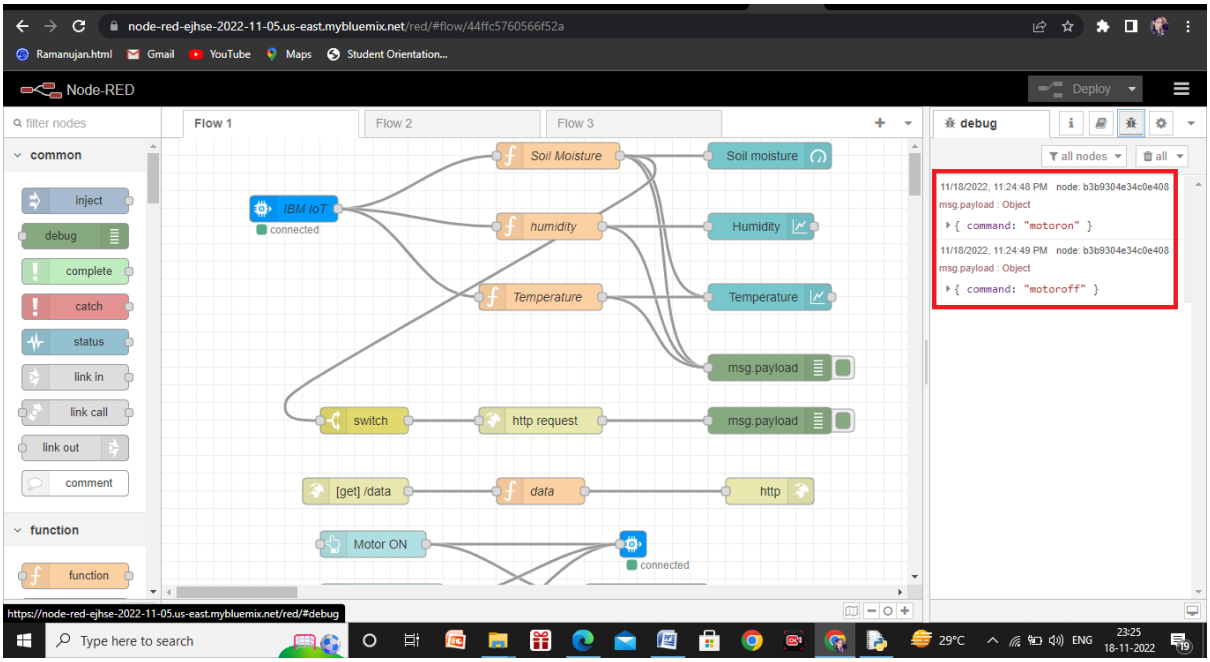
Checking is the goal of testing. Testing is the process of creating an attempt to find every feasible flaw or weakness in a particular work product. It explains how to picture pieces, sub-assemblies, assemblies, and/or a finished product in practise. It is a technique for physically testing software to ensure that it fulfils its requirements, satisfies user expectations, and doesn't malfunction in an unacceptable way.

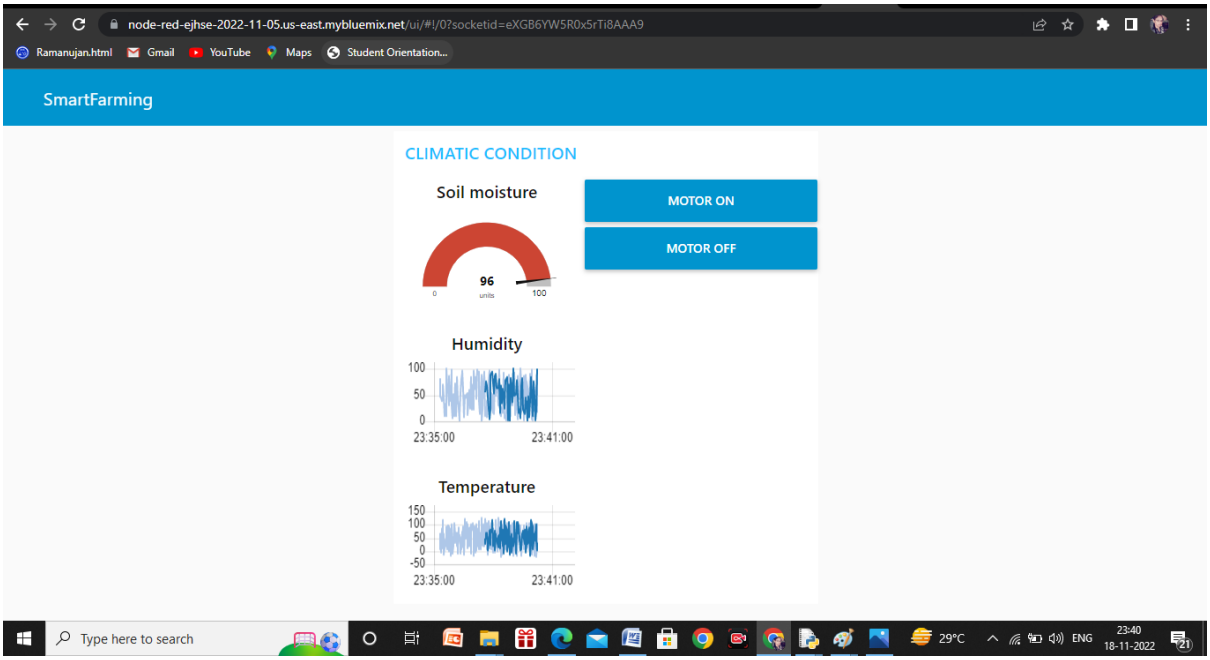
TEST CASE	SCENARIO	TEST STEPS	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT	COMMENT
IBM watson IoT platform	Check the Watson device get connected.	Login	ID, Password	Successfully login	Successfully login	Pass	Good
		Organization ID	Org ID	Display the ID	Display the ID	Pass	Good
		Create the device, generate the API key	Device name, type	Successfully created	Successfully created	Pass	Good
Python software	Write the code for the project and connect the device in Watson.	Check temperature, humidity, soil moisture	Temperature, humidity, soil moisture	Display the values.	Display the values.	Pass	Good
Node-red	Build the connection in node-red	Display the graph in web application	Graph, Motor ON, Motor OFF	Display the graph and Motor ON, Motor OFF	Display the graph	Pass	Good
MIT App inventor	Build the application	Display the output in mobile phone.	Temperature, Humidity, soil moisture value	Display the values.	Display the values.	Pass	Good

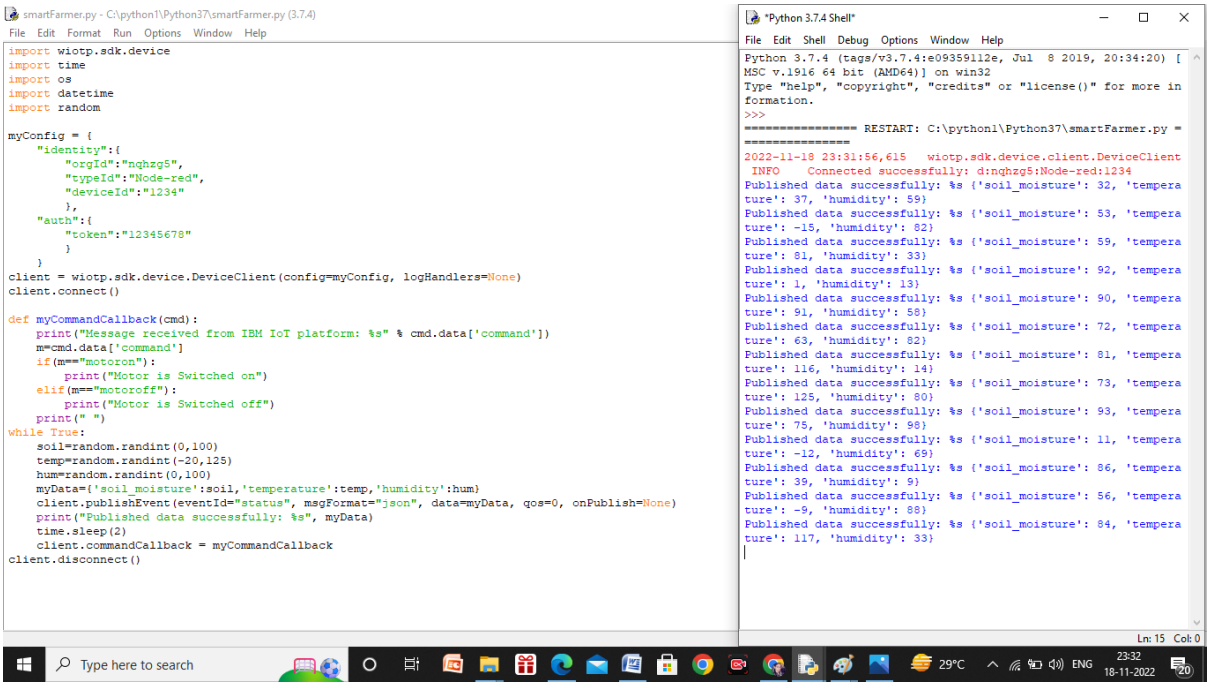
8.2 USER ACCEPTANCE TESTING

Acceptance by users any project testing phase may be crucial, and the tool used for user's participation is crucial. Additionally, it guarantees that the system satisfies real-world requirements. At this point, all the cases are executed to ensure that the programme is accurate and complete.

Before the customer will accept the programme, the test must be passed successfully. After customer personnel have verified that the preliminary production statistics load is accurate and that the test suite has been completed flawlessly, the customer formally accepts the delivery of this system.




node-red-ejhse-2022-11-05.us-east.mybluemix.net/ui/#!/0?socketid=eXGB6YW5R0x5rTi8AAA9
Ramanujan.html
Gmail
YouTube
Maps
Student Orientation...
SmartFarming
CLIMATIC CONDITION
Soil moisture
96
units
0
100
MOTOR ON
MOTOR OFF
Humidity
100
50
0
23:35:00
23:41:00
Temperature
150
100
50
0
-50
23:35:00
23:41:00
Type here to search
29°C
ENG
23:40
18-11-2022

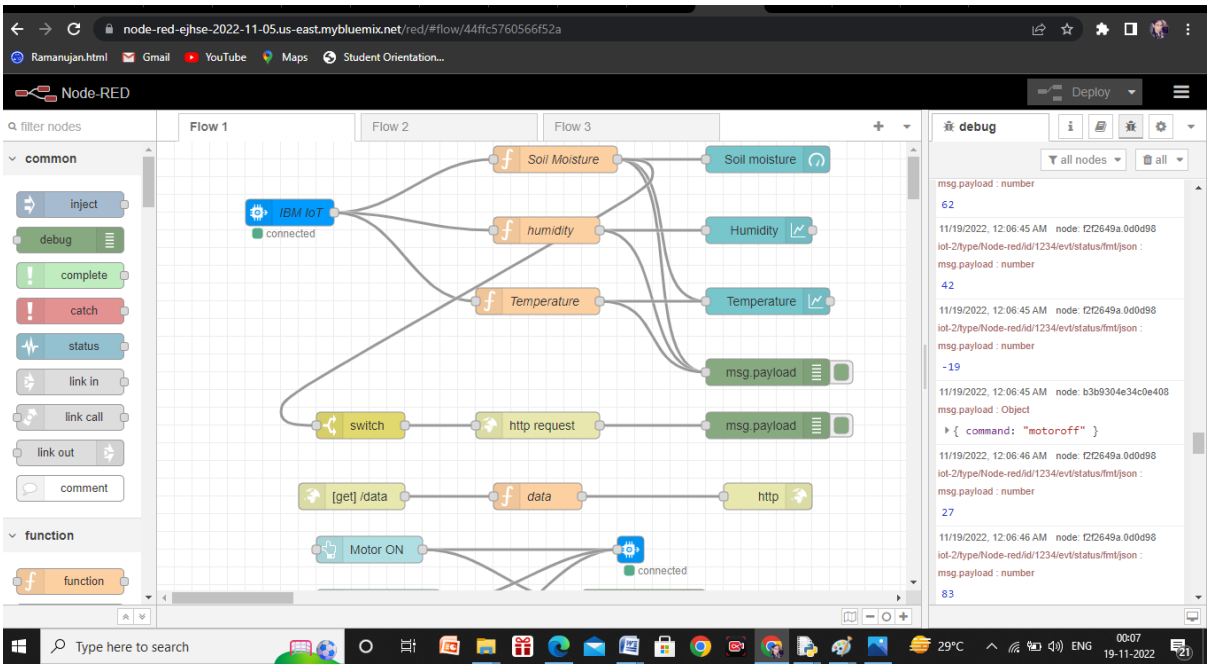

smartFarmer.py - C:\python1\Python37\smartFarmer.py (3.7.4)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
"identity":{
"orgId":"nqhzg5",
"typeId":"Node-red",
"deviceId":"1234"
},
"auth":{
"token":"12345678"
}
}
client = 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(-20,125)
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(2)
client.commandCallback = myCommandCallback
client.disconnect()
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [
MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>>
========== RESTART: C:\python1\Python37\smartFarmer.py =
==========
2022-11-18 23:31:56,615 wiotp.sdk.device.client.DeviceClient
INFO Connected successfully: d:nqhzg5:Node-red:1234
Published data successfully: %s {'soil_moisture': 32, 'tempera
ture': 37, 'humidity': 59}
Published data successfully: %s {'soil_moisture': 53, 'tempera
ture': -15, 'humidity': 82}
Published data successfully: %s {'soil_moisture': 59, 'tempera
ture': 81, 'humidity': 33}
Published data successfully: %s {'soil_moisture': 92, 'tempera
ture': 1, 'humidity': 13}
Published data successfully: %s {'soil_moisture': 90, 'tempera
ture': 91, 'humidity': 58}
Published data successfully: %s {'soil_moisture': 72, 'tempera
ture': 63, 'humidity': 82}
Published data successfully: %s {'soil_moisture': 81, 'tempera
ture': 116, 'humidity': 14}
Published data successfully: %s {'soil_moisture': 73, 'tempera
ture': 125, 'humidity': 80}
Published data successfully: %s {'soil_moisture': 93, 'tempera
ture': 75, 'humidity': 98}
Published data successfully: %s {'soil_moisture': 11, 'tempera
ture': -12, 'humidity': 69}
Published data successfully: %s {'soil_moisture': 86, 'tempera
ture': 39, 'humidity': 9}
Published data successfully: %s {'soil_moisture': 56, 'tempera
ture': -9, 'humidity': 88}
Published data successfully: %s {'soil_moisture': 84, 'tempera
ture': 117, 'humidity': 33}
Ln: 15 Col: 0
Type here to search
29°C
ENG
23:32
18-11-2022

CHAPTER 9

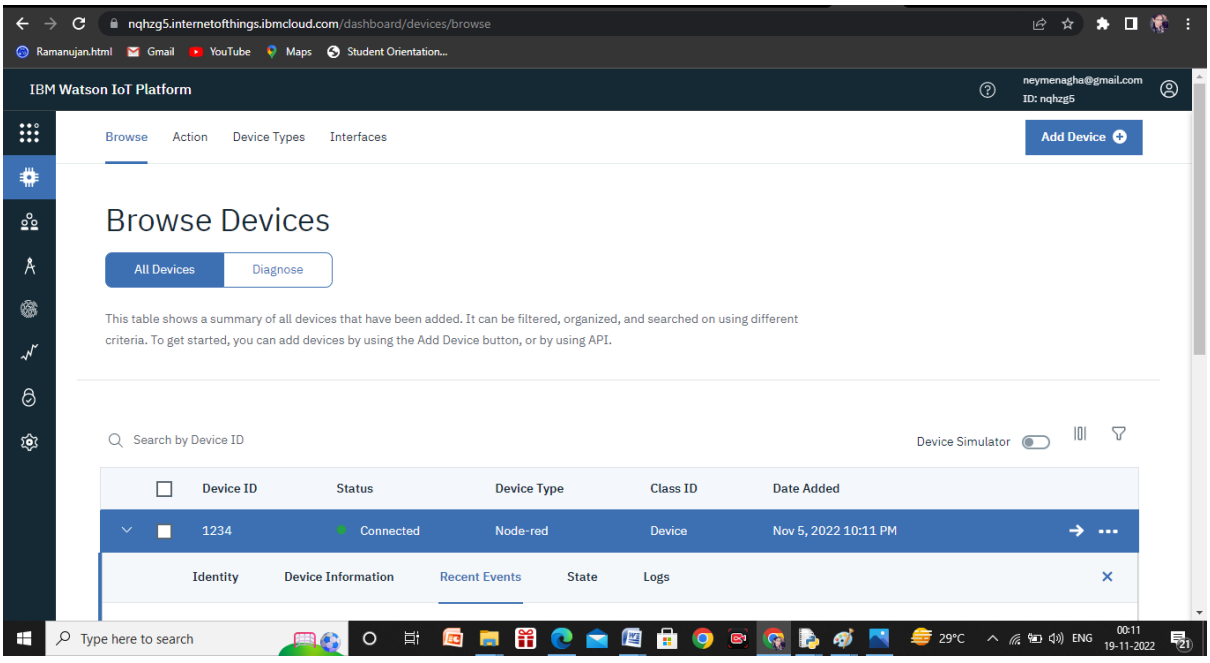
RESULTS

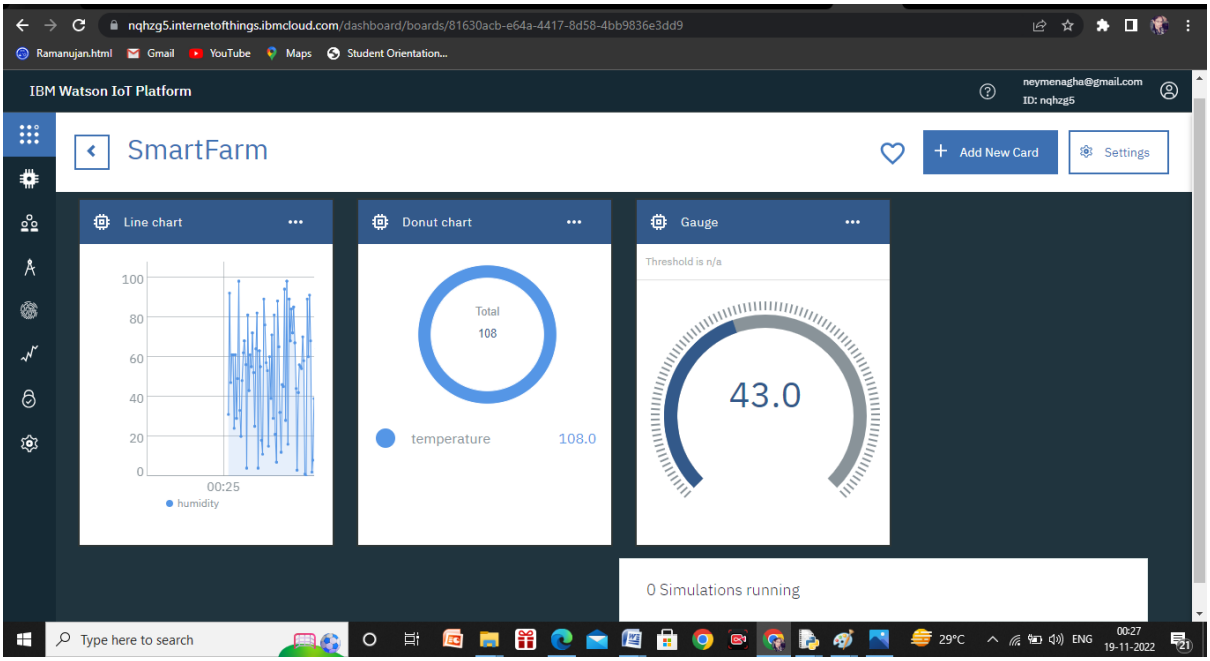
9.1 PERFORMANCE METRICS

NODE-RED CONNECTION:

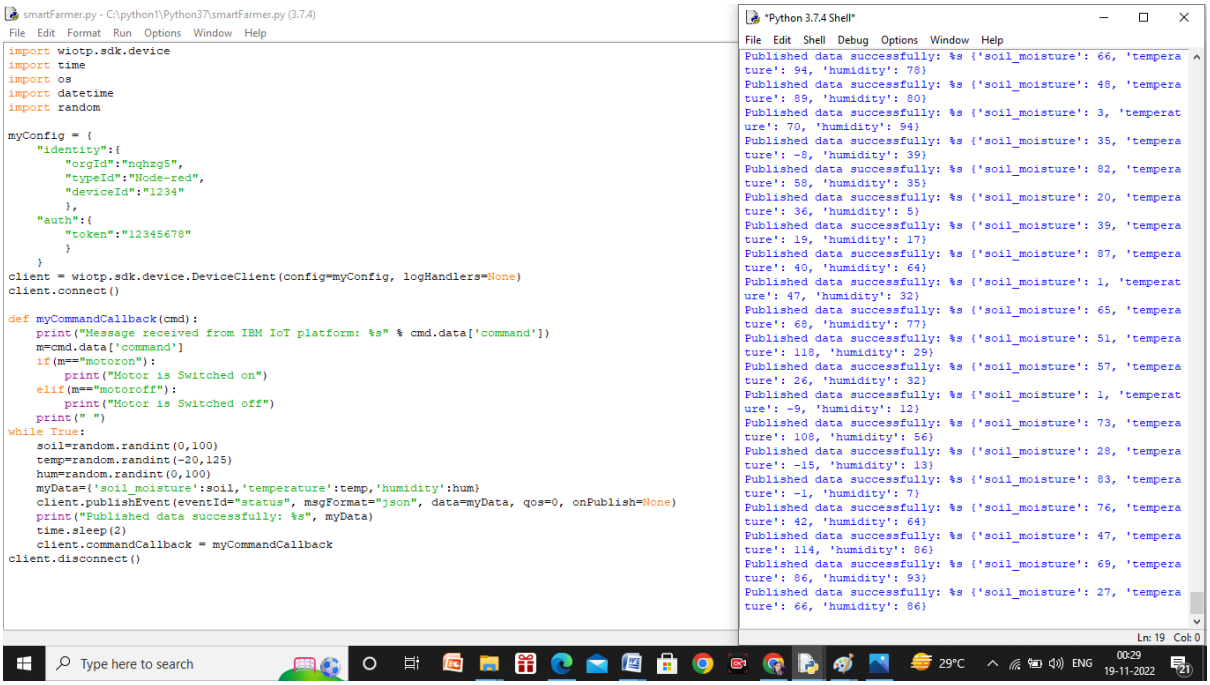


IBM WATSON IOT PLATFORM AND DEVICE





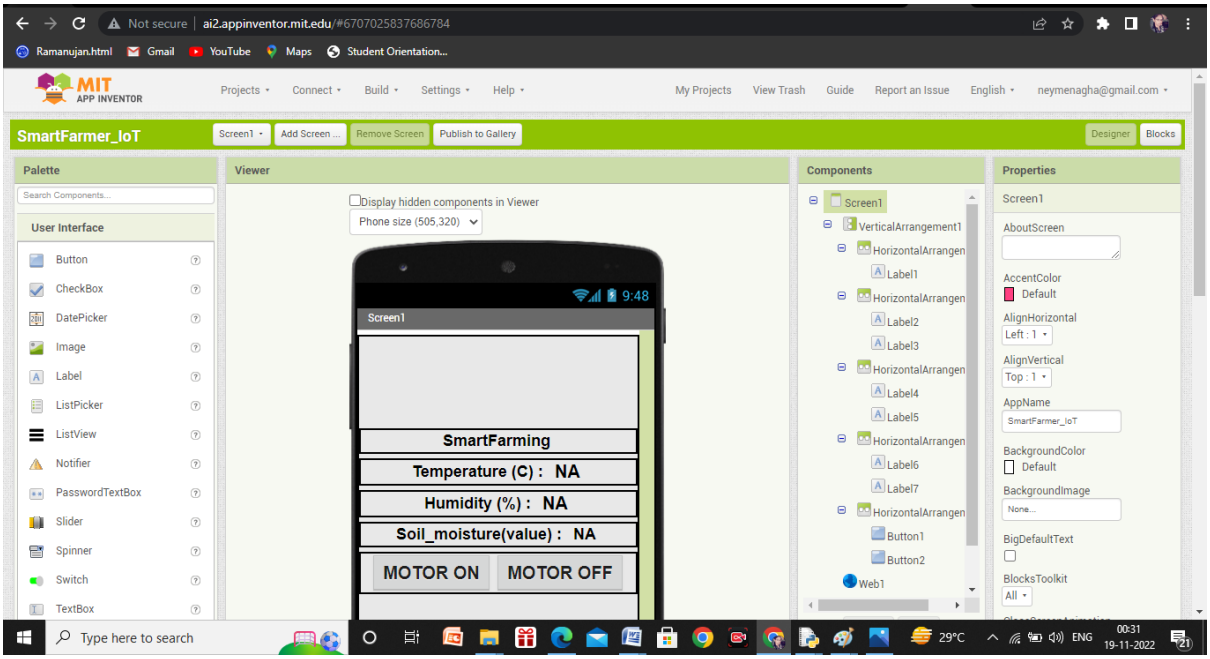
PYTHON SCRIPT:



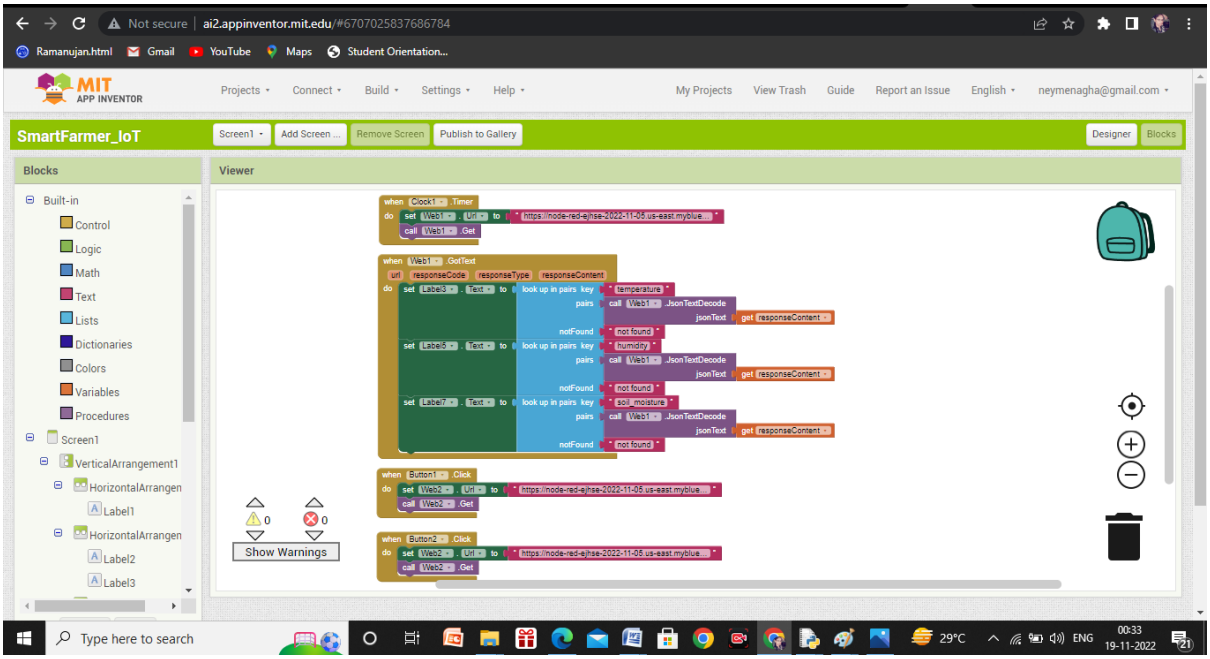
```
import wiotp.sdk.device
import time
import os
import datetime
import random

myConfig = {
    "identity":{
        "orgId":"nqhzg5",
        "typeId":"Node-red",
        "deviceId":"1234"
        },
    "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(-20,125)
    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(2)
    client.commandCallback = myCommandCallback
client.disconnect()
```

```
Published data successfully: %s {'soil_moisture': 66, 'tempera
ture': 94, 'humidity': 78}
Published data successfully: %s {'soil_moisture': 48, 'tempera
ture': 89, 'humidity': 80}
Published data successfully: %s {'soil_moisture': 3, 'temperat
ure': 70, 'humidity': 94}
Published data successfully: %s {'soil_moisture': 35, 'tempera
ture': -8, 'humidity': 39}
Published data successfully: %s {'soil_moisture': 82, 'tempera
ture': 58, 'humidity': 35}
Published data successfully: %s {'soil_moisture': 20, 'tempera
ture': 36, 'humidity': 5}
Published data successfully: %s {'soil_moisture': 39, 'tempera
ture': 19, 'humidity': 17}
Published data successfully: %s {'soil_moisture': 87, 'tempera
ture': 40, 'humidity': 64}
Published data successfully: %s {'soil_moisture': 1, 'temperat
ure': 47, 'humidity': 32}
Published data successfully: %s {'soil_moisture': 65, 'tempera
ture': 68, 'humidity': 77}
Published data successfully: %s {'soil_moisture': 51, 'tempera
ture': 118, 'humidity': 29}
Published data successfully: %s {'soil_moisture': 57, 'tempera
ture': 26, 'humidity': 32}
Published data successfully: %s {'soil_moisture': 1, 'temperat
ure': -9, 'humidity': 12}
Published data successfully: %s {'soil_moisture': 73, 'tempera
ture': 108, 'humidity': 56}
Published data successfully: %s {'soil_moisture': 28, 'tempera
ture': -15, 'humidity': 13}
Published data successfully: %s {'soil_moisture': 83, 'tempera
ture': -1, 'humidity': 7}
Published data successfully: %s {'soil_moisture': 76, 'tempera
ture': 42, 'humidity': 64}
Published data successfully: %s {'soil_moisture': 47, 'tempera
ture': 114, 'humidity': 86}
Published data successfully: %s {'soil_moisture': 69, 'tempera
ture': 86, 'humidity': 93}
Published data successfully: %s {'soil_moisture': 27, 'tempera
ture': 66, 'humidity': 86}
```

MOBILE APPLICATION:



BACKEND CONNECTIVITY:



MOBILE SCREEN:


Screen1
SmartFarming
Temperature (C) : 100
Humidity (%) : 7
Soil_moisture(value) : 57
MOTOR ON
MOTOR OFF

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Monitoring weather parameters such as temperature, pressure, humidity, soil moisture remotely.
- Controlling motors easily through buttons.
- Alert farmers in case of any calamities.
- Threshold values are set any anomalies will be reported to the farmer.
- User friendly and efficient.
- Low cost.

DISAVANTAGES:

- Sensors may sometime malfunction.
- Maybe inaccurate sometimes.
- Farmer needs internet connectivity.
- Farmer must have a phone and have basic knowledge to operate it.

CHAPTER 11

CONCLUSION

The Third Green Revolution is being paved over by smart farming and IoT-driven agriculture.

Agriculture is being displaced by the Third Green Revolution. This transformation depends on the coordinated use of data-driven analytics technologies, including robots, IoT, "big data" analytics, unmanned aerial vehicles (UAVs), and precision farming equipment.

The consumption of pesticides and fertilisers will decline while total productivity will increase in the future as predicted by the smart farming revolution. Through improved food traceability made possible by IoT technologies, food safety will ultimately enhance. It will also be good for the environment, for instance, by maximising inputs and treatments or by using water more effectively.

As a result, smart farming has the ability to bring about a more productive and sustainable kind of agricultural production that is based on a more precise and resource-efficient strategy. The human race's enduring goal will be fulfilled through new farms.

CHAPTER 12

FUTURE SCOPE

The need to increase farm productivity has become urgent due to factors such as the exponential expansion of the global population, which would require the globe to produce 70% more food by 2050, dwindling agricultural lands, and the depletion of limited natural resources. The problem has been made worse by the limited supply of natural resources including fresh water and arable land as well as declining yield patterns in a number of essential crops. The agricultural workforce's changing organisational structure is another hindering factor. In addition, agricultural work has decreased in the majority of the nations. The demand for physical labour has decreased as a result of the shrinking agricultural workforce, which has prompted the introduction of internet connectivity solutions in farming techniques.

By ensuring high yields, profitability, and environmental preservation, IoT solutions are aimed at assisting farmers in bridging the supply-demand gap. Precision agriculture is a method that uses IoT technology to assure the best use of resources to provide high crop yields and lower operational expenses. IoT technologies for agriculture include customised machinery, wireless connectivity, software, and IT services.

CHAPTER 13

APPENDIX

SOURCE CODE:

```
import wiotp.sdk.device
import time
import os
import datetime
import random
myConfig = {
    "identity":{
        "orgId":"nqhzg5",
        "typeId":"Node-red",
        "deviceId":"1234"
        },
    "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(-20,125)
    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(2)
    client.commandCallback = myCommandCallback
client.disconnect()
```

SOURCE CODE LINK:

https://github.com/IBM-EPBL/IBM-Project-2927-1658486979/tree/main/Final%20Deliverables/FINAL%20CODE

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-2927-1658486979

DEMO LINK:

https://youtu.be/vyPXSbYuy9c