

**SMARTFARMER - IOT ENABLED SMART FARMING
APPLICATION**

NALAIYA THIRAN PROJECT BASED LEARNING

Submitted by

Team ID: PNT2022TMID46067

TAMILARASAN.M	814719106063
NAVANEESWAAR.S.R	814719106038
MOHAMED ASLAM ABBAS	814719106037
PHLEGON BREEZER.P	814719106042

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONIC AND COMMUNICATION ENGINEERING

SRM TRP ENGINEERING COLLEGE

SRM NAGAR, IRUNGALUR,

TIRUCHIRAPPALLI

ANNA UNIVERSITY:: CHENNAI 600 025

NOV 2022

INDEX

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Non-Functional requirement

4.2 Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

6.3 Reports from JIRA

7. CODING & SOLUTIONING

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

1 Introduction:

1.1 overview:

In this project I 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 .

1.2 Purpose:

Agriculture plays a crucial role in the life of an economy. It is the backbone of our economic system, so improving the quality and way of production is crucial. Here comes the Smart Agriculture system. Smart agriculture helps in automated farming, collection of data from the field and then analyses it so that the farmer can make accurate decision in order to grow high quality crop.

IoT based Smart Farming improves the entire Agriculture system by monitoring the field in real-time. With the help of sensors and interconnectivity, the Internet of Things in Agriculture has not only saved the time of the farmers but has also reduced the extravagant use of resources such as Water. and Electricity.

2 Literature Survey:

2.1 Existing problem

Agriculture is extremely dependent on the climate. Temperature increases and carbon dioxide can boost some crop yields depending on the location; but other conditions must also exist, such as humidity, pressure, and water availability. Although slight warming and more carbon dioxide in the atmosphere could benefit some plants to grow faster, severe warming, floods, and drought would reduce yields. Farmer need to spend a lot of time to maintain these. Heat is not

the only extreme weather. Extreme cold can benefit farmers by freezing the soil deep beneath the ground. In parts of the upper Midwest, frost depths exceed 40 inches. A deep frost depth can aid farmers in diverse ways. The cold helps nitrogen that is applied in the fall from vaporizing during the winter. The cycle of freezing and thawing of water helps soften the soil after the thaw. Extreme cold and frozen soils also reduce the survival rate of some insects.

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 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.

2.3 Problem Statement Definition

Customer Problem Statement Template: Raja is a farmer, his son completed Engineering course stream in Electronic & Communication Engineering. His son gave him the idea to improve agriculture with the help of the technology he learnt. It also helps him in reducing manpower. His son is working on a new idea to improve the irrigation facility, soil fertility and crop rotation. This problem can be actively solved with the help of the application he is building.

I'm	I'm a farmer
I'm trying to	Increase the yield and reduce the labour
But	It increases cost as well as risk in managing.
Because	Of less income and higher expenditure
Which makes me feel	Angry and depressed

Example:

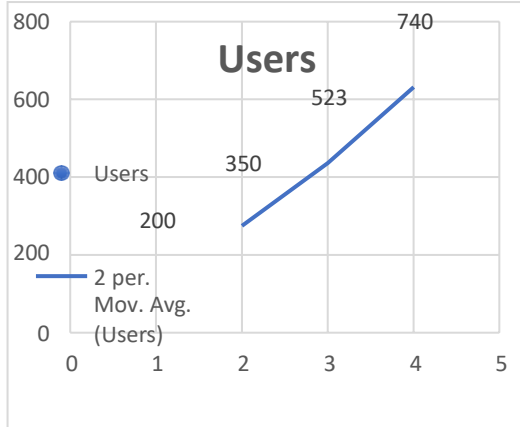


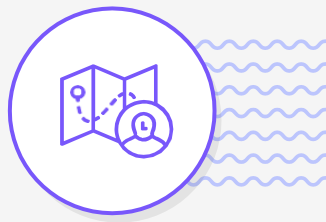
3 IDEATION & PROPOSED SOLUTION:

3.3 Proposed Solution:

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	1. 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	<p>It is better than the present modern irrigation system by using this method we can control soilerosion. There will be better production yield.</p>												
5.	Business Model (Revenue Model)	<p>IoT can also support the growth and sales of e-commerce companies.</p>  <table><caption>Users</caption><thead><tr><th>Period</th><th>Users</th></tr></thead><tbody><tr><td>0</td><td>400</td></tr><tr><td>1</td><td>200</td></tr><tr><td>2</td><td>350</td></tr><tr><td>3</td><td>523</td></tr><tr><td>4</td><td>740</td></tr></tbody></table>	Period	Users	0	400	1	200	2	350	3	523	4	740
Period	Users													
0	400													
1	200													
2	350													
3	523													
4	740													
6.	Scalability of the Solution	<p>It is definitely scalable we can increase the constraints when the problem arises.</p>												



Document an existing experience

Narrow your focus to a specific scenario or process within an existing product or service. In the **Steps** row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.

TIP

As you add steps to the experience, move each these “Five Es” the left or right depending on the scenario you are documenting.

SCENARIO	Entice	Enter	Engage	Exit	Extend
Browsing, booking, attending, and rating a local city tour					
Steps					
Interactions					
Goals & motivations					
Positive moments					
Negative moments					
Areas of opportunity					



Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

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

CS

Farmers are our customers

6. CUSTOMER CONSTRAINTS

CC

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

The availability of devices, proper network facilities, and budget are several constraints, and knowledge about the application.

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem

or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper

The most commonly used irrigation type is drip irrigation its common disadvantage is when the water is not filtered properly there will be clogs and tubes will get affected easily. In smart framing, we can use solar-powered smart irrigation systems to overcome this.

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

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

To make farming easier and more quantitatively.

- Monitoring the farm's climatic conditions.
- Automatic systems for irrigation and fertilization.
- Soil analysis.

9. PROBLEM ROOT CAUSE

RC

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

When there is no knowledge about the soil problem arising on what to be sowed, climatic conditions also play an important role. Knowledge of how to water the plants accordingly.

7. BEHAVIOUR

BE

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

The customers will reach us when they don't have idea on how to analysis the soil and to improve the current irrigation system.

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

<p>3. TRIGGERS TR</p> <p>What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.</p>	<p>10. YOUR SOLUTION SL</p> <p>If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.</p>	<p>8. CHANNELS of BEHAVIOUR CH</p> <p>8.1 ONLINE</p> <p>What kind of actions do customers take online? Extract online channels from #7</p>
<p>To get correct accuracy on what to be done on the farm and to produce more crops and livestock quantitatively.</p>	<p>There will be less weed growth, maximum use of water efficiently, control of soil erosion and, maximum crop yield.</p>	<p>8.2 OFFLINE</p> <p>What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.</p>
<p>4. EMOTIONS: BEFORE / AFTER EM</p> <p>How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.</p> <p>When the productivity increases farmers will be satisfied. They will not worry about the loss. Irrigation will be more efficient than before.</p>		<p>We will reach the customers directly to ask about their problem matching our application and provide with them knowledge about our application to make their farming even more easier.</p> <p>In online mode will be doing do digital marketing using advertisements.</p>

Project Design Phase-I Solution Fit Template

Date	21 October 2022
Team ID	PNT2022TMID46067
Project Name	Smart Farmer-IoT Enabled Smart Farming Application
Maximum Marks	2 Marks

4 REQUIREMENT ANALYSIS

4.1 Non-Functional Requirements:

Solution Requirements (Functional & Non-functional)

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It uses remote sensors, analytical tools, and the whole system is monitored and managed through websites. This makes them User friendly of the system and no use of this product. No prior knowledge required.
NFR-2	Security	This system includes data masking, which is the process of removing all personally identifiable information from data, such as names, addresses, geographical identifiers, and access controls that help maintain privacy and security.
NFR-3	Reliability	It is possible to endure extreme weather events and open space circumstances by using sensors, specialised software, and IOT platforms. The system can last a longer time and delivers reliable data measurement.
NFR-4	Performance	Utilizing contemporary technical innovations aids in bridging the gap between production and yields in terms of both quantity and quality. Data Ingestion ensures quick action and less harm to the crops while boosting system performance by collecting and importing data from the many sensors for usage in real-time or database storage.
NFR-5	Availability	By incorporating new components with superior characteristics, the current system can be made better.

		By connecting data about crops (or weather) and equipment to automatically modify temperature and humidity, farming equipment can be adjusted automatically.
NFR-6	Scalability	<p>The cloud database deployment used by this system can be thought of as the medium in between the hardware system and the user's mobile app.</p> <p>The proposed method is scalable thanks to increased productivity, decreased operating costs, and precise farm and field evaluation.</p>

4.2 Functional Requirement:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Login	Check username and password Check access from another device
FR-4	Management of data	Managing data of crop conditions Managing data of weather conditions
FR-5	Management of Modules	Managing user Managing admins Managing roles of access
FR-6	Logout	Exit

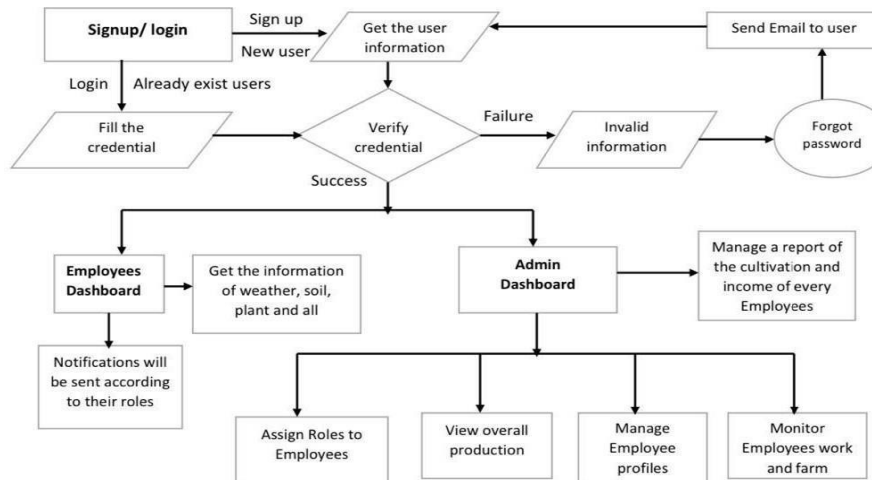
5.PROJECT DESIGN

5.1 Data Flow Diagram :

Data Flow Diagrams:

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

Smart Farming Data Flow:

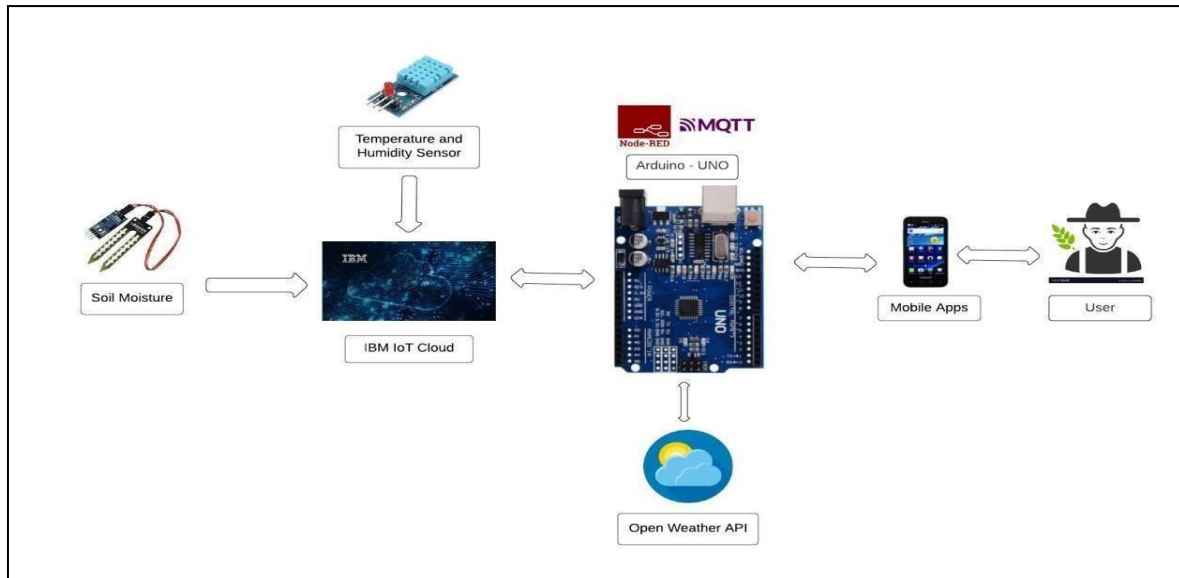


5.2 User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a Customer, 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 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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through 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 customer, I need to receive notification and details.	I get the details about what need to be done in different weather condition.	High	Sprint-1
Customer (Web user)		USN-7	As a user, I can reset my password if I forgot the old one.	I can use my account even if I forgot my password.	Medium	Sprint-2
Customer Care Executive	Know more	USN-8	As a user,I will be learn more about the work to be done.	Give more details from the data.	Medium	Sprint-3
Administrator	Assignment of roles	USN-9	As a admin, I will be able to assign role to the user.	I can assign role to the users.	High	Sprint-1
		USN-10	As a admin,I can note done the progress of all the expense of the work done.	I can note down	Low	Sprint-3

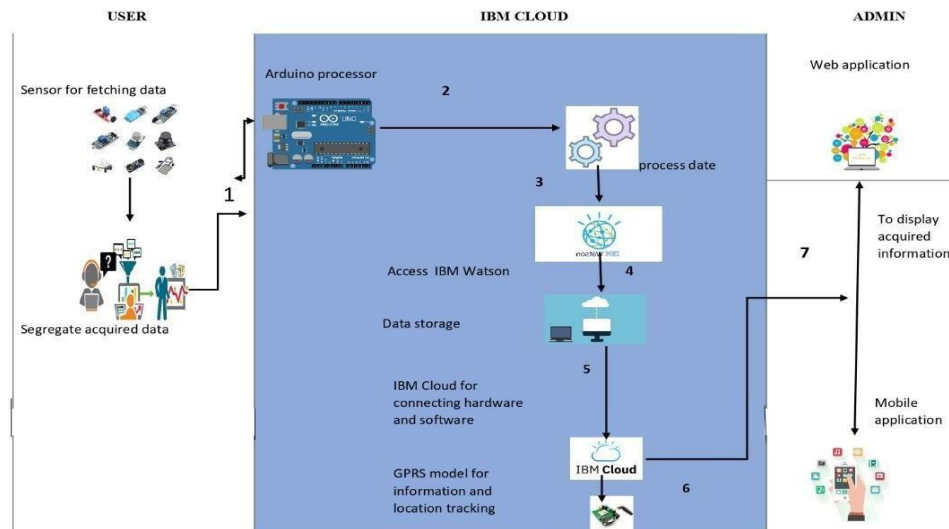
5.3 Solution & Technical Architecture:



- Utilizing a variety of sensors, the various soil parameters (temperature, humidity, and soil moisture) are measured, and the results are saved in the IBM cloud.
 - The processing unit used to process weather data from weather API and data from sensors is called Arduino UNO.
 - The hardware, software, and APIs are wired using the programming tool Node-red.
 - For communication, the MQTT protocol is used.
 - Through a smartphone application created with the aid of MIT App Inventor, the user is given access to all the collected data. Depending on the sensor results, the user may decide whether to irrigate the crop or not using an app.
- *They can control the motor switch from a distance using the app.

Technical Architecture:

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



Guidelines:

- . Include all the processes (As an application logic / Technology Block)
- . Provide infrastructural demarcation (Local / Cloud)
- . Indicate external interfaces (third party API's etc.)
- . Indicate Data Storage components / services
- . Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. WebUI, Mobile App, Chatbot etc.	MIT app
2.	Application Logic-1	The code will incorporate a number of circumstances, such as controlling water flow based on moisture content and humidity levels, and notifying users through message and mail if temperatures rise above a specific threshold.	Java / Python
3.	Application Logic-2	In this case, we can create a device and then design the software process by adding Node RED as an interface.	IBM Watson STT service
4.	Application Logic-3	Here, the conditions and sensed data can be compared to determine the ultimate outcome.	IBM Watson Assistant
5.	Database	To allow users to obtain data whenever needed, we can store all the data in SQL or any other database.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	The fetched data can be stored in a file in IMB Block Storage or local filesystem for analysis	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Temperature sensor	Monitors the temperature of the crop	sensor
9.	Soil moisture sensor (Tensiometers)	Monitors the soil temperature	sensor
10.	RTC module	Date and time configuration	
11.	Humidity sensor	Monitors the humidity	sensor

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	Here, we are using the IBM Cloud, which is a very safe location from where we can store data and access it as needed and encryption.	Encryption
3.	Scalable Architecture	Cloud-based IoT is a solution that is growing more and more popular and desired. In this study, an architecture specifically developed for monitoring cattle using Internet of Things (IoT) devices and a wide range of cloud native applications is presented. IBM executed a stress test to demonstrate the viability of the designed architecture for data processing.	IBM Cloud
4.	Availability	This application has a lot of important features available. Instead of wasting time by staying on the farm and monitoring the conditions, we have the moisture, humidity, and temperature which will denote the corresponding quantities. Additionally, we have both automatic and manual mode so once the certain conditions are met pump will be on/off and messages will be sent when needed so the farmer only needs to check the message in their phone and can take decisions in accordance with	IBM Watson IoT, IBM Cloud, Weather API'S, Analytics, Sensor Networks
5.	Performance	Quality improvement: Farmers may better comprehend the intricate relationships between environmental factors and crop quality using soil and crop sensors, aerial drone surveillance, and farm mapping.	IBM Watson IoT, Weather API'S, Analytics, Sensor Networks, IBM Cloud

		<p>Increased efficiency: Farmers must produce more despite failing soil, dwindling land availability, and growing weather variability. Farmers can monitor their product and environmental conditions in realtime thanks to IoT-enabled agriculture. They can quickly gain insights, anticipate problems before they arise, and decide how to prevent them using knowledge. IoT solutions for farming also include automation, such as demand-based irrigation, fertilisation, and robot harvesting.</p> <p>Increased Usage: 70% of the population will reside in cities by the time there are 9 billion people on the earth. Short food supply chains are made possible by IoT-based greenhouses and hydroponic systems, which should be able to feed everyone. Smart closed-cycle agricultural systems make it possible to grow food virtually anywhere, even on the walls and rooftops of buildings, in shipping containers, and, of course, inside everyone's cosy homes.</p> <p>Reduced resources: Agriculture in plenty IoT solutions are geared toward maximising the use of resources, including land, water, and energy. IoTbased precision farming depends on data gathered from various field sensors, which enables farmers to precisely allocate the right amount of resources to each plant.</p> <p>Agility: One of the benefits of using IoT in agriculture is the increased agility of the processes. In the conditions of extreme weather changes, new capabilities help agriculture professionals save the crops.</p>	
--	--	---	--

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning, Estimation & Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Simulation creation	USN-1	python code	2	High	Tamil, Navanees, Abbas
Sprint-2	Software	USN-2	Creating device in the IBM Watson IoT platform, workflow for IoT scenarios using Node-Red	2	High	Breezer, Abbas, Navanees
Sprint-3	MIT App Inventor	USN-3	Develop an application for the Smart farmer project using MIT App Inventor	2	High	Navanees, Tamil
Sprint-3	Dashboard	USN-3	Design the Modules and test the app	2	High	Abbas
Sprint-4	Web UI	USN-4	To make the user to interact with software.	2	High	Tamil, Abbas, Breezer, Navanees

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	7 Days	30 Oct 2022	06 Nov 2022	20	29 Oct 2022
Sprint-2	20	9 Days	31 Oct 2022	09 Nov 2022		05 Oct 2022
Sprint-3	20	6 Days	06 Nov 2022	13 Nov 2022		12 Oct 2022
Sprint-4	20	6 Days	11 Nov 2022	17 Nov 2022		15 Oct 2022

6.2 Milestone and Activity List

Title	Description	Date
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	09 OCTOBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements.	09 OCTOBER 2022
Brainstorming ideas	List the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	09 OCTOBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	19 OCTOBER 2022
Problem Solution Fit	Prepare problem - solution Fit document.	19 OCTOBER 2022
Solution Architecture	Prepare solution Architecture document.	23 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	24 OCTOBER 2022
Data Flow Diagrams	Draw the data flow Diagrams and submit for review.	24 OCTOBER 2022
Technology Architecture	Architecture diagram.	25 OCTOBER 2022
Sprint Delivery	Prepare the Sprint delivery on Number of Sprint planning meetings organized, Minutes of meeting recorded.	04 NOVEMBER 2022
Milestone & Activity List	Prepare the milestones & Activity list of the project.	04 NOVEMBER 2022
Project Development Delivery of Sprints	Develop & submit the developed code by testing it.	14 NOVEMBER 2022

7. CODING & SOLUTIONING

7.1 Feature 1:

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.

Installation :

- First install npm/node.js
- Open cmd prompt
- Type => npm install node-red

To run the application :

- Open cmd prompt
- Type=>node-red
- Then open <http://localhost:1880/> in browser

IBM Watson IoT Platform:

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

Steps to configure:

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

- Create API key and store API key and token elsewhere.

Python IDE:

Install Python3 compiler Install any python IDE to execute python scripts. The python code subscribed to IoT platform to form the connectivity layer with node -red commands.

Connecting IoT Simulator to IBM Watson IoT Platform:

Give the credentials of your device in IBM Watson IoT Platform Click on connect My credentials given to simulator are:

```
organization = "ie8mpi"
```

```
deviceType = "IoT_device"
```

```
deviceId = "IoT_device_1"
```

```
authMethod = "token"
```

```
authToken = "12345678"
```

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

- You will receive the simulator data in cloud
- You can see the received data in Recent Events under your device
- Data received in this format(json) You can see the received data in graphs by creating cards in Boards tab
- You will receive the simulator data in cloud
- You can see the received data in Recent Events under your device
- Data received in this format(json)

```
{  
  "d": {
```

```
▪ "name": "abcd",  
▪ "temperature": 17,  
▪ "humidity": 76,  
▪ "Moisture ": 25  
}  
}
```

Configuration of Node-Red to collect IBM cloud data:

The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red. Once it is connected Node-Red receives data from the device. Display the data using debug node for verification. Connect function node and write the Java script code to get each reading separately. The Java script code for the function node is: `msg.payload=msg.payload.d.temperature` return msg; Finally connect Gauge nodes from dashboard to see the data in UI

7.2 FEATURE 2:

Configuration of Node-Red to collect data from OpenWeather

The Node-Red also receive data from the OpenWeather API by HTTP GET request. An inject trigger is added to perform HTTP request for every certain interval. HTTP request node is configured with URL we saved before in section The data we receive from OpenWeather after request is in JSON

```
{ "coord": { "lon": 79.0667, "lat": 12.2167 }, "weather": [ { "id": 804, "main": "Clouds", "description": "overcast clouds", "icon": "04n" } ], "base": "stations", "main": { "temp": 295.77, "feels_like": 296.54, "temp_min": 295.77, "temp_max": 295.77, "pressure": 1015, "humidity": 94, "sea_level": 1015, "grnd_level": 995 }, "visibility": 10000, "wind": { "speed": 2.05, "deg": 29, "gust": 4.22 }, "clouds": { "all": 96 }, "dt": 1668440978, "sys": { "country": "IN", "sunrise": 1668386454, "sunset": 1668428130 }, "timezone": 19800, "id": 1254327, "name": "salem", "cod": 200 }
```

In order to parse the JSON string we use Java script functions and get each parameters

In order to parse the JSON string we use Java script functions and get each parameters

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

In the above Java script code we take temperature parameter into a new variable and convert it from kelvin to Celsius Then we add Gauge and text nodes to represent data visually in UI. In the above Java script code we take temperature parameter into a new variable and convert it from kelvin to Celsius Then we add Gauge and text nodes to represent data visually in UI.

Configuration of Node-Red to send commands to IBM cloud ibmiot out node:

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

Here we add two buttons in UI

1 -> for motor on

2 -> for motor off

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

The Java script code for the analysis is:

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

Adjusting User Interface:

To display the parsed JSON data a Node-Red dashboard is created. Here we are using Gauges, text and button nodes to display in the UI and helps to monitor the parameters and control the farm equipment. Below images are the Gauge, text and button node configurations.

Using MIT APP Inventor:

It facilitates farmer to know the current parameters of their land through mobile app which already connected with web UI. Farmer can turn on or off the motor according to the condition of the field.

FEATURE 1

PYTHON CODE:

```
import time

import sys

import ibmiotf.application
import ibmiotf.device

#Provide your IBM Watson Device Credentials

organization = "ie8mpi"

deviceType = "IoT_device"

deviceId = "IoT_device_1"

authMethod = "token"

authToken = "12345678"


def myCommandCallback(cmd):

    print("Command received: %s" %cmd.data)

    if cmd.data['command']=='motoron':

print("MOTOR ON IS RECEIVED")


    elif cmd.data['command']=='motoroff':

print("MOTOR OFF IS RECEIVED")


    if cmd.command == "setInterval":

        if 'interval' not in cmd.data:

            print("Error - command is missing required information: 'interval'")

        else:

            interval = cmd.data['interval']

    elif cmd.command ==

"print":          if 'message' not

in cmd.data:
```

```

        print("Error - command is missing required information: 'message'")
    else:
        output=cmd.data['message'] print(output)

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()

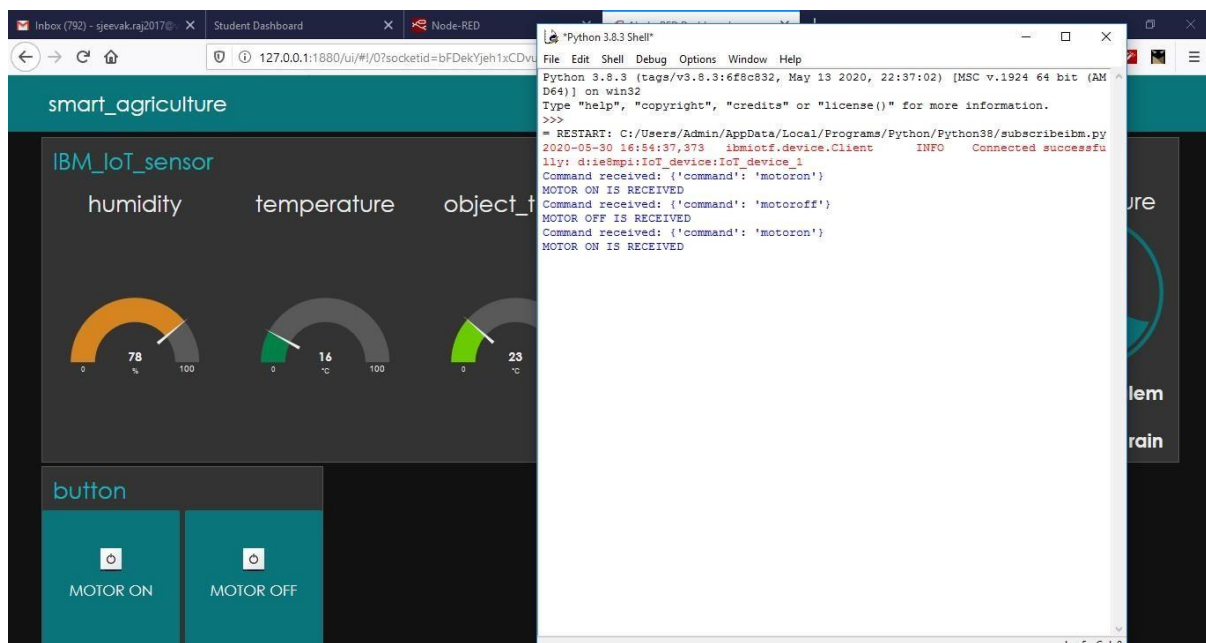
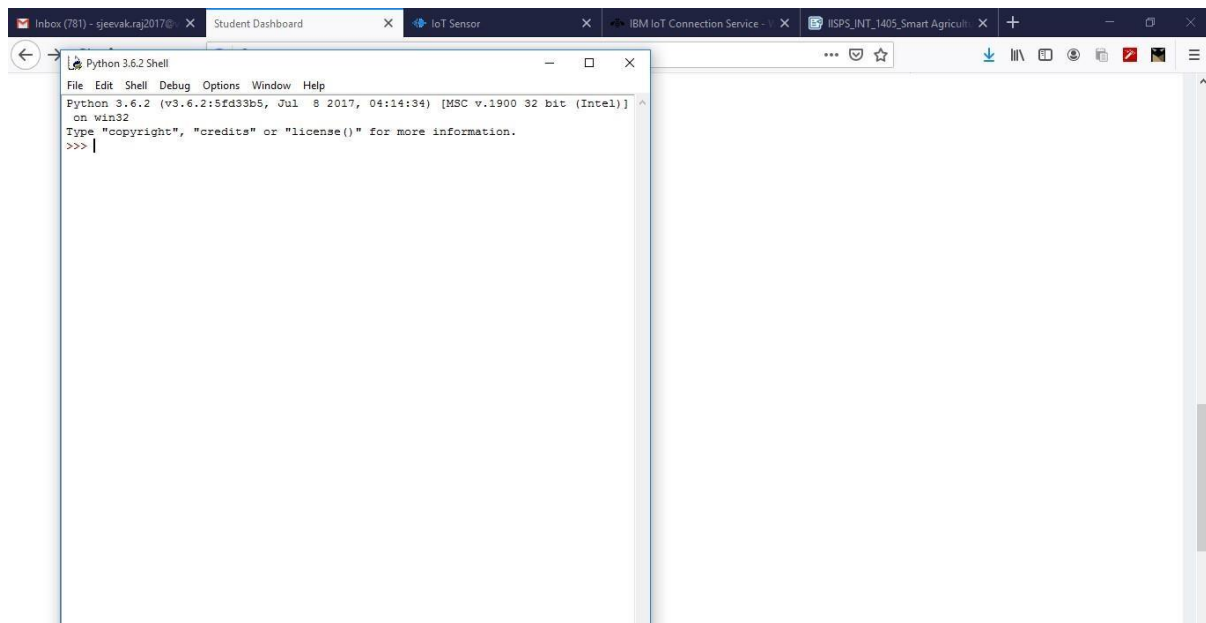
# Connect and send a datapoint "hello" with value "world" into the cloud as an event of
type
"greeting" 10 times
deviceCli.connect()

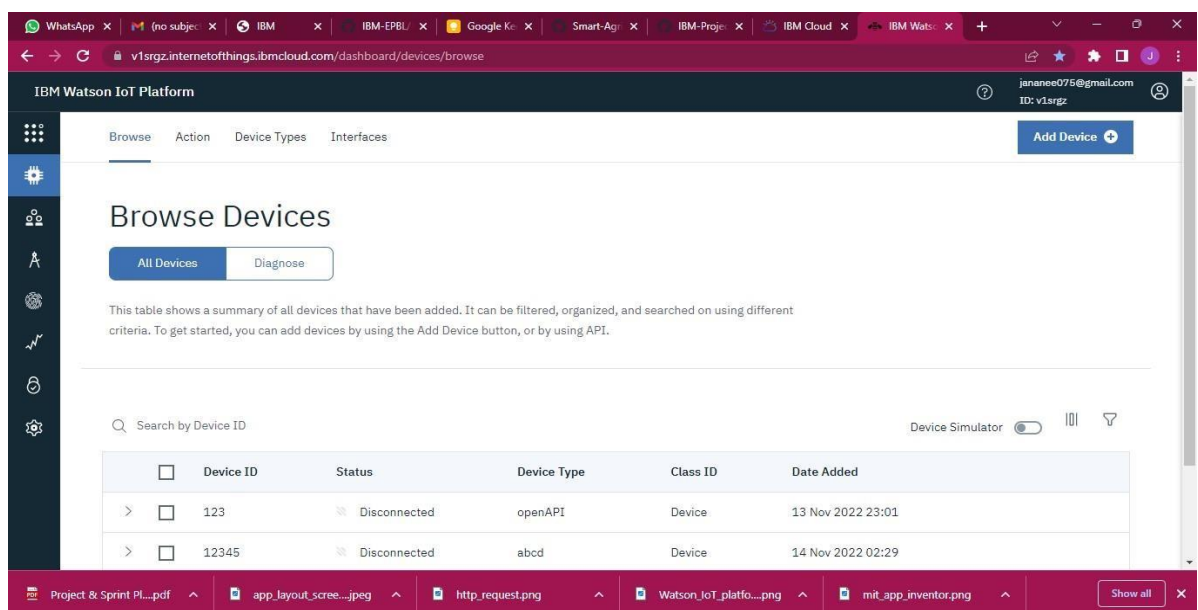
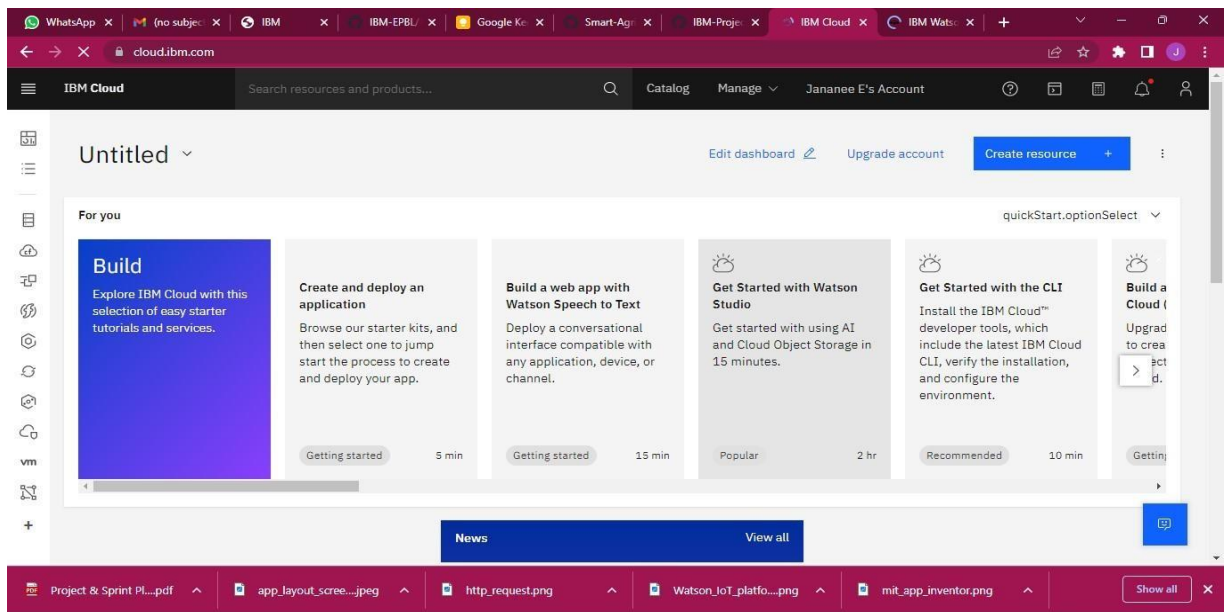
while True:

    deviceCli.commandCallback = myCommandCallback

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

```





WhatsApp (no subject) IBM IBM-EPBL Google Ke Smart-Ag IBM-Proje IBM Cloud IBM Wats v1srgz.internetofthings.ibmcloud.com/dashboard/boards/janane075@gmail.com-UsageDefaultBoard.1

IBM Watson IoT Platform janane075@gmail.com ID: v1srgz

Usage Overview

Total 2 devices

8.6 MB This month
0.0 MB Previous month

Data transferred

0.0 MB Data transferred today

Project & Sprint Pl...pdf app_layout_scre...jpeg http_request.png Watson_IoT_platfo...png mit_app_inventor.png Show all

127.0.0.1:1880/#flow/a4a75e6d.5b5a68

Node-RED Deploy

filter nodes

Flow 1 Flow 2

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

temperature

objectTemp

humidity

function

alert

temperature

object_temp

humidity

msg.payload

debug


all nodes all

Project & Sprint Pl...pdf app_layout_scre...jpeg http_request.png Watson_IoT_platfo...png mit_app_inventor.png Show all

smart_agriculture


IBM_IoT_sensor

humidity temperature object_temp



open_weather

humidity pressure temperature




region: Salem

weather_description overcast clouds

chart1


temperature



20:58:41

chart2


object temp



20:58:41

chart3

humidity



20:58:41

button

MOTOR ON MOTOR OFF

Node-RED

Flow 1 Flow 2

filter nodes

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

make request

http request

function

temperature

pressure

region

humidity

weather_description

msg payload

temperature

pressure

region abc

humidity

weather_description abc

127.0.0.1:1880/#9a87c2cc307ec

Project & Sprint Pl...pdf app_layout_scre...jpeg http_request.png Watson_IoT_platfo...png mit_app_inventor.png Show all

Python 3.8.3 Shell

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
* RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python38/subscribeibm.py
2020-05-30 16:54:37,373 ibmiotf.device.Client INFO Connected successfully
lty: d:\ieSmp\IoT_device\IoT_device_1
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
```

smart_agriculture

IBM_IoT_sensor

humidity temperature object_t

78 % 16 °C 23 °C

button

MOTOR ON MOTOR OFF

Watson IoT Sensor Simulator IoT_device_1

Temperature

15°C

connected

swipe left/right for more

Python 3.8.3 Shell

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python38/subscribeibm.py
2020-05-30 16:54:37,373 ibmiotf.device.Client INFO Connected successfully
lly: d:ieSmpi:IoT_device:IoT_device_1
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
```

smart_agriculture

IBM_IoT_sensor

humidity temperature object_t

78 % 16 °C 23 °C

button

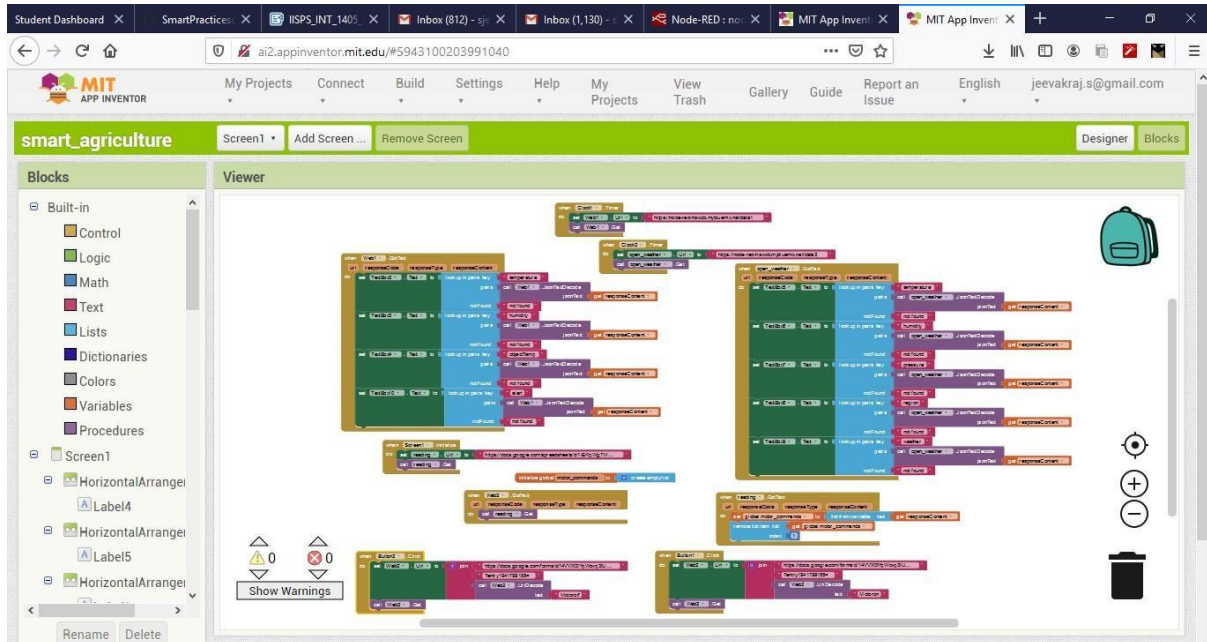
MOTOR ON MOTOR OFF

api.openweathermap.org/data/2.5/weather?q=tiruvannamalai&appid=c5ac3c5487ad57ad7d180da88716ea93

```
{
  "coord": {
    "lon": 79.8667,
    "lat": 12.2167
  },
  "weather": [
    {
      "id": 804,
      "main": "Clouds",
      "description": "overcast clouds",
      "icon": "04n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 29.77,
    "feels_like": 29.54,
    "temp_min": 29.77,
    "temp_max": 29.77,
    "pressure": 1015,
    "humidity": 94,
    "sea_level": 1615,
    "grnd_level": 995,
    "visibility": 10000,
    "wind": {
      "speed": 12.05,
      "deg": 29,
      "gust": 4.22
    },
    "clouds": {
      "all": 96
    },
    "dt": 1668440978,
    "sys": {
      "country": "IN",
      "sunrise": 1668386454,
      "sunset": 1668428130,
      "timezone": 19800,
      "id": 1254327,
      "name": "Tiruvannamalai",
      "cod": 200
    }
  }
}
```

FEATURE 2

Develop an application for the Smart farmer project using MIT App Inventor and Design the Modules and test the app



MOBILE INTERFACE

Personal Hotspot : 1 connections, Used 431 MB

Screen1

Smart Agriculture Using IoT

From IBM IoT_sensor

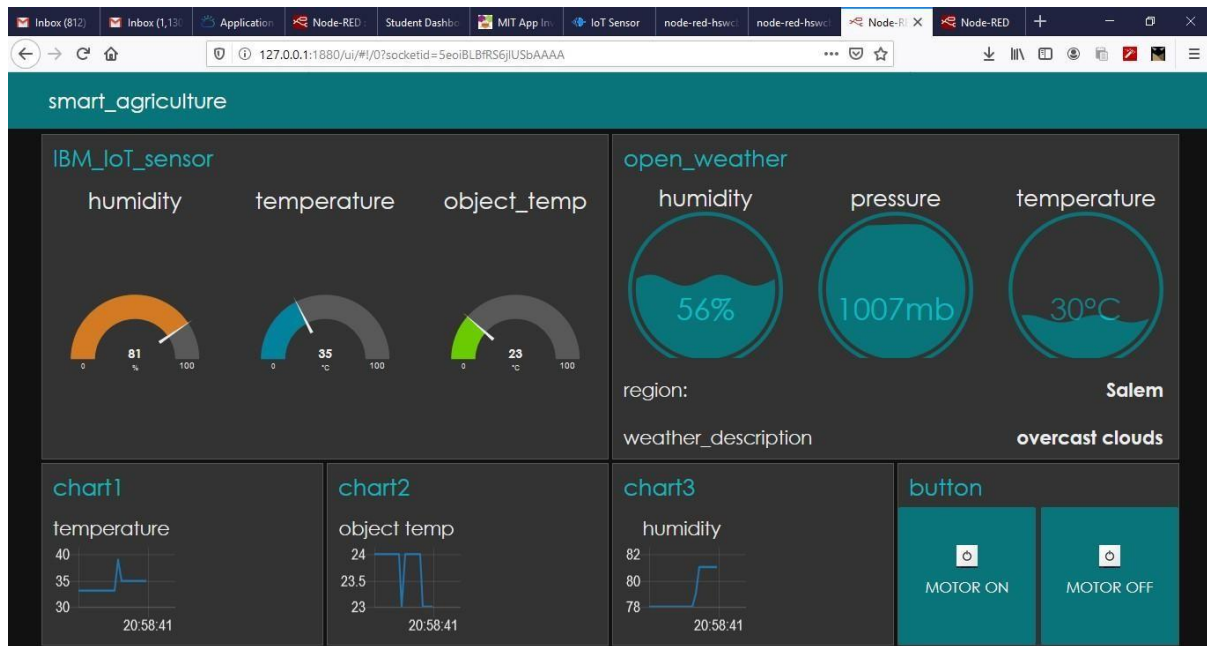
Temperature	15
Humidity	76
Object Temp	24

From Open_Weather

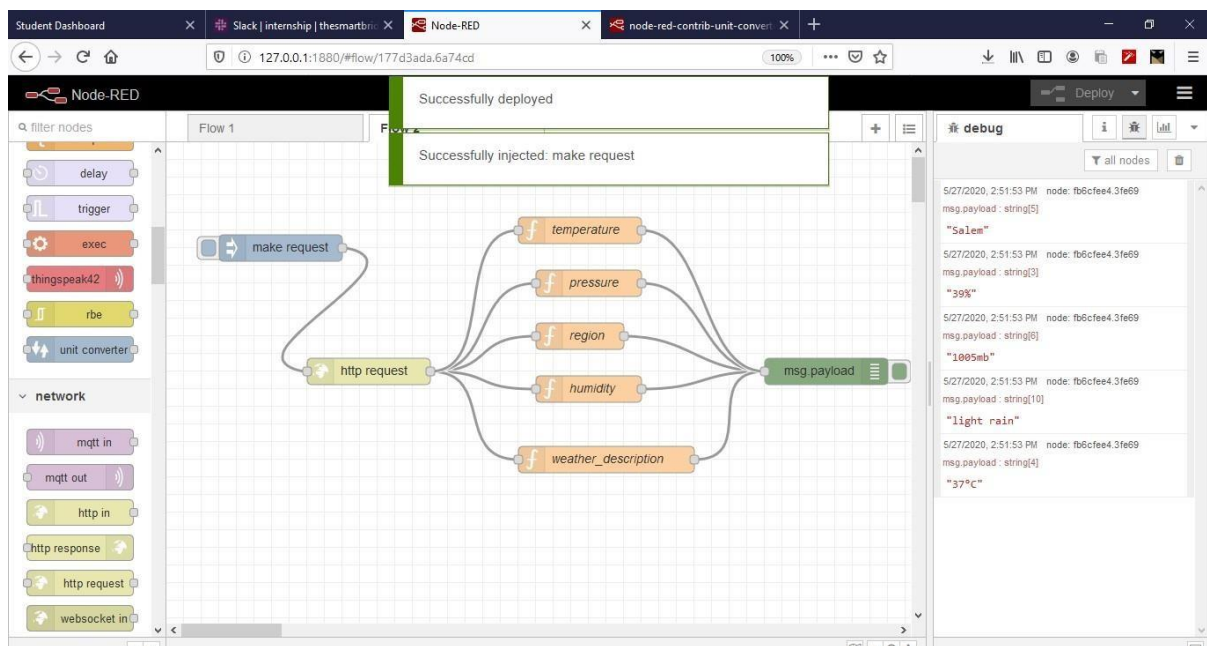
Temperature	30
Humidity	56
Pressure	1007
Region	Salem
Weather	overcast clouds
Alert	humidity is high

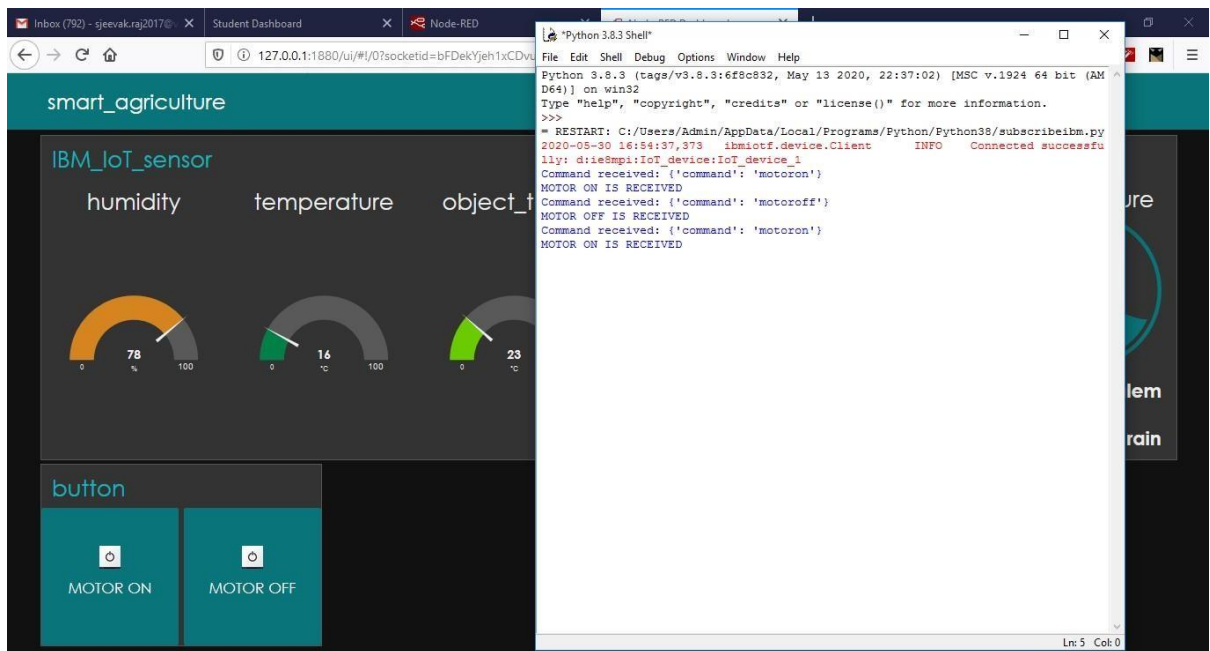
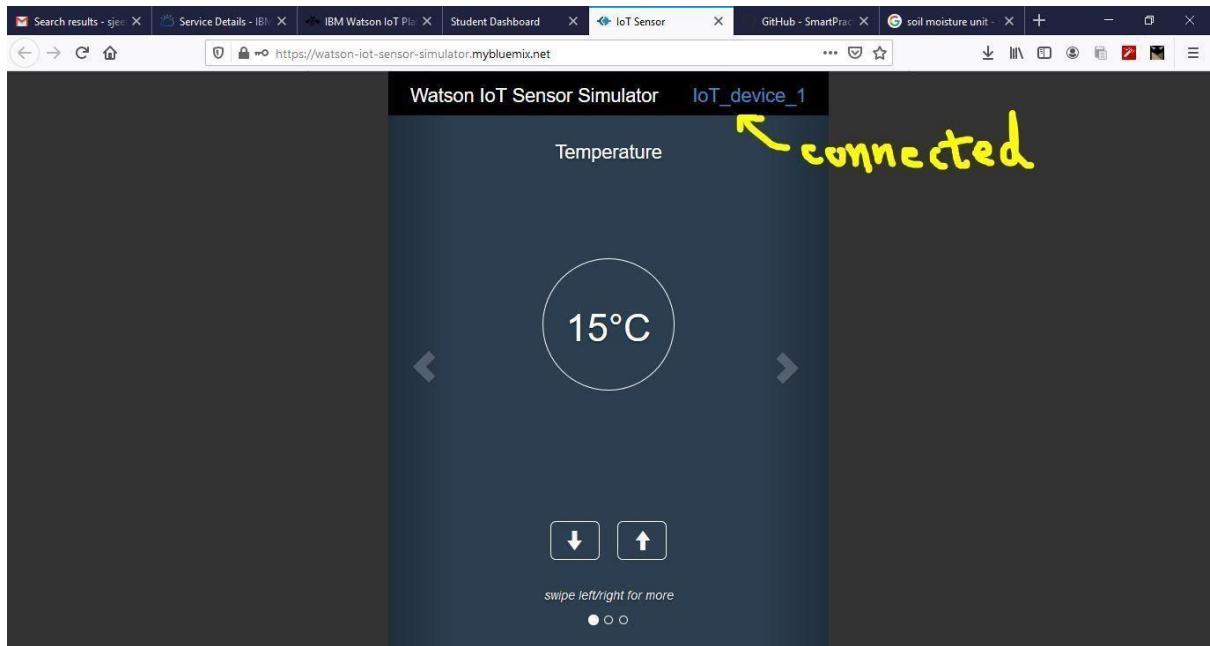
Motor ONMotor OFF

TEST THE UI INTERFACE :



SUCCESSFULLY DEPLOYED:





smart_agriculture

IBM_IoT_sensor

humidity temperature object

78 % 16 °C 23 °C

button

MOTOR ON MOTOR OFF

Python 3.8.3 Shell

```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
** RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python38/subscrib...
2020-05-30 16:54:37,373 ibmiotf.device.Client INFO Connected successfully
lly: d:\ieSmpi\IoT_device\IoT_device_1
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
Command received: {'command': 'motoroff'}
MOTOR OFF IS RECEIVED
Command received: {'command': 'motoron'}
MOTOR ON IS RECEIVED
```

Ln: 5 Col: 0

4:54 PM

8. TESTING

8.1 Test Cases

1

▼

:

×

✓

fx

A	B	C	D	E	F	G	H	I	J
Team ID : PNT2022TMID21901									
NFT - Risk Assessment									
No	Modules	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Volumen Changes	Risk Score	Justification
1	Node-Red	New	Low	No Changes	Moderate	Time Complexity	>5 to 10%	ORANGE	As we have seen the changes
2	IoT simulator	New	High	No Changes	Moderate	Time Complexity	6%	ORANGE	seen changes
3	Devices	New	High	No Changes	High	none	10%	GREEN	No changes
NFT - Detailed Test Plan									
S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/R	Approvals/SignOff					
1	Connects simulator with node-red	speed	Assumptions	Approvals					
End Of Test Report									
No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff	
	connects with web application and mobile application			Good performance	GO	None	Open	Approval	
1									

<

>

NFT - RA

DTP

+

:

<

8.2 User Acceptance Testing

The purpose of this document is to briefly explain the test coverage and open issues of the “SmartFarmer - IoT Enabled Smart Farming Application” project at the time of the release to User Acceptance Testing (UAT). Increasing control over production leads to better cost management and waste reduction. The ability to trace anomalies in crop growth or livestock health, for instance, helps eliminate the risk of losing yields. Additionally, automation boosts efficiency. Smart farming reduces the ecological footprint of farming. Minimized or site-specific application of inputs, such as fertilizers and pesticides, in precision agriculture systems will mitigate leaching problems as well as the emission of greenhouse. This report shows the number of resolved or closed bugs at each severity level, and how they were resolved. This report shows the number of test cases that have passed, failed, and untested.

1.DEFECT ANALYSIS

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

2.TEST ANALYSIS

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

9. RESULT:

9.1 PERFORMANCE METRICS

Hence a helpful and useful system is built for farmers to assist them in farming and also prevent them from natural calamities. It also saves farmers time to maintain all these things as this is working on cloud he can turn on/off motor from anywhere so basically it helps farmers and make them relived thus helping our economy to grow.

10.ADVANTAGES & DISADVANTAGES:

Advantage:

- 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

Disadvantage:

- sensors may sometime malfunction
- maybe inaccurate sometimes
- farmer needs internet connectivity
- farmer must have a phone and have basic knowledge to operate it

Applications:

- Monitoring of Climate Conditions -Probably the most popular smart agriculture gadgets are weather stations, combining various smart farming sensors. Located across the field, they collect various data from the environment and send it to the cloud. The provided measurements can be used to map the climate conditions, choose the appropriate crops, and take the required measures to improve their capacity (i.e. precision farming).

- Greenhouse Automation-In addition to sourcing environmental data, weather stations can automatically adjust the conditions to match the given parameters. Specifically, greenhouse automation systems use a similar principle.
- Crop Management - One more type of IoT product in agriculture and another element of precision farming is crop management devices. Just like weather stations, they should be placed in the field to collect data specific to crop farming; from temperature and precipitation to leaf water potential and overall crop health, these can all be used to readily collect data and information for improved farming practices.
- Cattle Monitoring and Management-Just like crop monitoring, there are IoT agriculture sensors that can be attached to the animals on a farm to monitor their health and log performance. This works similarly to IoT devices for pet care.
- End-to-End Farm Management Systems-A more complex approach to IoT products in agriculture can be represented by the so-called farm productivity management systems. They usually include a number of agriculture IoT devices and sensors, installed on the premises as well as a powerful dashboard with analytical capabilities and in-built accounting/reporting features.

11.CONCLUSION:

Smart Farming and IoT-driven agriculture are paving the way for what can be called a Third Green Revolution.The Third Green Revolution is taking over agriculture. That revolution draws upon the combined application of data-driven analytics technologies, such as precision farming equipment, IoT, “big data” analytics, Unmanned Aerial Vehicles (UAVs or drones), robotics, etc.

In the future this smart farming revolution depicts, pesticide and fertilizer use will drop while overall efficiency will rise. IoT technologies will enable better food traceability, which in turn will lead to increased food safety. It will also be beneficial for the environment, for example, more efficient use of water, or optimization of treatments and inputs. Therefore, smart farming has a real potential to deliver a more productive and sustainable form of agricultural production, based on a more precise and resource-efficient approach. New farms will finally realize the eternal dream of mankind.

12.FUTURE SCOPE:

With the exponential growth of world population, according to the UN Food and Agriculture Organization, the world will need to produce 70% more food in 2050, shrinking agricultural lands, and depletion of finite natural resources, the need to enhance farm yield has become critical. Limited availability of natural resources such as fresh water and arable land along with slowing yield trends in several staple crops, have further aggravated the problem. Another impeding concern over the farming industry is the shifting structure of agricultural workforce. Moreover, agricultural labor in most of the countries has declined. As a result of the declining agricultural workforce, adoption of internet connectivity solutions in farming practices has been triggered, to reduce the need for manual labor. IoT solutions are focused on helping farmers close the supply demand gap, by ensuring high yields, profitability, and protection of the environment. The approach of using IoT technology to ensure optimum application of resources to achieve high crop yields and reduce operational costs is called precision agriculture. IoT in agriculture technologies comprise specialized equipment, wireless connectivity, software and IT services.

13 APPENDIX:

Python code for the motor is:

<https://github.com/IBM-EPBL/IBM-Project-50102-1660892879>

Github repository link:

<https://github.com/IBM-EPBL/IBM-Project-50102-1660892879>

Demolink:

<https://clipchamp.com/watch/75KDd6xQmpX>

<https://youtu.be/pDfQs1TRydA>