Smart Farmer-IOT Enabled Smart Farming Application

A NAALAIYA THIRAN PROJECT REPORT

Submitted by

HARIPRASATH.T	(142219104034)
MUTHU SATHISH.P	(142219104072)
PORSELVI.S	(142219104083)
RAJALAKSHMI.V	(142219104093)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

Department of Computer Science and Engineering

SRM VALLIAMMAI ENGINEERING COLLEGE
(An Autonomous Institution)
SRM Nagar, Kattankulathur, Chengalpattu Dt – 603203,Tamil Nadu

NOVEMBER 2022

SRM VALLIAMMAI ENGINEERING COLLEGE
(An Autonomous Institution)
SRM Nagar, Kattankulathur Chengalpattu DT -603203,Tamilnadu
Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

Certified that this project report **"SMART FARMER-IOT ENABLED SMART FARMING APPLICATION"** is the bonafide work of **"HARIPRASATH.P (142219104034), MUTHUSATHISH.P (142219104072), PORSELVI.S (142219104083), RAJALAKSHMI.V (142219104093)"** who carried out the project work under my supervision.

MENTOR	EVALUATOR	HEAD OF THE DEPARTMENT
A. LALITHA	**A. VIDHYA**	**Dr. B. VANATHI**
Assistant professor	**Assistant professor**	**HEAD OF THE DEPARTMENT**
Department OF CSE	**Department OF CSE**	**Department OF CSE**
SRM Valliammai Engineering College	**SRM Valliammai Engineering College**	**SRM Valliammai Engineering college**
Chengalpattu-603203	**Chengalpattu-603203**	**Chengalpattu-603203**

Submitted for the end semester viva voce examination held on

____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We sincerely express our gratitude in depth to our esteemed Founder Chairman & Chancellor **Dr. T. R. Paarivendhar, Thiru. Ravi Pachamoothoo Chairman, Mrs. Padmapriya Ravi** Vice Chairman**, Ms. R Harini** Correspondent, **SRM VALLIAMMAI ENGINEERING COLLEGE** for the patronage on our welfare rooted in the academic year.

We express our sincere gratitude to our respected Director **Dr.B.Chidhambararajan, M.E., Ph.D.,** for his constant encouragement, which has been our motivation to strive towards excellence.

We thank our sincere **Principal Dr. M. Murugan, M.E., Ph.D.,** for his constant encouragement, which has been our motivation to strive towards excellence.

We extend our hand of thanks to our Head of the Department, **Dr.B.Vanathi, M.E., Ph.D.,** Professor for her unstinted support.

We are thankful to our project coordinator, **Mrs. S. Suma, M.S,** for proving us with the essential facilities

We are grateful to our internal guide **Dr.L.Karthikeyan, M.TECH., Ph.D.,,** Assistant Professor without whose invaluable guidance and encouragement, this project would not have been a success.

We also like to thank all Teaching and non-teaching staff members of our department, for their support during the course of the project. We finally thank our friends and family for their support during the course of the project

ABSTRACT

Information and communication technology is being improved from farm to fork to benefit farmers, croppers, and other users of associated intelligent services. Smart device and Internet of Things service development is integrated with the technological revolution. The agriculture sector must be expanded to feed the world's expanding population.

The Internet of Things makes it possible for smart farming techniques to boost agricultural output. IoT technologies provide historical and real-time data for predicting soil quality, weather conditions, and crop health, which is a service that benefits farmers. The expanded capability for process automation, evaluation, and waste reduction is provided by smart farming. As a result, all of these elements significantly improve the food items' quality and quantity while also lowering their production costs. This essay describes the innovative techniques used in the field of agriculture.

Keywords: Smart Farming, Internet of Things, Green House, IoT agriculture.

TABLE OF CONTENTS

LIST OF FIGURES

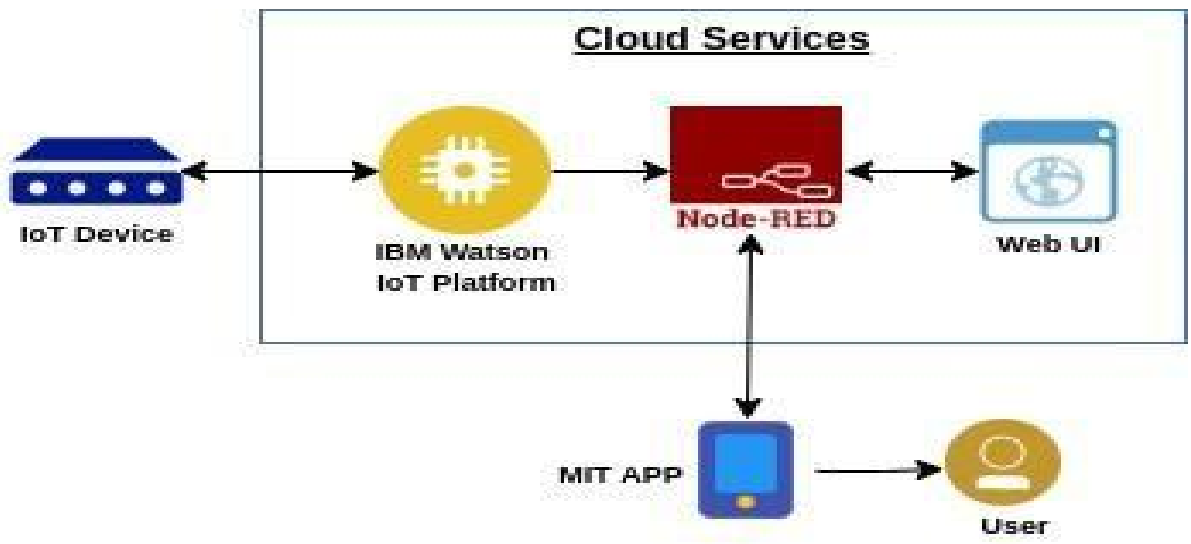
LIST OF ABBREVIATIONS

ABBREVATION	DESCRIPTION
IOT	Internet Of Things
ISP	Internet Service Provider
HTML	Hypertext Markup Language
CSS	Cascade Style Sheet

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Agriculture has always been the backbone of any economic development.To promote further growth of agriculture, it must be integrated with modern practices and technologies. With the wide spread acceptance of technology, it can be used in farming to make farmers perform their activity with ease. Electronics and IoT has found its application in many of the personal assistant devices. This can be extended to many vital fields like agriculture where their assistants can help solve many issues faced. Electronics can help devices get physically connected with their operational environment and analyze and collect data. IoT can help analyze and transfer the data to the user. The combination of these gives rise to an all-in-one device capable of carrying out a task.



1.2 PURPOSE

- In recent times, the erratic weather and climatic changes have caused issues for farmers in predicting the perfect conditions to initiate farming. Though on a superficial scale it seems unpredictable, it can be determinedwith certain parameters with which crop planning can be done.
- Maintenance of farm fields during and after cultivation are also important. These can be performed by measuring soil moisture, humidityand temperature.
- They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.
- IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, Temperature, humidity using some sensors. Farmers can monitor all the sensor parameters by using a web or mobile application even if the farmer is not near his field.
- Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.
- In large farmland, Internet of Things equipped drone helps to receive the currentstate of crops and send the live pictures of farmland.

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

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

There has been several attempts and solution to help farmers adopt technological practices. Few solutions restricted their performance with just suggestions and alerts. While few employed IoT independent electronics. Few of the cases of previous attempts and researches are described below.

i. "IoT based smart sensors agriculture stick for live temperature and moisture monitoring using Arduino, cloud computing & solar technology". This work was performed using Cloud computing platform (Things Speak) for data acquisition. The circuit was designed using Arduino and DHT 11 sensors.
ii. "Smart Farming using IoT, a solution for optimally monitoring farming conditions". This work used ESP-32 based IoT platform and Blynk mobile application.
iii. "Smart farming using IoT". The automation and interface part made use of water pump and HTTP protocol for parameters monitoring using website.

The above stated prior works lacked one or two features, which when included could have enhanced the performance. In the first work, including a Raspberry Pi based controller in place of Arduino can help reduce the design area while also providing microcontroller with additional UI and IoT interfaces. In the second stated work, going with MIT app inventor instead of Blynk application can improve the possibility of feature expansion. Farmers or developers won't need to go for a paid version of the app to include new features. In the third work, control of water pump can be enhanced with the use of servo-based water valves to direct and control the flow of water rather than using a bi-statedlogic.

2.2 REFERENCE

Reference.No	**Reference**
[1]	Implementation of Smart Farming using IoT Asian Journal of Applied Science and Technology (AJAST) Volume 5, Issue 2, Pages 58-67, April-June 2021
[2]	A control system in an intelligent farming by using arduino technology 2016 Fifth ICT International Student Project Conference (ICT-ISPC), p. 53 – 56, 2016
[3]	Muhammad Shoaib Farooq et al. [9] have done a Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming .
[4]	Abhiram MSD, Jyothsnavi Kuppili, N.Alivelu Manga Smart Farming System using IoT for Efficient Crop Growth
[5]	M. Gupta, M. Abdelsalam, S. Khorsandroo and S. Mittal, "Security and Privacy in Smart Farming: Challenges and Opportunities," in IEEE Access, vol. 8, pp. 34564-34584, 2020, doi: 10.1109/ACCESS.2020.2975142

[6]	A. Anitha, N. Sampath and M. A. Jerlin, "Smart Irrigation system using Internet of Things," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-7, doi: 10.1109/ic-ETITE47903.2020.271.
[7]	V. Puranik, Sharmila, A. Ranjan and A. Kumari, "Automation in Agriculture and IoT," 2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), 2019, pp. 1-6, doi: 10.1109/IoT-SIU.2019.8777619
[8]	S. R. Prathibha, A. Hongal and M. P. Jyothi, "IOT Based Monitoring System in Smart Agriculture," 2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT), 2017, pp. 81-84, doi: 10.1109/ICRAECT.2017.52.
[9]	Design and Implementation of a Smart Farm System (Article in Association of Arab Universities Journal of Engineering Sciences · January 2017)

2.3 PROBLEM STATEMENT DIFINITION

- Your customer's existing position should be fully described in a compelling customer problem statement. Take into account their emotions, the financial and emotional toll of their current circumstance, and any other pertinent information on their ideas or feelings.

- Creating a customer problem statement is easy with Miro. Using our collaborativeonline whiteboard, you can create an online problem statement that's easy to follow and shareable with your team. All you have to do is sign up for free, selectthis template, and follow your template.

- The problem statement in a nutshell covers all the possible technical aspects that can be included by farmer to convert farming in to smart and efficient farming.

- IoT enabled smart farming, on a wider perspective, concentrates on connecting all the independently operating sub-systems infarming automation into a single entity. IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture,temperature, and humidity using some sensors.

- The idea of IoT is further extended with the help of mobile and web application where farmers can monitor all the sensor parameters even if the farmer is not near his field. Watering the crop is one of the important tasks for the farmers. They can make the decision whether to water the crop or postpone it by monitoring the sensor parameters and controlling the motor pumps from the mobile application itself.

- The soil moisture sensor measures wetness content in the soil. The Arduino UNO microcontroller used to receive input from a various sensor and it can be controlled automatically.

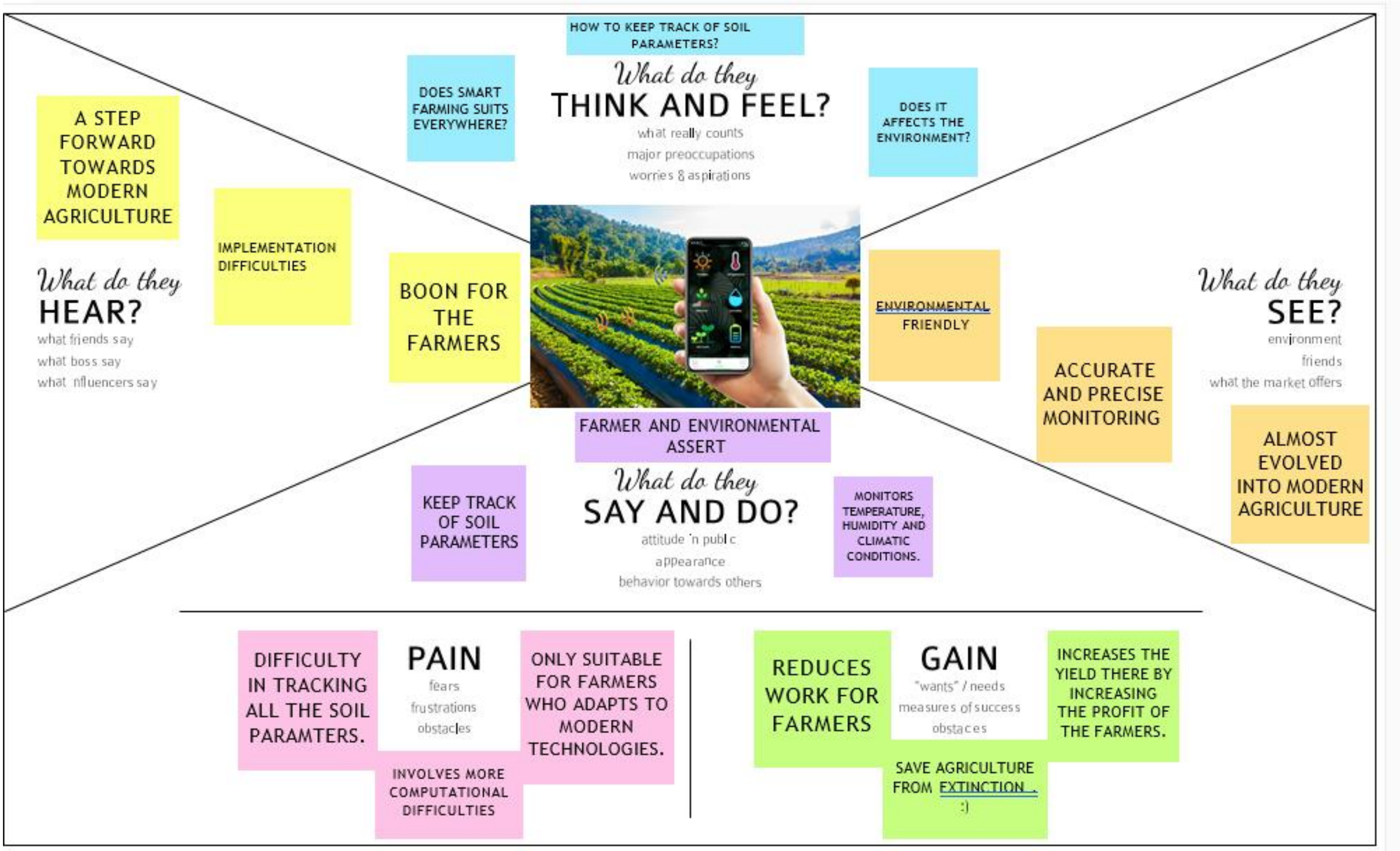
- When soil moisture sensor goes low the water pump will be on and it exceeds defined levels of the water motor will turn off automatically. We can constantly monitor the growth of a crop using ultrasonic sensor.

- PIR sensor detects the motion or unusual movement in the agricultural land. This device his very helpful to the former to monitor and control environmental parameters at their field. The farmers did not go to their field, they can remotely monitor and control using cloud.

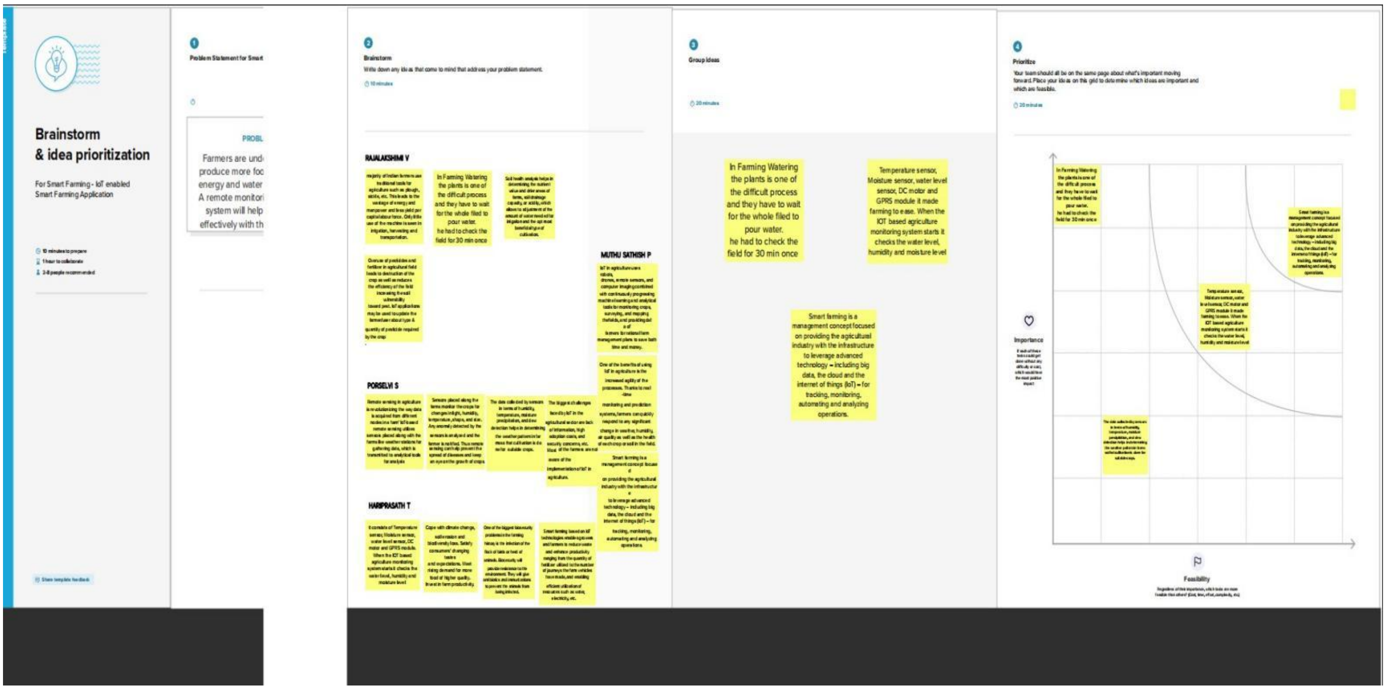
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem ant the person who is experiencing it The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 BRAINSTROMING



RAJALAKSHIMI V

- Majority of Indian farmers use traditional tools for agriculture such as plough, sickle, etc. This leads to the wastage of energy and manpower and less yield per capita labor force. Only little use of the machine is seen in irrigation, harvesting and transportation.
- In Farming Watering the plants is one of the difficult process and they have to wait for the whole fled to pour water. he had to check the feld for 30 min once.
- Soil health analysis helps in determining the nutrient value and drier areas of farms, soil drainage capacity, or acidity, which allows to adjustment of the amount of water needed for irrigation and the opt most beneficial type of cultivation.
- Overuse of pesticides and fertilizer in agricultural fields leads to destruction of the crop as well as reduces the efficiency of the field increasing the soil vulnerability toward pest. IoT applications may be used to update the farmer/user about type & quantity of pesticide required by the crop.

PORSELVI S

- Remote sensing in agriculture is revolutionizing the way data is acquired from different nodes in a farm' IoT-based remote sensing utilizes sensors placed along with the farms like weather stations for gathering data, which is transmitted to analytical tools for analysis.
- Sensors placed along the farms monitor the crops for changes in light, humidity, temperature, shape, and size. Any anomaly detected by the sensors is analyzed and the farmer is notified. Thus remote sensing can help prevent the spread of diseases and keep an eye on the growth of crops.
- The data collected by sensors in terms of humidity, temperature, moisture precipitation, and dew detection helps in determining the weather pattern in far so that cultivation is done for suitable crops.
- The biggest challenges faced by IoT in the agricultural sector are lack of information, high adoption costs, and security concerns, etc. Most of the farmers are not aware of the implementation of IoT in agriculture.

HARIPRASATH T

- It consists of Temperature sensor, Moisture sensor, water level sensor, DC motor and GPRS module. When the IOT based agriculture monitoring system starts it checks the water level, humidity and moisture level
- Cope with climate change, soil erosion and biodiversity loss. Satisfy consumers' changing tastes and expectations. Meet rising demand for more food of higher quality. Invest in farm productivity.
- One of the biggest biosecurity problems in the farming history is the infection of the flock of birds or herd of animals. Biosecurity will provide resistance to the environment. They will give antibiotics and immunizations to prevent the animals from being infected.
- Smart farming based on IoT technologies enables growers and farmers to reduce waste and enhance productivity ranging from the quantity of fertilizer utilized to the

number of journeys the farm vehicles have made, and enabling efficient utilization of resources such as water, electricity, etc.

MUTHU SATHISH P

- IoT in agriculture uses robots, drones, remote sensors, and computer imaging combined with continuously progressing machine learning and analytical tools for monitoring crops, surveying, and mapping the fields, and providing data of farmers for rational farm management plans to save both time and money.
- One of the benefits of using IoT in agriculture is the increased agility of the processes. Thanks to real -time monitoring and prediction systems, farmers can quickly respond to any significant change in weather, humidity, air quality as well as the health of each crop or soil in the field.
- Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology – including big data, the cloud and the internet of things (IoT) – for tracking, monitoring, automating and analyzing operations.

3.3 Proposed Solution

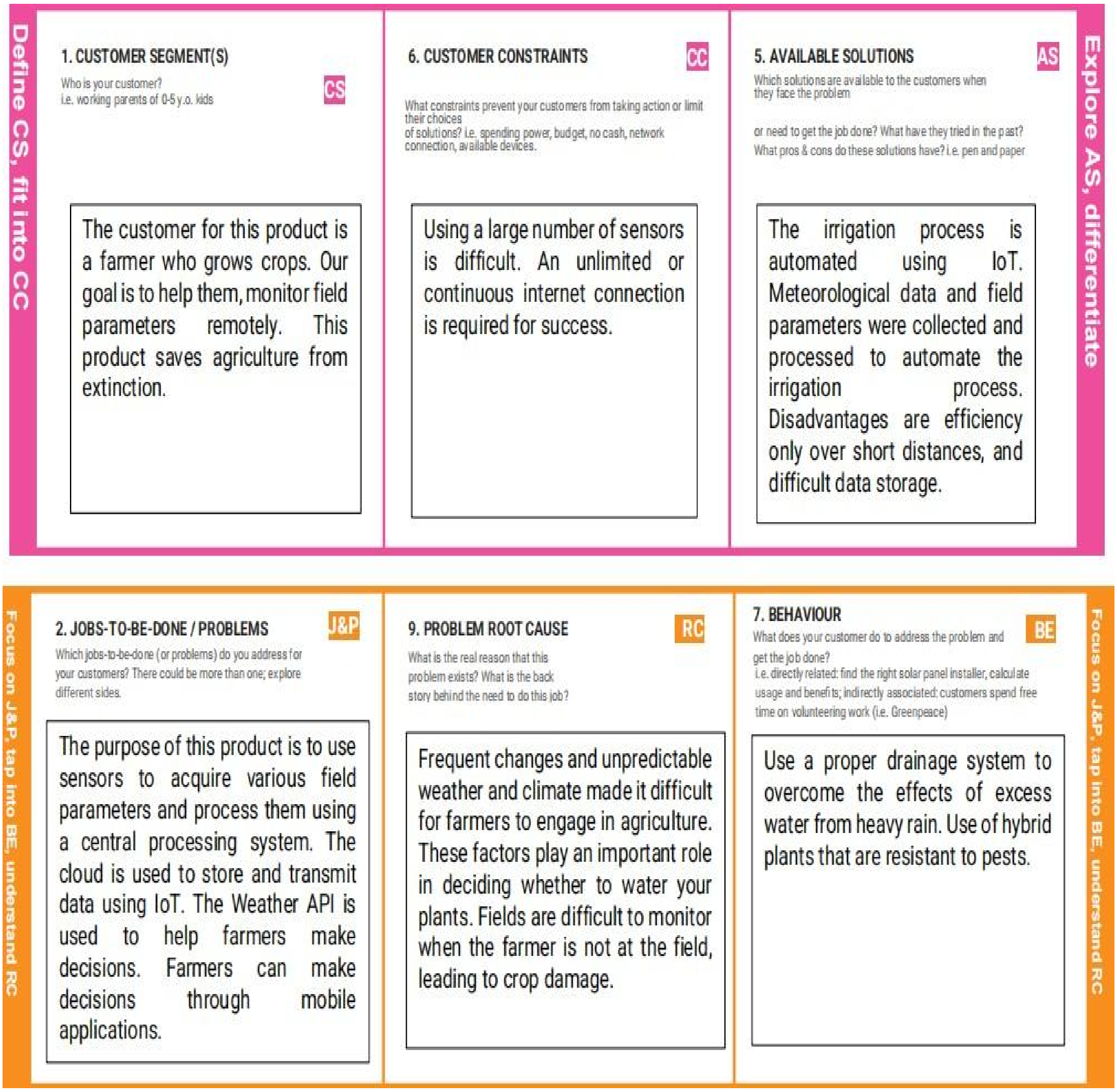
Proposed Solution Template:

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

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	In agriculture, there are two major problems one is unpredictable climate change and another one is the yields of the crops that havebeen damaged by improper irrigation. Our project will give the solution to overcome theseproblems with help of IOT.
2.	Idea / Solution description	It collects the data from different types of sensors and it sends the value to the main server. It also collects the weather data fromthe weather API. The ultimate decision, whether to water the crop or not is taken by the farmer using mobile application.
3.	Novelty / Uniqueness	It depends on IOT thus eliminating the need of physical work of farmers and thus increasing the productivity in every possible manner. Theweather data are taken from the reliable source.
4.	Social Impact / Customer Satisfaction	The informations collected are from reliable sources and hence the farmer could make moreprecise decision, thereby the productivity increases.
5.	Business Model (Revenue Model)	Smart farming is an advanced and innovative way to get maximum cultivation and minimizethe human efforts.
6.	Scalability of the Solution	Automatic farming equipment adjustment is made feasible by integrating information such as crops/weather and equipment to automatically alter temperature, humidity, andso on. With the use of sensors, it has enabled farmers to reduce waste and increase output.

S.No.	Parameter	DESCRIPTION
7.	Problem Statement (Problem to besolved)	To incorporate the process of working andalso elevate the smart farming using IOT enabled smart Farming technique since the traditional Farming technique I very Complex one.
8.	Idea / Solution description	To automate irrigation in accordance to theamount of moisture present in soil
9.	Novelty / Uniqueness	Automation of irrigation to amount of moisture
10.	Social Impact / Customer Satisfaction	The problems faced by the farmers in the process of irrigation gets solved and this full fillsand saves their crops from over irrigation
11.	Business Model (Revenue Model)	The process of fulfilling this process brings revolution in drip irrigation systems also makesa revolutionary change in market
12.	Scalability of the Solution	The design scale of solution has been plannedin a compact manner

3.4 Problem Solution fit



3. TRIGGERS

TR

What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.

Farmers struggle to provide adequate irrigation. Inadequate water supply reduces yields and affects farmers' profit levels. Farmers have a hard time predicting the weather.

4. EMOTIONS: BEFORE / AFTER

EM

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.

BEFORE: Lack of knowledge in weather forecasting →Random decisions →low yield.
AFTER: Data from reliable source → correct decision →high yield

10. YOUR SOLUTION

SL

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.

Our product collects data from various types of sensors and sends the values to our main server. It also collects weather data from the Weather API. The final decision to irrigate the crop is made by the farmer using a mobile application.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

ONLINE: Providing online assistance to the farmer, in providing knowledge regarding the pH and moisture level of the soil. Online assistance to be provided to the user in using the product

OFFLINE: Awareness camps to be organized to teach the importance and advantages of the automation and IoT in the development of agriculture.

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

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	Sensor Function for framing System	Measure the Temperature and Humidity Measure the Soil Monitoring Check the crop diseases
FR-4	Manage Modules	Manage Roles of User Manage User permission
FR-5	Check whether details	Temperature details and Humidity details
FR-6	Data Management	Manage the data of weather conditions Manage the data of crop conditions Manage the data of live stock conditions

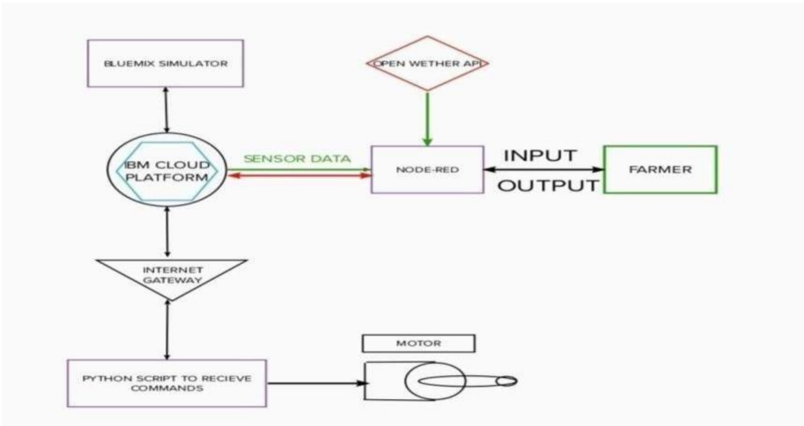
4.2 NON-FUNCTIONAL REQUIREMENT

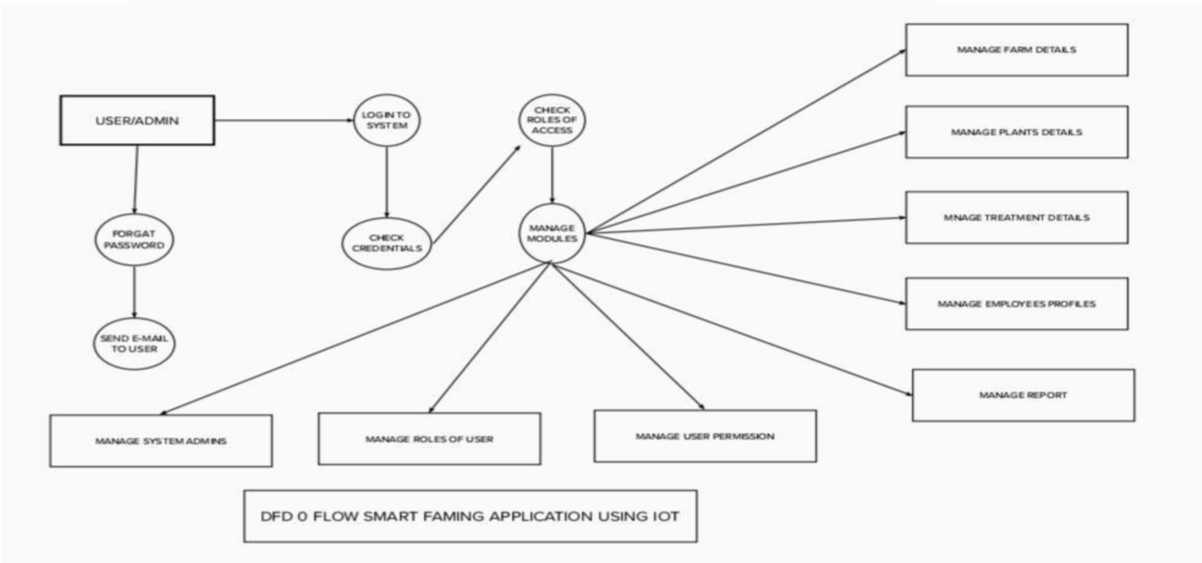
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Usability includes easy learn ability, efficiency in use, remember ability, lack of errors in operation and subjective pleasure.
NFR-2	**Security**	All the details about the user are protected from unauthorized access. Detection and identification of any misfunctions of sensors.
NFR-3	**Reliability**	The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages.
NFR-4	**Performance**	The use of modern technology solutions helps to achieve the maximum performances thus resulting in better quality and quantity yields.
NFR-5	**Availability**	With permitted network connectivity the application is accessible
NFR-6	**Scalability**	Scalability refers to the ability to increase available resources and system capability without the need to go through a major system redesign or implementation

5. PROJECT DESIGN

5.1 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.

1. The different soil parameters temperature, soil moistures and then humidity are sense during different sensors and obtained value is stored in the IBMcloud.

2. Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weatherAPI.

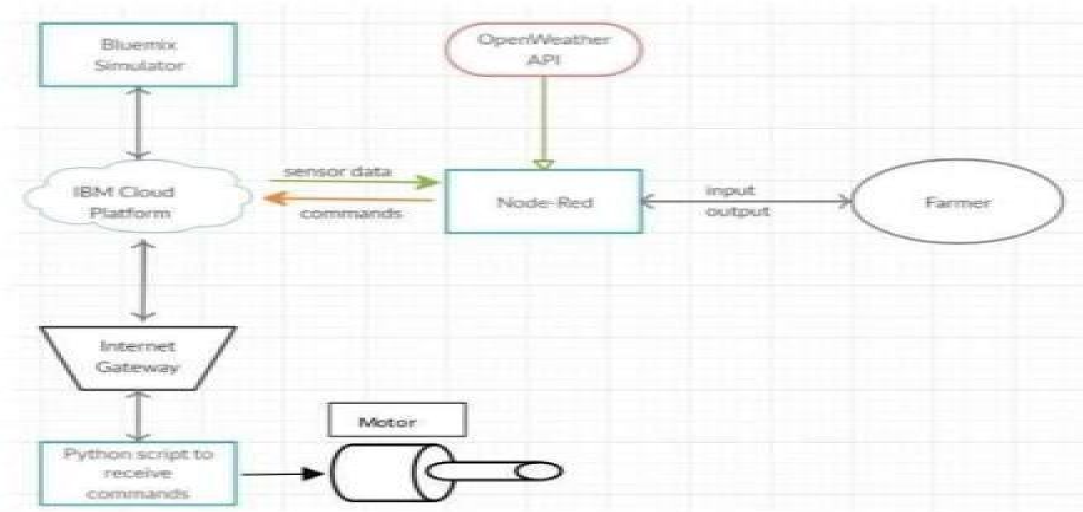
3. NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed forth communication.

4. All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could plan through an app, weather to water the crop or not depending upon the sensor values. By using the app they can remotely operate to the motor switch.

5.2 Solution & Technical Architecture

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

1. The different soil parameters temperature, soil moistures and then humidity are sensed using different sensors and obtained value is stored in the IBM cloud.
2. Arduino UNO is used as a processing Unit that process the data obtained from the sensors and whether data from the weather API.
3. NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.
4. All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could decide through an app, weather to water the crop or not depending upon the sensor values. By using the app, they can remotely operate the motor switch.



<u>Guidelines</u>:

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local/ Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components /services 5. Indicate interface to machine learning models (if applicable)

- The different soil parameters temperature, soil moistures and humidity are sensed using different sensors and obtained value is stored in the IBM cloud.
- Here, instead of using Raspberry Pi processor unit, random values are generated for various soil parameters using Python.
- NODE-RED is used as a programming tool to write the hardware, software, and APIs. The MQTT protocol is followed for the communication.
- All the collected data are provided to the user through a mobile application that was developed using the MIT app inventor. The user could decide through an app, weather to water the crop or not depending upon the sensor values. By using the app, they can remotely operate the motor switch.

Table-1:

Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts withapplication e.g. Web UI, Mobile App,Chabot etc.	MIT app
2.	Application Logic-1	Logic for a process in theapplication	Node red/IBM Watson/MIT app
3.	Application Logic-2	Logic for a process in theapplication	Node red/IBM Watson/MIT app
4.	Application Logic-3	Logic for a process in theapplication	Node red/IBM Watson/MIT app
5.	Database	Data Type, Configurationsetc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM cloud.
7.	Temperature sensor	Monitors the temperature ofthe crop	
8.	Humidity sensor	Monitors the humidity	
9.	Soil moisture sensor (Torsiometer's)	Monitors the soiltemperature	

10.	Weather sensor	Monitors the weather	.
11.	Solar panel		.
12.	RTC module	Date and time configuration	
13.	Relay	To get the soil moisturedata	

Table-2:

Application Characteristics:

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

5.3 USER STORIES

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

			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 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
Customer(webuser)						
administrator						
CustomerCare Executive						

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

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

6.2 Sprint Delivery Schedule

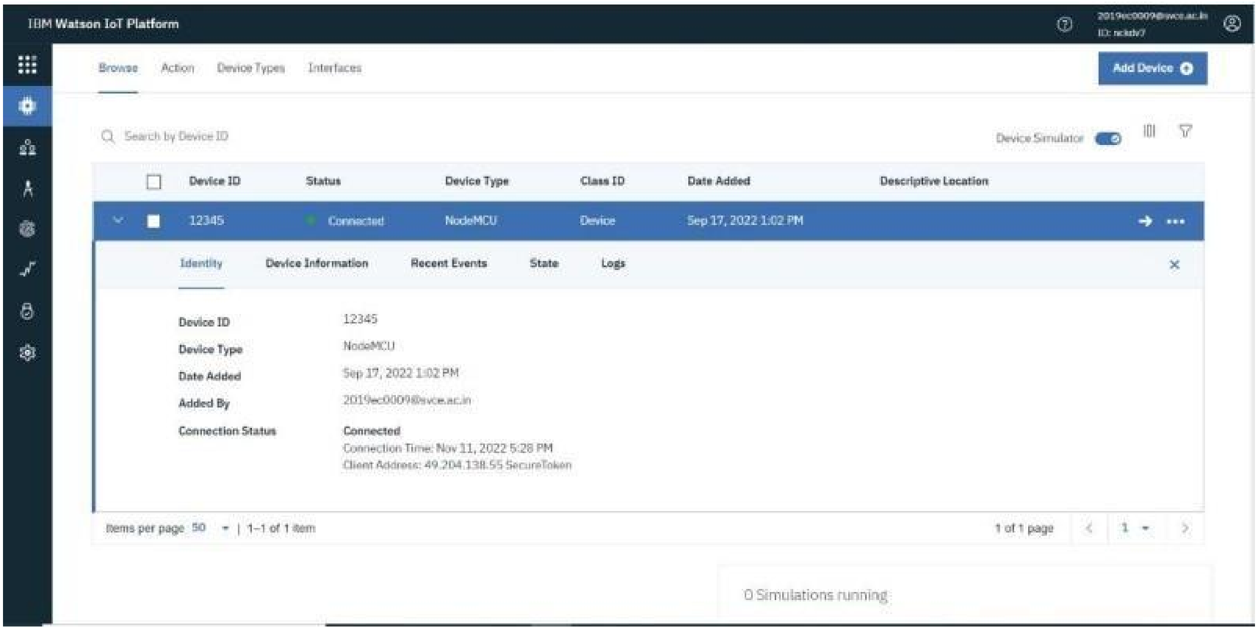
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	18 Oct 2022	24 Oct 2022	20	24 Oct 2022
Sprint-2	20	6 Days	25Oct 2022	30 Oct 2022	20	30 Oct 2022
Sprint-3	20	6 Days	31 Oct 2022	5 Nov 2022	20	5 Nov 2022
Sprint-4	20	6 Days	6 Nov 2022	10 Nov 2022	20	10 Nov 2022

7. CODING & SOLUTION

- **Configuration of the IBM Watson IOT Platform and a device:**

In the IBM Watson IOT Platform, under the catalog list, under the Internet of Things platform, a device has been created. From that the device Credentials such as device ID, Device Type, Organization ID.

Authentication code were obtained



- **Development of Python Script to publish data to IBM Watson IOTplatform:**

<u>Code</u>:

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

#Provide your IBM Watson Device Credentials
organization = "nckdv7" deviceType =
"NodeMCU"
deviceId = "12345" authMethod = "token" authToken = "12345678" # Initialize GPIO 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: #Get Sensor Data from DHT11 temp=random.randint(0,100) pulse=random.randint(0,100) moisture= random.randint(0,100) humidity=random.randint(0,100); lat = 17 lon = 18 data = { 'temperature' : temp, 'humidity' : humidity, 'Moisture' :

moisture} #print

data

def myOnPublishCallback():

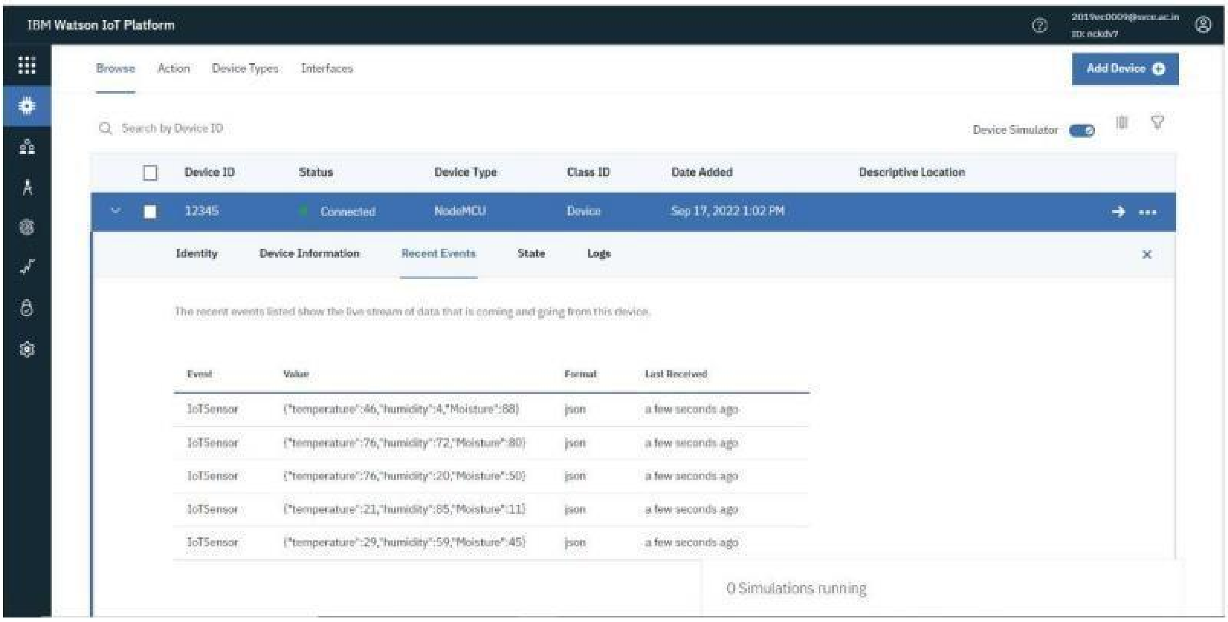
print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % humidity, "Soil Moisture = %s %%" % moisture,"to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback) if not success: print("Not connected to IoTF") time.sleep(1) deviceCli.commandCallback = myCommandCallback # Disconnect the device and application from the cloud deviceCli.disconnect()

Python Code Output:

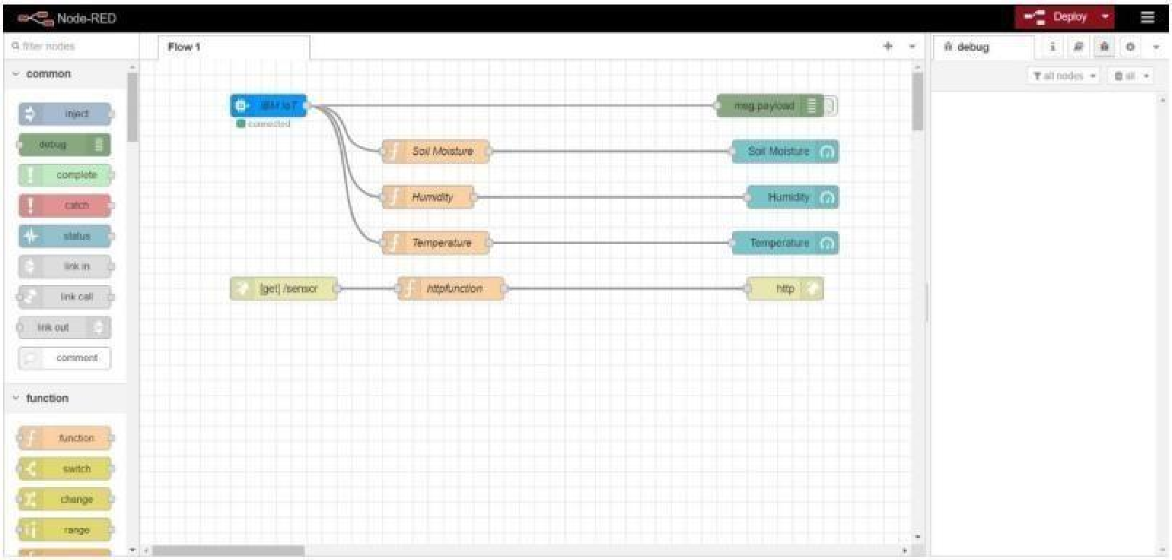
```
*Python 3.7.0 Shell*
File  Edit  Shell  Debug  Options  Window  Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\manoj-pt5890\Documents\python\project.py =========
2022-11-11 17:28:32,248   ibmiotf.device.Client      INFO    Connected successfully: d:nckdv7:NodeMCU:12345
Published Temperature = 89 C Humidity = 70 % Soil Moisture = 3 % to IBM Watson
Published Temperature = 78 C Humidity = 5 % Soil Moisture = 2 % to IBM Watson
Published Temperature = 85 C Humidity = 61 % Soil Moisture = 34 % to IBM Watson
Published Temperature = 75 C Humidity = 83 % Soil Moisture = 23 % to IBM Watson
Published Temperature = 72 C Humidity = 34 % Soil Moisture = 70 % to IBM Watson
Published Temperature = 38 C Humidity = 36 % Soil Moisture = 48 % to IBM Watson
Published Temperature = 62 C Humidity = 36 % Soil Moisture = 35 % to IBM Watson
Published Temperature = 34 C Humidity = 64 % Soil Moisture = 29 % to IBM Watson
Published Temperature = 95 C Humidity = 40 % Soil Moisture = 100 % to IBM Watson
Published Temperature = 47 C Humidity = 95 % Soil Moisture = 58 % to IBM Watson
```

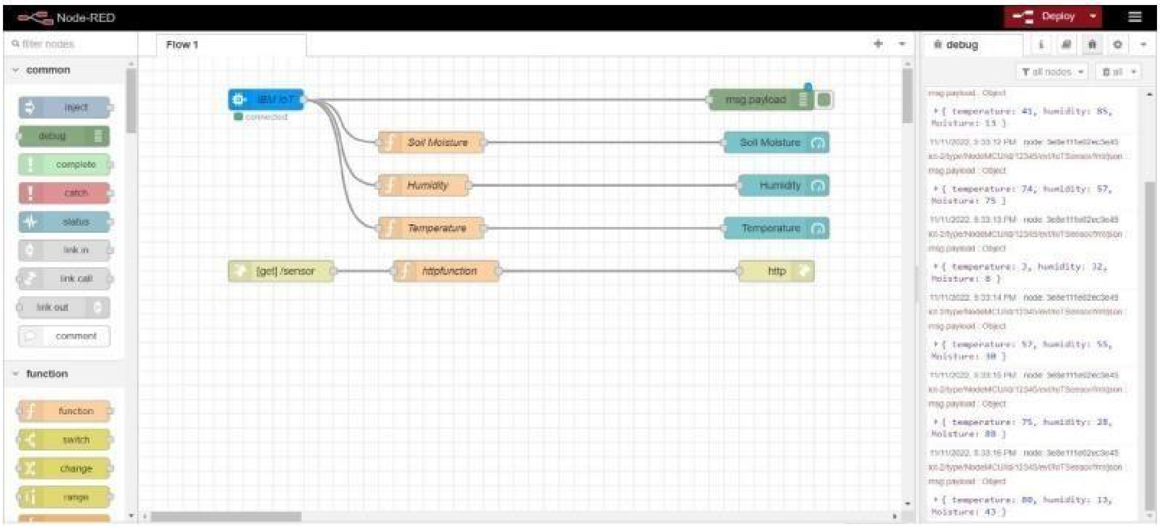
IBM Cloud after publishing data:



- **Creation of Node Red Service for device events:**

  In the IBM Watson IOT platform, under the catalog, under the Node Red app service, an application is deployed using cloud foundry. In the cloud foundry, a group has been created and using the ci pipeline, the app url is obtained. Using the URL, the Node red is launched. The IBM Watson IOT platform is connected to Node red using the IBM IoT palette. Using appropriate palettes, the data published in the IBM IoT platform is printed in the debug window of Node red.





Code block for the function palette:

1) <u>Soil moisture</u>:

Soil = msg.payload.Moisture
msg.payload = "Soil Moisture :
" global.set('m',Soil)
msg.payload = Math.round(Soil)
return msg;

2) <u>Humidity:</u>

Humidity = msg.payload.humidity

msg.payload = "Humidity : "

global.set('h',Humidity) msg.payload

= Math.round(Humidity) return msg;

3) **Temperature:**

Temperature = msg.payload.temperature

msg.payload = "Temperature : "

global.set('t',Temperature) msg.payload

=Math.round(Temperature) return msg;

4) **HTTP Function:**

msg.payload = {"Temperature:": global.get('t'),"Humidity:": global.get('h'),"Soil Moisture:": global.get('m')} return msg;

- **Creation of Website dashboard:**

A website dashboard has been created using the gauge palette. It can be accessed by adding "/ui" in the main url of Node red. This dashboard displays the gauge representation of the data published in the IBM IOT platform.

Python code used:

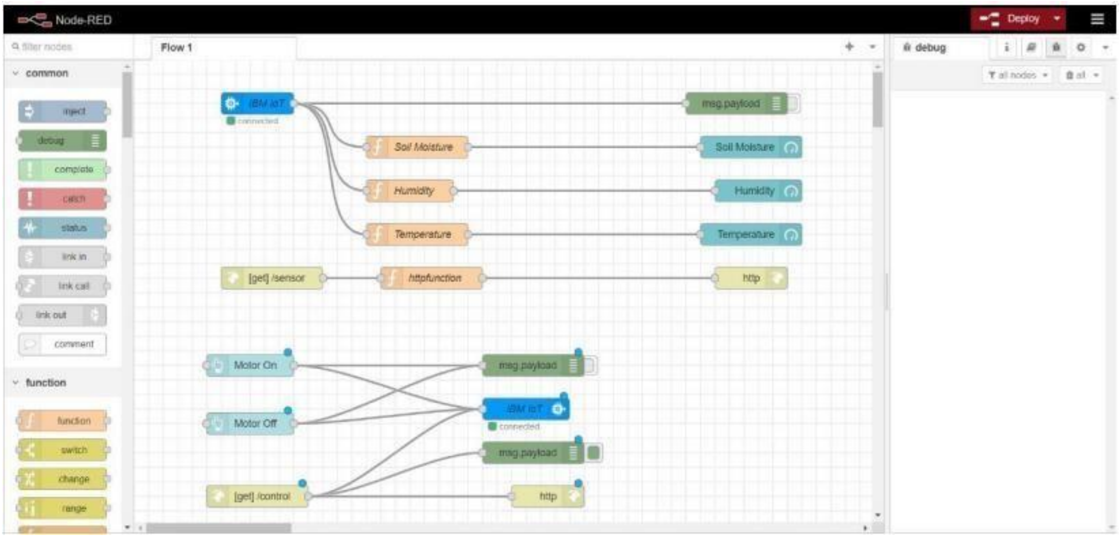
```
import time import sys
import
ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device
Credentials organization = "nckdv7"
deviceType = "NodeMCU" deviceId =
"12345"  authMethod = "token"  authToken =
"12345678" # Initialize GPIO  try:
        deviceOptions = {"org": 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:     #Get     Sensor     Data     from     DHT11
temp=random.randint(0,100)            pulse=random.randint(0,100)
moisture=                                    random.randint(0,100)
humidity=random.randint(0,100);  lat  =  17  lon  =  18  data  =
{ 'temperature' : temp, 'humidity' :
        humidity, 'Moisture' : moisture}
```

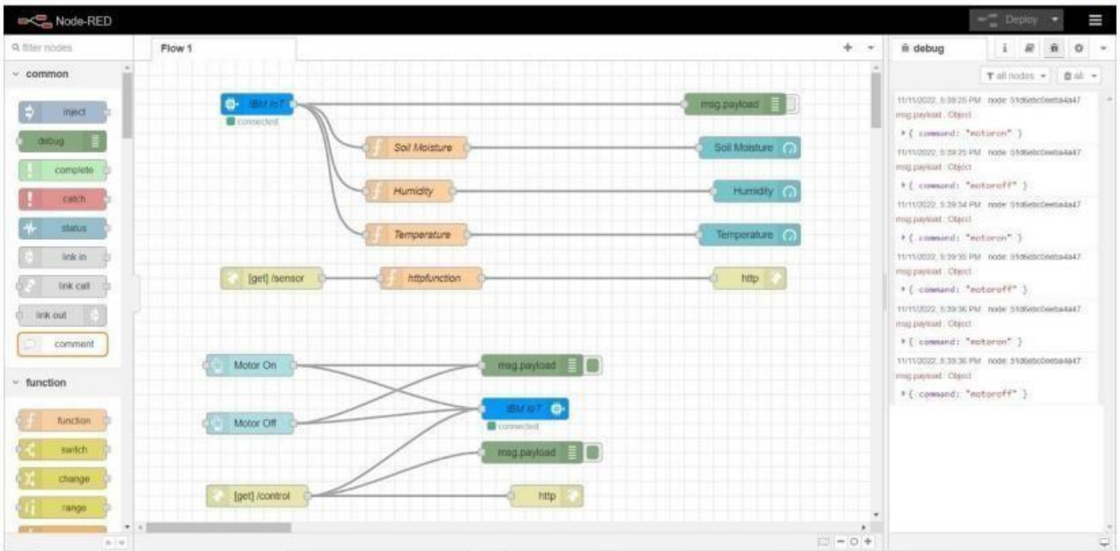
#print data def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % humidity, "Soil Moisture = %s %%" % moisture,"to IBM Watson")
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback) if not success:
print("Not connected to IoTF")
time.sleep(1) deviceCli.commandCallback = myCommandCallback # Disconnect the device and application

from the cloud deviceCli.disconnect()

Creation of Node Red service for device commands:

In addition to the palettes used in the Sprint-2, additional palettes such as buttons have been included to control devices by giving commands and the output is printed in the debug whenever a specific command is given.





Development of Python script to subscribe command from the IBM IOTplatform:

Code:

```
import time import sys
import
ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials organization =
"nckdv7" deviceType = "NodeMCU" deviceId = "12345"
authMethod = "token" authToken = "12345678" # Initialize
GPIO def myCommandCallback(cmd): print("Command
received: %s" % cmd.data['command'])
status=cmd.data['command']
        if status=="motoron":
                print("Motor is ON")  else:
                print("Motor is OFF")
         #print(cmd)  try:  deviceOptions  =  {"org":  organization,  "type":
deviceType,  "id":  deviceId,  "authmethod":  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:
        #Get   Sensor   Data   from   DHT11   temp=random.randint(0,100)
        pulse=random.randint(0,100)    moisture=    random.randint(0,100)
        humidity=random.randint(0,100);   lat   =   17   lon   =   18   data   =
        { 'temperature' : temp, 'humidity' : humidity, 'Moisture' :
```

```
moisture}
#print data

def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s
%%" % humidity, "Soil Moisture = %s %%" % moisture,"to IBM
Watson")  success = deviceCli.publishEvent("IoTSensor",
"json", data, qos=0,   on_publish=myOnPublishCallback)  if not
success:
print("Not connected to IoTF")
time.sleep(1)                deviceCli.commandCallback       =
myCommandCallback # Disconnect the device and
application from the cloud deviceCli.disconnect()
```

Output:

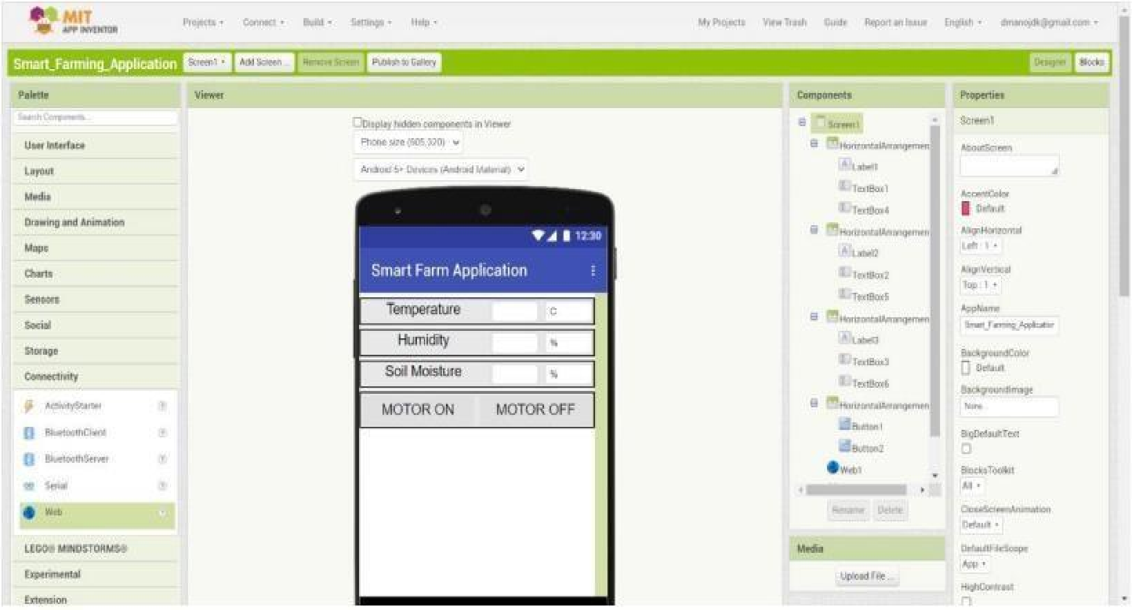
```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Published Temperature = 88 C Humidity = 66 % Soil Moisture = 3 % to IBM Watson
Published Temperature = 50 C Humidity = 97 % Soil Moisture = 63 % to IBM Watson
Published Temperature = 24 C Humidity = 33 % Soil Moisture = 50 % to IBM Watson
Published Temperature = 73 C Humidity = 29 % Soil Moisture = 56 % to IBM Watson
Published Temperature = 23 C Humidity = 1 % Soil Moisture = 90 % to IBM Watson
Published Temperature = 31 C Humidity = 12 % Soil Moisture = 38 % to IBM Watson
Published Temperature = 91 C Humidity = 62 % Soil Moisture = 58 % to IBM Watson
Published Temperature = 15 C Humidity = 49 % Soil Moisture = 70 % to IBM Watson
Published Temperature = 51 C Humidity = 81 % Soil Moisture = 84 % to IBM Watson
Published Temperature = 61 C Humidity = 17 % Soil Moisture = 37 % to IBM Watson
Published Temperature = 91 C Humidity = 87 % Soil Moisture = 70 % to IBM Watson
Published Temperature = 35 C Humidity = 6 % Soil Moisture = 95 % to IBM Watson
Published Temperature = 52 C Humidity = 41 % Soil Moisture = 63 % to IBM Watson
Published Temperature = 40 C Humidity = 51 % Soil Moisture = 86 % to IBM Watson
Published Temperature = 33 C Humidity = 21 % Soil Moisture = 38 % to IBM Watson
Published Temperature = 29 C Humidity = 48 % Soil Moisture = 22 % to IBM Watson
Published Temperature = 45 C Humidity = 32 % Soil Moisture = 23 % to IBM Watson
Published Temperature = 98 C Humidity = 38 % Soil Moisture = 8 % to IBM Watson
Published Temperature = 44 C Humidity = 71 % Soil Moisture = 16 % to IBM Watson
Command received: motoron
Motor is ON
Published Temperature = 62 C Humidity = 2 % Soil Moisture = 34 % to IBM Watson
Published Temperature = 21 C Humidity = 14 % Soil Moisture = 82 % to IBM Watson
Published Temperature = 35 C Humidity = 2 % Soil Moisture = 5 % to IBM Watson
Published Temperature = 34 C Humidity = 78 % Soil Moisture = 44 % to IBM Watson
Command received: motoroff
Motor is OFF

Published Temperature = 93 C Humidity = 81 % Soil Moisture = 87 % to IBM Watson
Command received: motoron
Motor is ON
Command received: motoroff
Motor is OFF
Published Temperature = 54 C Humidity = 36 % Soil Moisture = 81 % to IBM Watson
Published Temperature = 56 C Humidity = 76 % Soil Moisture = 56 % to IBM Watson
Published Temperature = 70 C Humidity = 53 % Soil Moisture = 74 % to IBM Watson
Published Temperature = 58 C Humidity = 22 % Soil Moisture = 68 % to IBM Watson
Command received: motoron
Motor is ON
Published Temperature = 93 C Humidity = 34 % Soil Moisture = 11 % to IBM Watson
Command received: motoroff
Motor is OFF
Published Temperature = 86 C Humidity = 67 % Soil Moisture = 38 % to IBM Watson
Published Temperature = 49 C Humidity = 70 % Soil Moisture = 61 % to IBM Watson
Published Temperature = 94 C Humidity = 48 % Soil Moisture = 77 % to IBM Watson
Published Temperature = 59 C Humidity = 6 % Soil Moisture = 11 % to IBM Watson
Published Temperature = 16 C Humidity = 6 % Soil Moisture = 41 % to IBM Watson
```

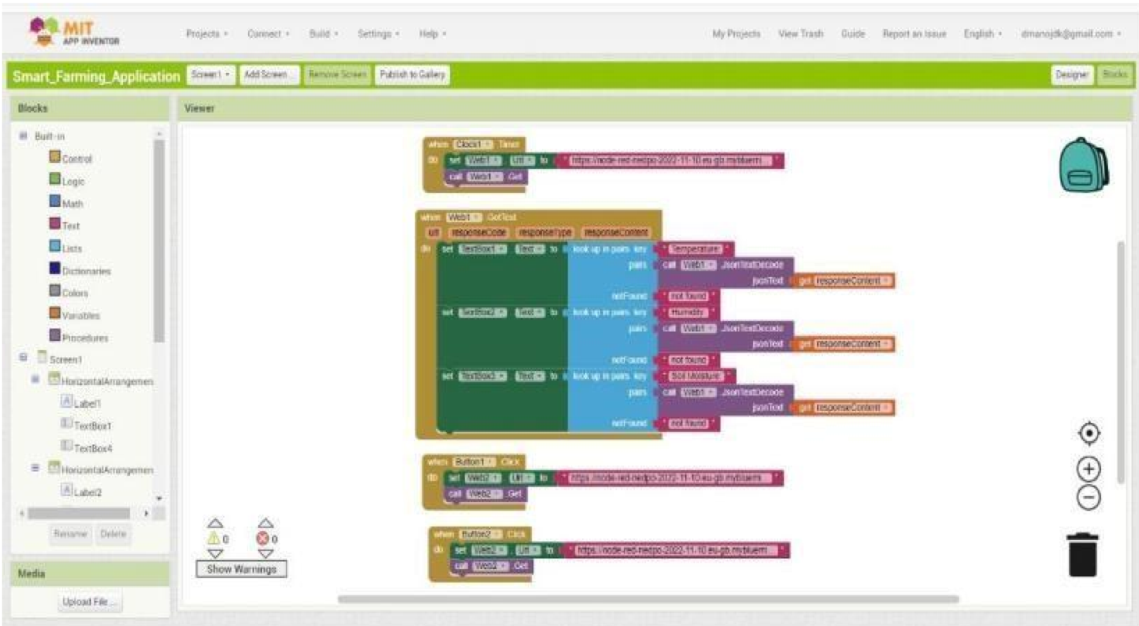
- **Development of Mobile application using MIT App Inventor:**

  In the MIT App Inventor platform, an application is created which monitors the farmland parameters such as temperature, humidity, soil moisture and controls the actuators such as motors.

MIT App Front End:



Backend:



App working:

The app works based on HTTP protocol. The app uses HTTP GET method to parse the JSON data from the Node red website and displays the value in the UI. Using the HTTP POST method, the app sends command when a specific button is pressed. From where, the python code subscribes the command data from the cloud thereby notifying the command is received.

Python code:

```
 import timeimport sys
import
ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials organization =
"nckdv7" deviceType = "NodeMCU" deviceId = "12345"
authMethod = "token" authToken = "12345678" # Initialize
GPIO def myCommandCallback(cmd): print("Command
received: %s" % cmd.data['command'])
status=cmd.data['command'] if status=="motoron":
print("Motor is ON") else:
        print("Motor is OFF")
        #print(cmd)

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:
    #Get Sensor Data from DHT11 temp=random.randint(0,100)
    pulse=random.randint(0,100) moisture= random.randint(0,100)
    humidity=random.randint(0,100); lat = 17 lon = 18 data =
    { 'temperature' : temp, 'humidity' : humidity, 'Moisture' :
```

```
moisture}
#print data def
myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity = %s
%%" % humidity, "Soil Moisture = %s %%" % moisture,"to IBMWatson")
success = deviceCli.publishEvent("IoTSensor",
"json", data, qos=0, on_publish=myOnPublishCallback) if not
success:
print("Not connected to IoTF") time.sleep(1)
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

Output:

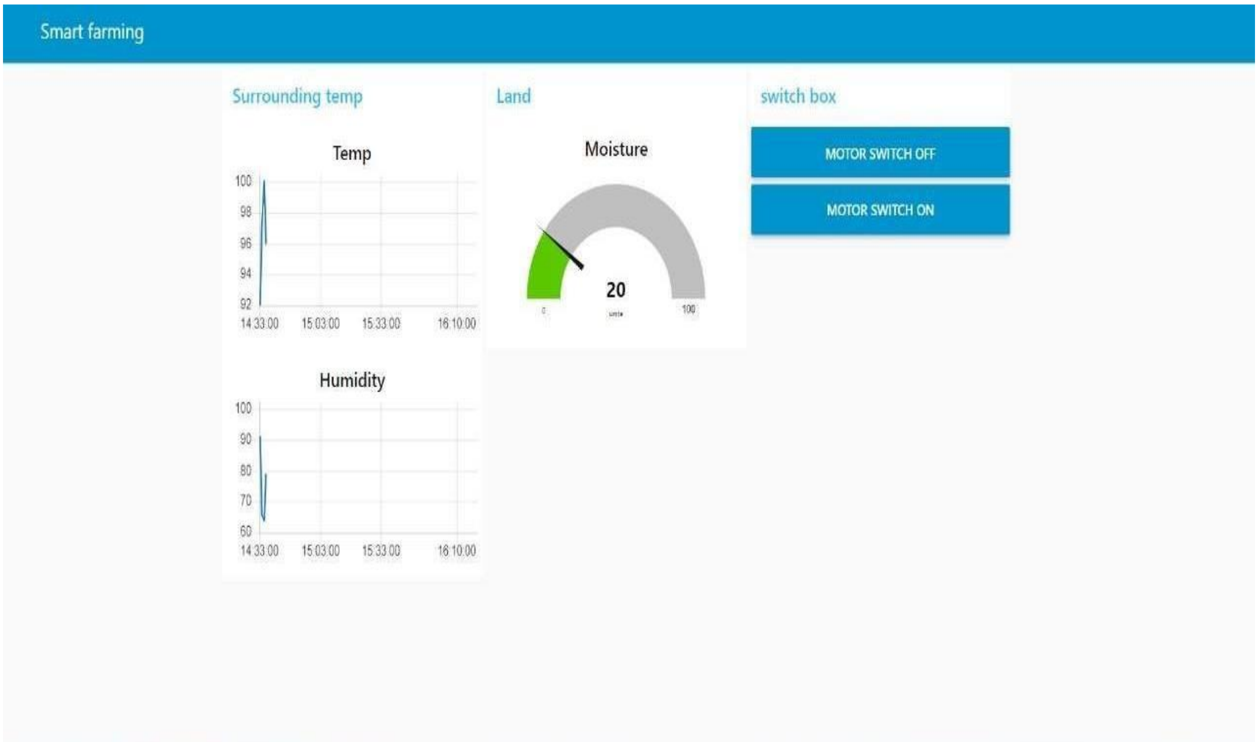
Python 3.7.0 Shell

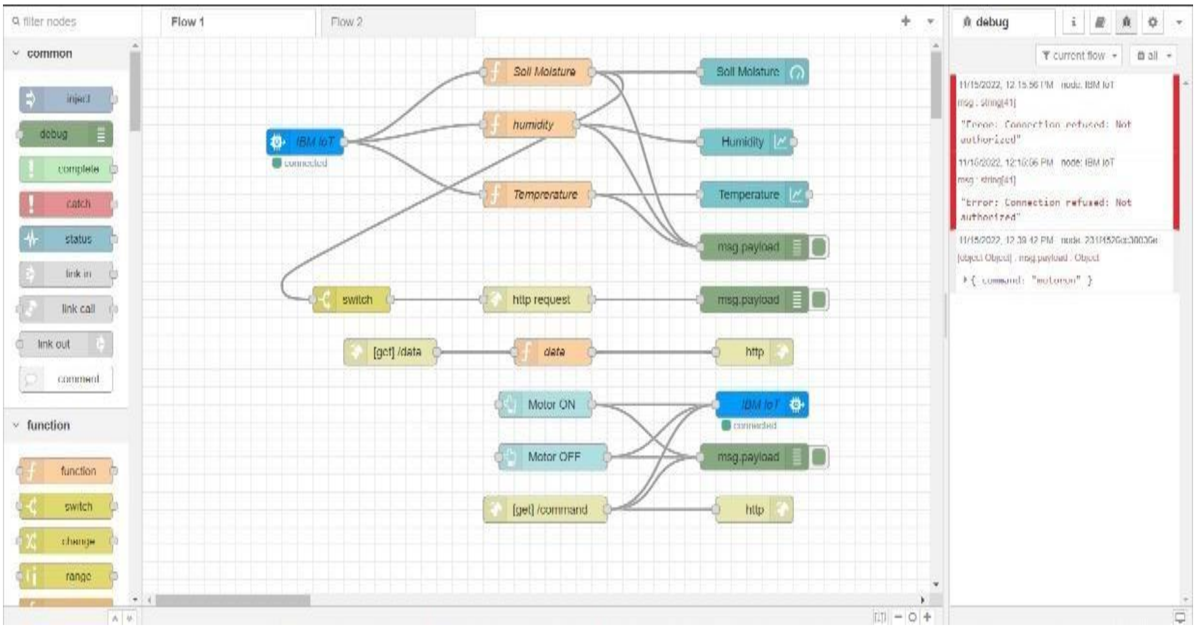
File Edit Shell Debug Options Window Help

```
Published Temperature = 88 C Humidity = 66 % Soil Moisture = 3 % to IBM Watson
Published Temperature = 50 C Humidity = 97 % Soil Moisture = 63 % to IBM Watson
Published Temperature = 24 C Humidity = 33 % Soil Moisture = 50 % to IBM Watson
Published Temperature = 73 C Humidity = 29 % Soil Moisture = 56 % to IBM Watson
Published Temperature = 23 C Humidity = 1 % Soil Moisture = 90 % to IBM Watson
Published Temperature = 31 C Humidity = 12 % Soil Moisture = 38 % to IBM Watson
Published Temperature = 91 C Humidity = 62 % Soil Moisture = 58 % to IBM Watson
Published Temperature = 15 C Humidity = 49 % Soil Moisture = 70 % to IBM Watson
Published Temperature = 51 C Humidity = 81 % Soil Moisture = 84 % to IBM Watson
Published Temperature = 61 C Humidity = 17 % Soil Moisture = 37 % to IBM Watson
Published Temperature = 91 C Humidity = 87 % Soil Moisture = 70 % to IBM Watson
Published Temperature = 35 C Humidity = 6 % Soil Moisture = 95 % to IBM Watson
Published Temperature = 52 C Humidity = 41 % Soil Moisture = 63 % to IBM Watson
Published Temperature = 40 C Humidity = 51 % Soil Moisture = 86 % to IBM Watson
Published Temperature = 33 C Humidity = 21 % Soil Moisture = 38 % to IBM Watson
Published Temperature = 29 C Humidity = 48 % Soil Moisture = 22 % to IBM Watson
Published Temperature = 45 C Humidity = 32 % Soil Moisture = 23 % to IBM Watson
Published Temperature = 98 C Humidity = 38 % Soil Moisture = 8 % to IBM Watson
Published Temperature = 44 C Humidity = 71 % Soil Moisture = 16 % to IBM Watson
Command received: motoron
Motor is ON
Published Temperature = 62 C Humidity = 2 % Soil Moisture = 34 % to IBM Watson
Published Temperature = 21 C Humidity = 14 % Soil Moisture = 82 % to IBM Watson
Published Temperature = 35 C Humidity = 2 % Soil Moisture = 5 % to IBM Watson
Published Temperature = 34 C Humidity = 78 % Soil Moisture = 44 % to IBM Watson
Command received: motoroff
Motor is OFF
Published Temperature = 93 C Humidity = 81 % Soil Moisture = 87 % to IBM Watson
Command received: motoron
Motor is ON
Command received: motoroff
Motor is OFF
Published Temperature = 54 C Humidity = 36 % Soil Moisture = 81 % to IBM Watson
Published Temperature = 56 C Humidity = 76 % Soil Moisture = 56 % to IBM Watson
Published Temperature = 70 C Humidity = 53 % Soil Moisture = 74 % to IBM Watson
Published Temperature = 58 C Humidity = 22 % Soil Moisture = 68 % to IBM Watson
Command received: motoron
Motor is ON
Published Temperature = 93 C Humidity = 34 % Soil Moisture = 11 % to IBM Watson
Command received: motoroff
Motor is OFF
Published Temperature = 86 C Humidity = 67 % Soil Moisture = 38 % to IBM Watson
Published Temperature = 49 C Humidity = 70 % Soil Moisture = 61 % to IBM Watson
Published Temperature = 94 C Humidity = 48 % Soil Moisture = 77 % to IBM Watson
Published Temperature = 59 C Humidity = 6 % Soil Moisture = 11 % to IBM Watson
Published Temperature = 16 C Humidity = 6 % Soil Moisture = 41 % to IBM Watson
```

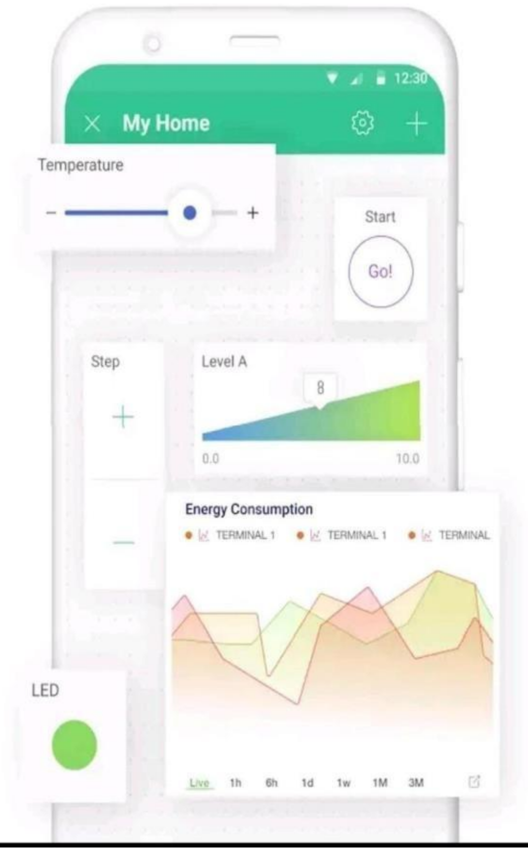
8.TESTING

8.1 Test Cases



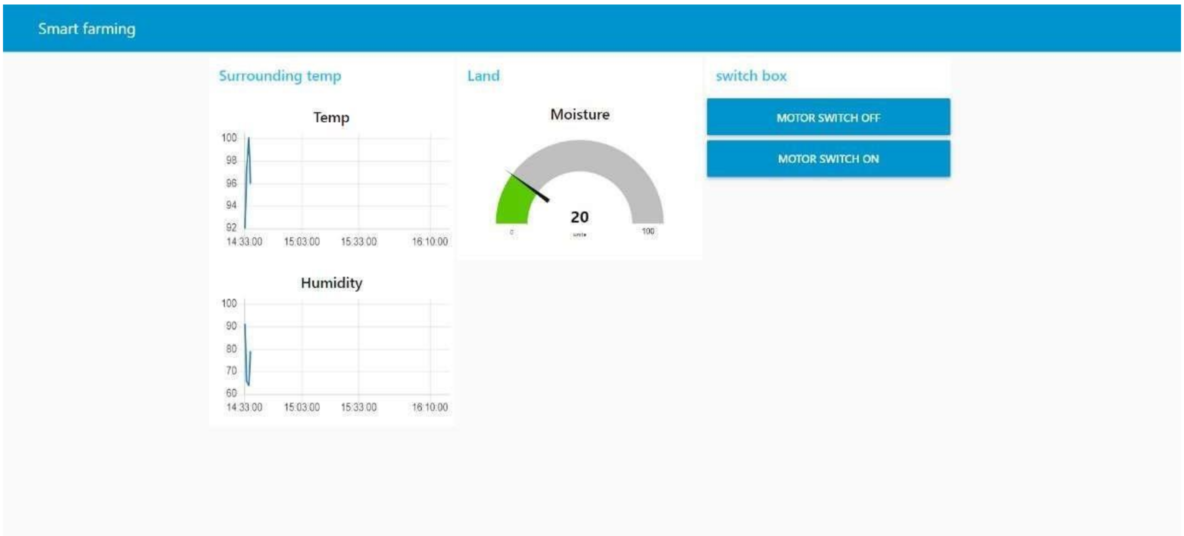


8.2 User Acceptance Testing



9. RESULTS

9.1 PERFORMANCE METRICS



PERFORMANCE METRICS:

S. No.	Name of the Phase	Tasks Performed	Performance Metrics
1.	Development of Problem Statement	The underlying problem analyzed and a rough idea of the solution was planned	The Problem statement was developed
2.	Ideation Phase	Extracting use and test cases	Empathy map, Ideation and Literature survey were formulated.

3.	Project Design Phase 1	Solution for the problem is formulated and architecture is designed	Problem solution fit was designed and the Proposed solution is finalized with the help of Solution architecture.
4.	Project Design Phase 2	In depth analysis of the solution is performed including requirements, tech stack, etc.	Solution Requirements, Overall Technology stack, Data flow diagrams, User stories were formulated.
5.	Project Planning Phase	Various sprints were designed as individual progressive steps.	Project Milestone and Sprint Plans were developed.

10.ADVANTAGES & DISADVANTAGES

ADVANTAGES:

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

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

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

- Submersible weight sensors or ultrasonic sensors can screen the degree of tanks, lakes, wells and different kinds of fluid stockpiling like fuel and compost. The product figures volume dependent on the tank or lake geometry after some time. It conveys alarms dependent on various conditions.
- By monitoring the soil parameters of the farm, the user can have a complete analysis of the field, in terms of numbers.
- Using the website and the application, an interactive experience can be achieved.
- As the data gets pushed to the cloud, one can access the data anywhere from this world.

- Without human intervention, water pump can be controlled through the mobile application and it's flowcan be customized using servo motors.
- By using Raspberry Pi MCU, scalability can be increased due to its high processing power and enough availability of GPIO pins

DISADVANTAGES:

- The smart agriculture needs availability of internet continuously. Rural part of most of the developing countries do not fulfil this requirement. Moreover internet connection is slower.
- The smart farming based equipment require farmers to understand and learn the use of technology. This is major challenge in adopting smart agriculture farming at large scale across the countries.
- Data transfer is through the internet. So data fetch and push might delay due to slow internet connection, depending on the location andother physical parameters.
- System can only monitor a certain area of the field. In order to senseand monitor an entire field, sensors should be placed in many places,which may increase the cost.
- Data accuracy may vary according to various physical parameters such as temperature, pressure, rain.
- Cost of the system is high due to usage of Raspberry Pi. o Rodent and insects may cause damage to the system.

11.CONCLUSION

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

12.FUTURE SCOPE

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

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

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

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

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

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

13.APPENDIX

SOURCE CODE

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "nckdv7"
deviceType = "NodeMCU"
deviceId = "12345"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
print("Command received: %s" % cmd.data['command'])
status=cmd.data['command']
if status=="motoron":
print("Motor is ON")
else:
print("Motor is OFF")
#print(cmd)
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 thecloud as an
event of type "greeting" 10 times
deviceCli.connect()
while True:
#Get Sensor Data from DHT11
temp=random.randint(0,100)
pulse=random.randint(0,100)
moisture= random.randint(0,100)
humidity=random.randint(0,100);
lat = 17
```

```
lon = 18
data = { 'temp' : temp, 'humidity' : humidity, 'Soil Moisture' :
moisture}
#print data
def myOnPublishCallback():
print ("Published Temperature = %s C" % temp, "Humidity
= %s %%" % humidity, "Soil Moisture = %s %%" % moisture,"to
IBM Watson")
success = deviceCli.publishEvent("IoTSensor",
"json", data,qos=0,
on_publish=myOnPublishCallback)
if not success:
print("Not
connected to
IoTF")
time.sleep(1)
deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

OUTPUT:

```
Published Moisture = 90 deg C Temperature = 96 C Humidity = 76 % to IBM Watson
Published Moisture = 102 deg C Temperature = 110 C Humidity = 60 % to IBM Watson
Published Moisture = 45 deg C Temperature = 99 C Humidity = 100 % to IBM Watson
Command received: motoron
motor is on
Published Moisture = 77 deg C Temperature = 91 C Humidity = 85 % to IBM Watson
Published Moisture = 73 deg C Temperature = 94 C Humidity = 86 % to IBM Watson
Command received: motoroff
motor is off
Published Moisture = 101 deg C Temperature = 104 C Humidity = 87 % to IBM Watson
```

GitHub link: **https://github.com/IBM-EPBL/IBM-Project-11328-1659312802**