IBM - Nalaya Thiran

Smart Farmer IoT Enabled Smart Farming Application

Project Report Document

Team ID : PNT2022TMID27283

Team Members :

Rahul R R

Vishnu B

Yuvanchandru M

Ranjan Joseph Arunkumar

College Name : KCG College of Technology

1. *INTRODUCTION*

1.1 Project Overview

- The project consists of a mixture of hardware and software components in which the farmer can use different sensors like humidity, temperature and soil moisture and Softwares such as Smart Farmer App(Mobile App) to get the Live Feed of the different sensors and remotely control the Water Pump for a better yield.

1.2 Purpose

- Precision agriculture lowers total costs by making farming more connected and intelligent while also enhancing product quality and quantity, agricultural sustainability, and consumer experience. Better cost control and waste elimination result from increased production control. It reduces the possibility of produce loss to be able to track anomalies in crop growth or livestock health, for example. Automation also increases effectiveness. Smart devices enable simultaneous activation of numerous processes, and automated services improve product quality and volume by more effectively managing production processes.
- In order to minimize waste, smart farming systems also enable careful monitoring of demand forecasting and timely delivery of goods to markets. In order to produce the proper crop that is in need, precision agriculture focuses on managing the supply of land and, depending on its state, concentrating on the right growing characteristics, such as moisture, fertilizer, or material content. The sorts of precision farming systems that are

employed rely on the business management software that is used. Control systems oversee sensor input, offer distant data for supply and decision support, and automate machinery and equipment for addressing new problems and assisting with production.

2. LITERATURE SURVEY

2.1 Existing problem

- The main factors in traditional farming are the monitoring of temperature, humidity, ammonia, rain, electrical conductivity, air quality, light intensity, and moisture in the litter by physical human interaction.
- Real-time data from the temperature sensor and humidity sensor assists the DSS in maintaining the necessary temperature for crop growth. Water usage is controlled using a rainfall sensor.
- Precision agriculture, often known as smart farming as opposed to traditional farming, is one of the new trends.
- The farmer can get a current and overall view of their field, including the water level, temperature, and humidity of the farming soil, by using IoT with real-time data feed.
- These sensors have a threshold of their own, which aids in automation, which facilitates use and lessens workload.

- Using a network of sensors, a decision support system (DSS) can intelligently distribute water and fertilizer used in crop production based on the age of the plant and information gathered from the soil and surrounding environment.
- Designing statistical data structures utilizing the information from the sensors installed in the farms allows the farmer to increase productivity.
- We use a mobile application that can plot the data using several representations to examine all parameters data in a graphical style, view live and historical data, and save those data in a secure location.
- If the end user is in a remote location, a small area local network can be used to access the data. Cloud technologies can be used to implement the current technology so that data can be accessed remotely.

2.2 References

- An Urban Based Smart IOT Farming System : Njoroge Mungai Bryan1 , Ka Fei Thang2 and Thiruchelvam Vinesh

- FAO 2017 The future of food and agriculture trends and challenges FAO Agri.services Bulletin (Rome)

- Pamidi Srinivasulu , R Venkat , M. Sarath Babu , K Rajesh ” Cloud Service Oriented Architecture (CSoA) for agriculture through Internet of Things (IoT) and Big Data”, 2017

International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE2017)

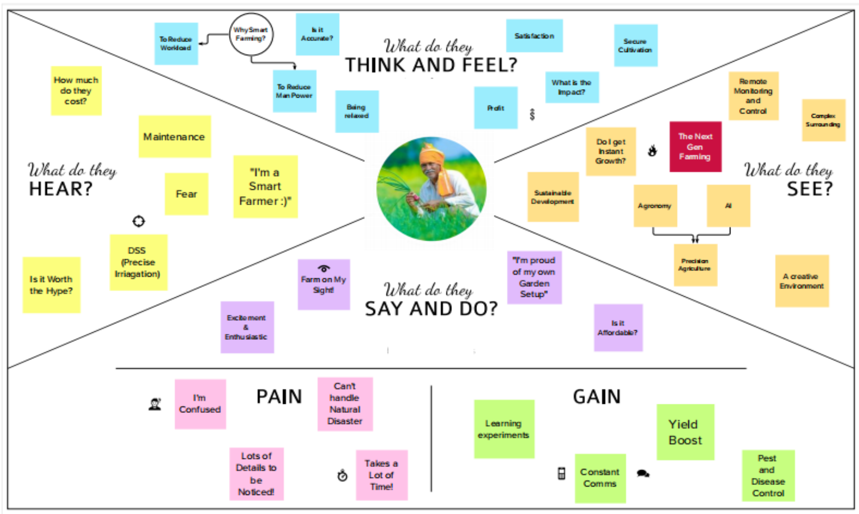
- Rautaray, S. K. (2021). Field Design for Enhancing Water Productivity in Waterlogged Areas with Efficient Water Harvesting and Farming System. Agricultural Research, 10(2), 255-261.

2.3 *Problem Statement Definition*

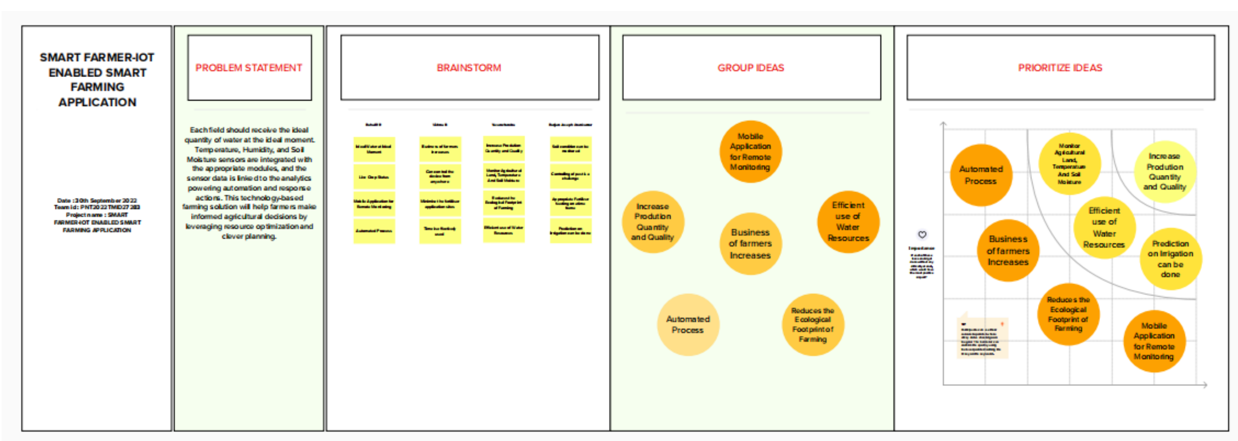
- To facilitate the ability for the farmer to increase productivity by effectively managing the resources and obtaining a greater harvest with the aid of Internet of Things and AI/ML.
- By incorporating IoT sensors, including those needed for the automation process, such as moisture, temperature, light dependent resistor, humidity, and air quality.
- The system has a special algorithm that compares sensor data in addition and adjusts the watering, usage of fertilizers and ventilation system as necessary.
- Real-time data are statistical measures that can be displayed graphically and provide a clear report on the cultivation.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

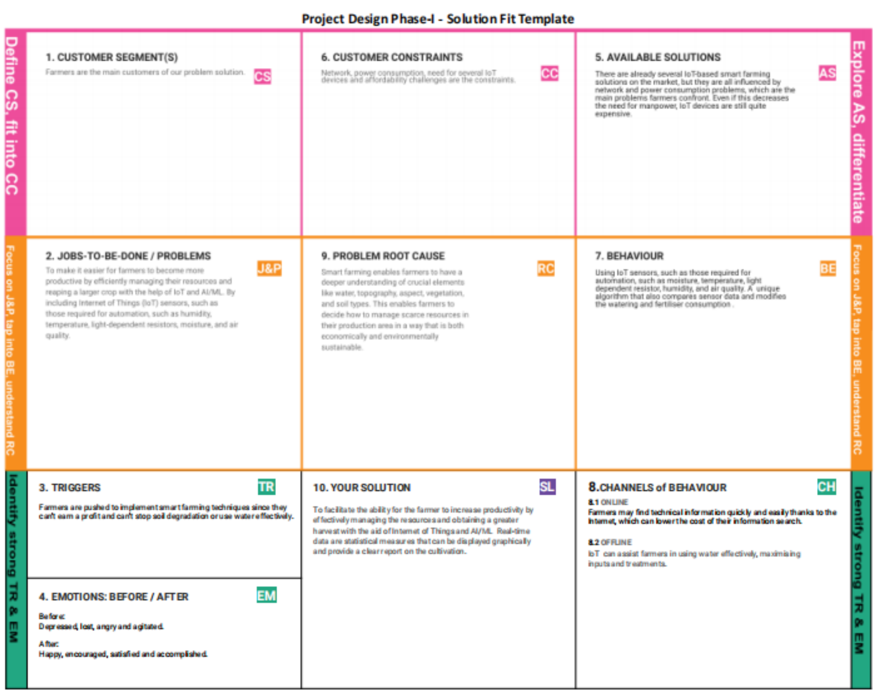


3.3 *Proposed Solution*

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To facilitate the ability of the farmer to increase productivity by effectively managing the resources and obtaining a greater harvest with the aid of the Internet of Things and AI/ML. By incorporating IoT sensors, including those needed for the automation process, such as moisture, temperature, light-dependent resistor, humidity, and air quality. The system has a special algorithm that compares sensor data in addition and adjusts the watering, usage of fertilizers and ventilation system as necessary. Real-time data are statistical measures that can be displayed graphically and provide a clear report on the cultivation.
2.	Idea / Solution description	IOT-based agricultural application
3.	Novelty / Uniqueness	Gather information on the various soil types and predict the yield for the targeted crop to be successfully grown.
4.	Social Impact / Customer Satisfaction	Reduce the workload—time management.

		Farmers can monitor the sensors' parameters through mobile applications.
5.	Business Model (Revenue Model)	Cost Efficient Fuel Efficient
6.	Scalability of the Solution	To increase efficiency, we may combine our current smart farming tools with new innovations.

3.4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirements

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Sensor Function farming system	Measure the Temperature and Humidity Measure the soil Monitoring Check the soil levels
FR-4	Manage Modules	Manage Roles of User Manage Sensors
FR-5	Check Weather details	Temperature details Humdity details
FR-6	Data Management	Manage the data of weather conditions Manage the data of crop conditions Manage the levels of water level conditions

4.2 Non-Functional requirements

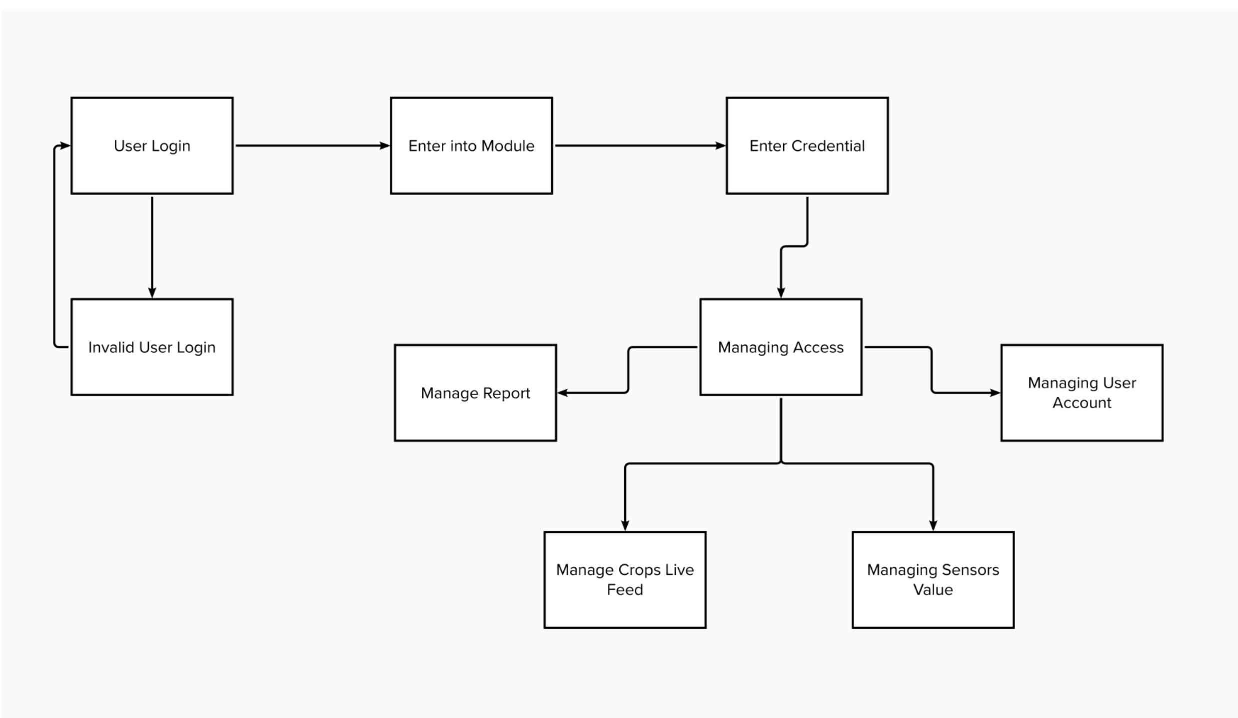
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User-friendly instructions are provided for users to use the functionalities. Users may utilize it easily and efficiently with a simple user interface.
NFR-2	**Security**	All of the user's information is encrypted and hidden from unauthorized users. Only by providing the user details, it is possible to manipulate sensors.

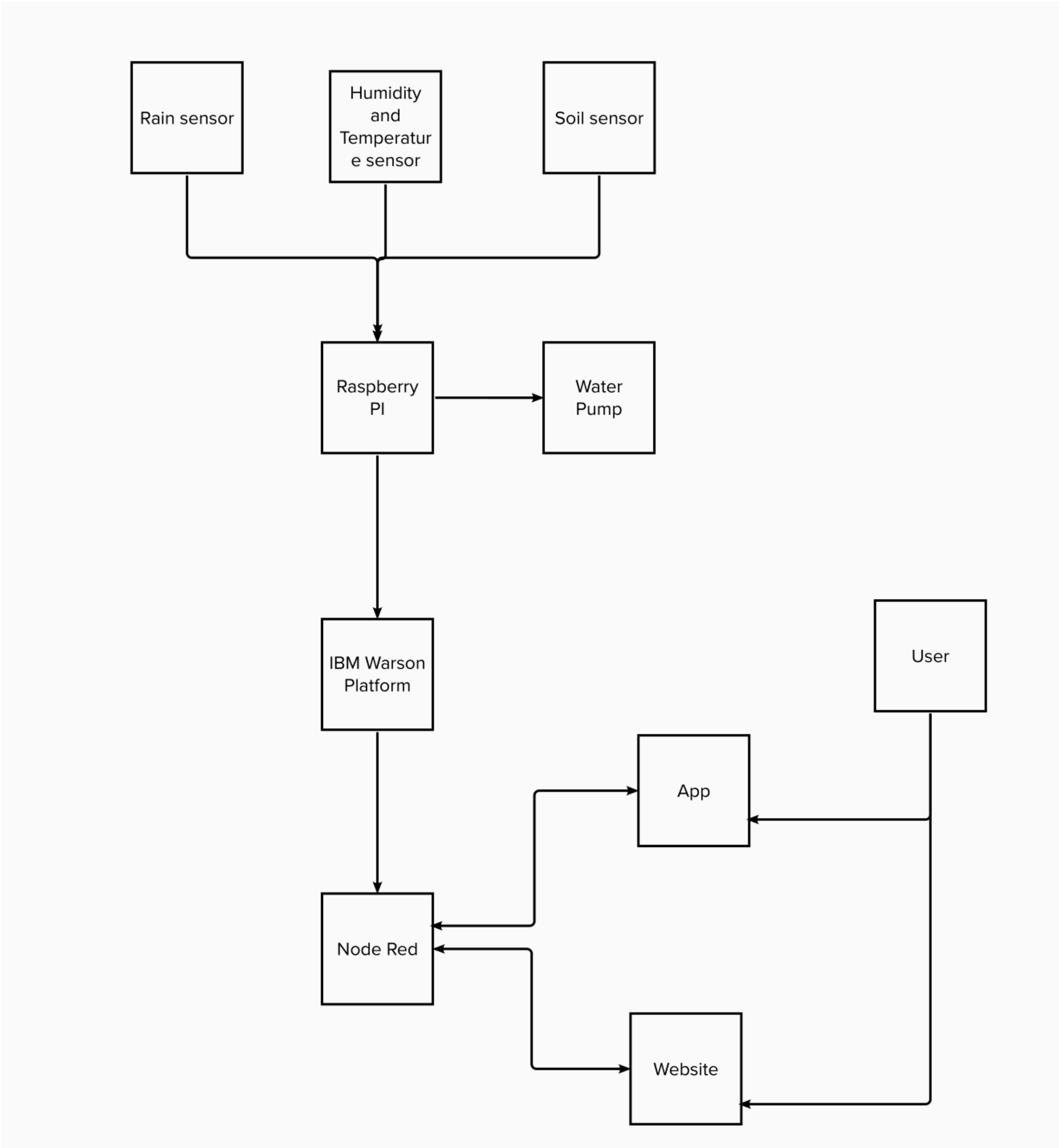
NFR-3	**Reliability**	Since the values of the result will be accurate, this application is trustable and consistent.

NFR-4	**Performance**	Utilizing contemporary technology solutions helps to maximize performance, producing greater quality and quantity yields. The usage of sensors also aids in understanding the water requirements and other necessities for a higher yield.
NFR-5	**Availability**	The application is available in the website and mobile app.
NFR-6	**Scalability**	It refers to the ability to expand resource availability and system capabilities without having to undergo a significant system redesign or implementation.

5. PROJECT DESIGN

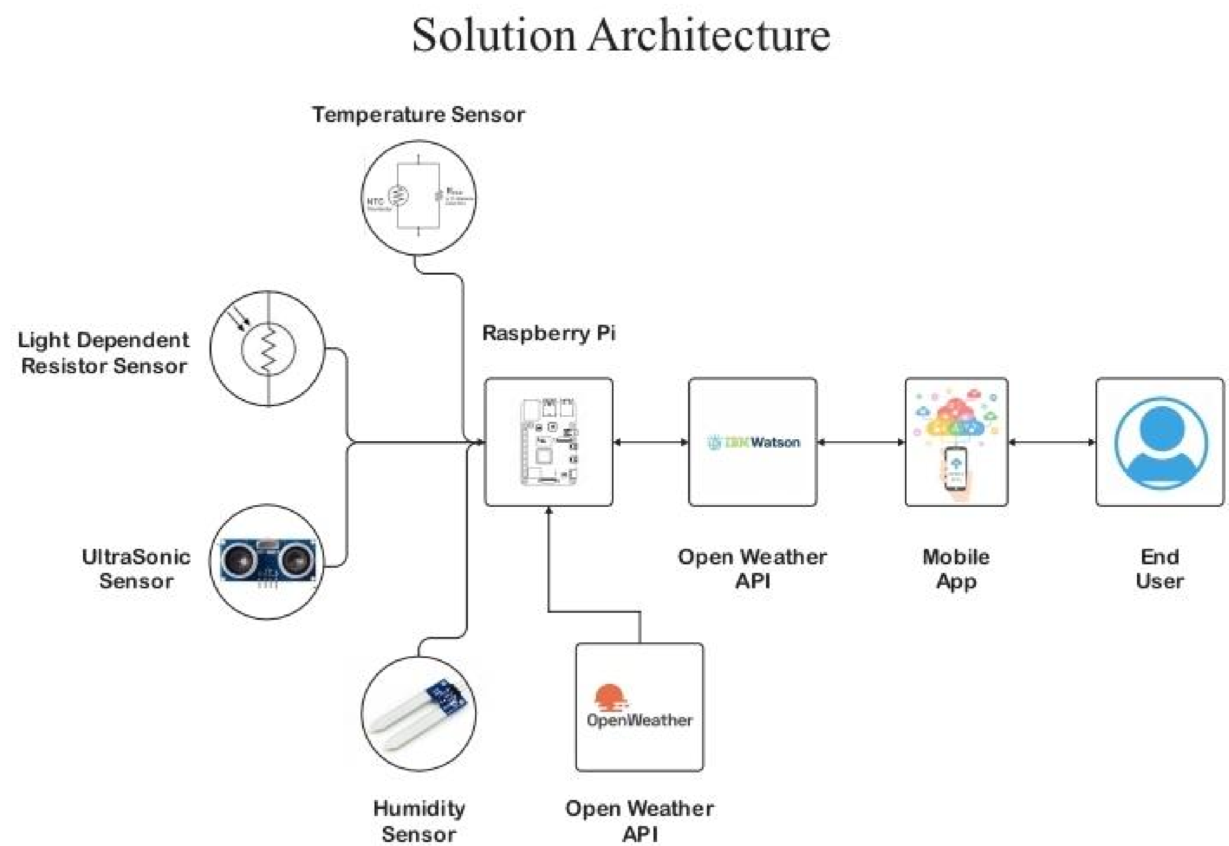
5.1 Data Flow Diagrams

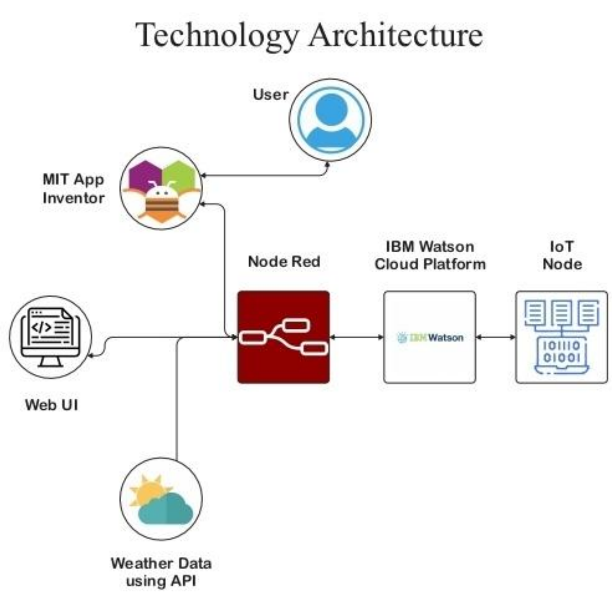




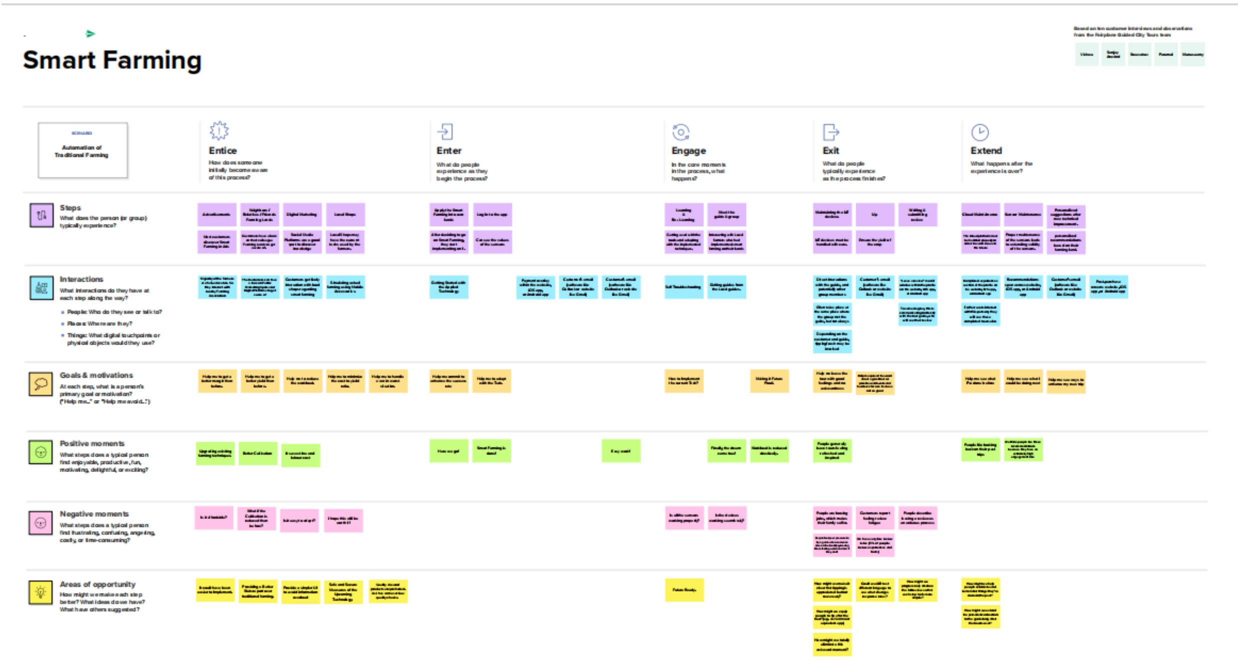
Rain sensor
Humidity and Temperatur e sensor
Soil sensor
Raspberry PI
Water Pump
IBM Warson Platform
User
App
Node Red
Website

5.2 Solution & Technical Architecture





5.3 User Stories



6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

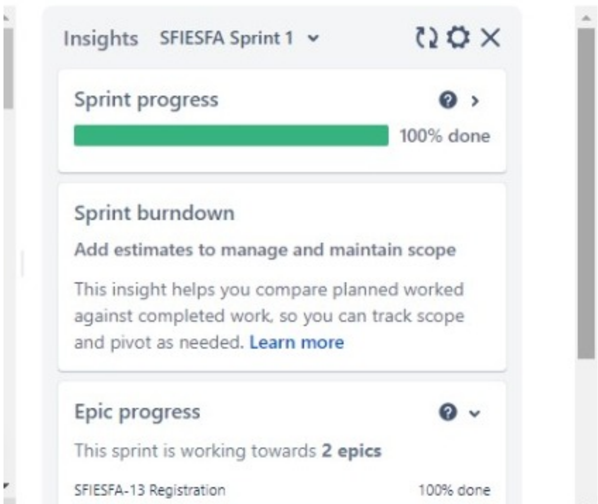
Sprint I	Hardware	USN - 1	Sensors and IoT with necessary code	2	High	Rahul R R
Sprint II	Software	USN - 2	IBM Watson IOT Platform, Workflows	2	High	Yuvanchandru B

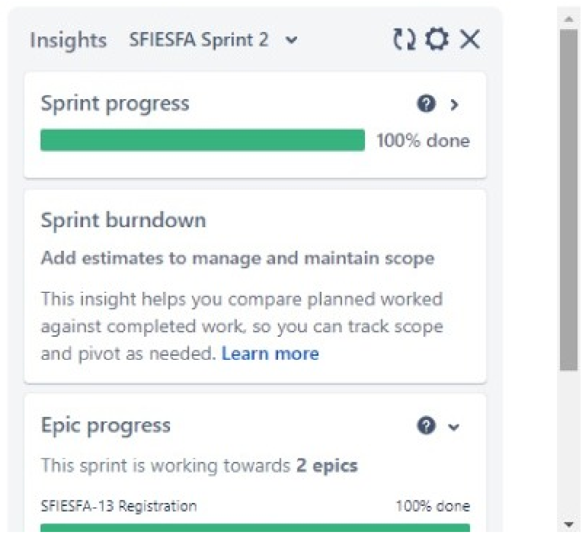
			for IoT scenarios using Node-Red			
Sprint III	MIT Mobile App	USN - 3	Develop MIT Mobile App	2	High	Vishnu B
Sprint IV	Web Application Interface	USN - 4	To make use of MIT App to the end user.	2	High	Ranjan Joseph Arunkumar

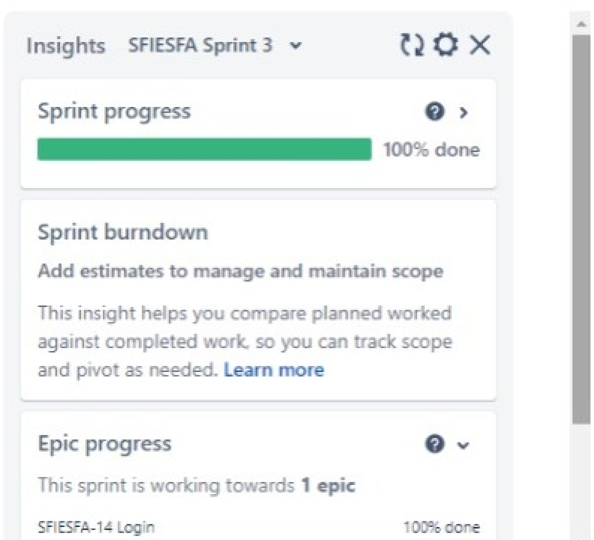
6.2 *Sprint Delivery Schedule*

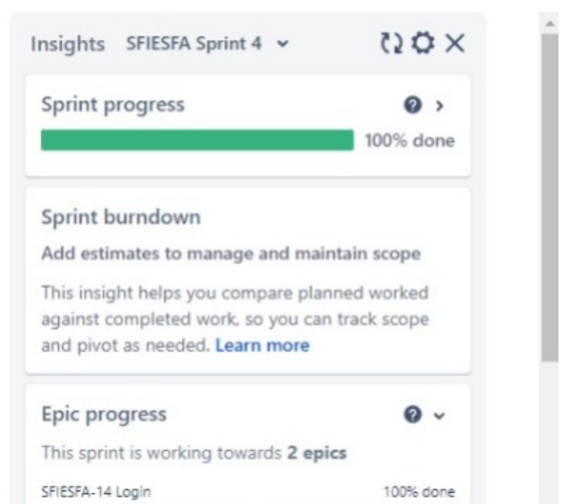
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date
Sprint 1	20	6 Days	24-10-2022	26-10-2022
Sprint 2	21	7 Days	31-10-2022	01-11-2022
Sprint 3	22	8 Days	07-11-2022	07-11-2022
Sprint 4	23	9 Days	14-11-2022	14-11-2022

6.3 *Reports from JIRA*









7. CODING & SOLUTIONING

7.1 Feature 1

ibmfinal.py - D:\IBM Project\ibmfinal.py (3.7.0)

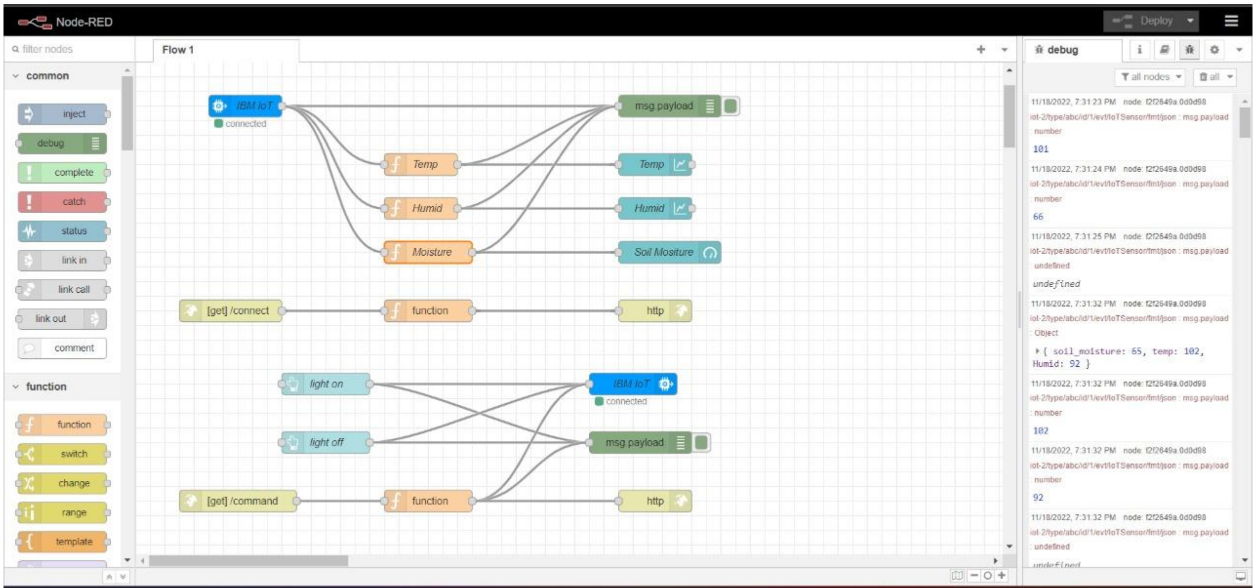
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials

organization = "po6ssd"
deviceType = "abc"
deviceId = "1"
authMethod = "token"
authToken = "12345678"

#Initialize GPIO
def myCommandCallback (cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else:
        print ("please send proper command")

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
    soil=random.randint(60,100)
    temp=random.randint(90,110)
    Humid=random.randint(60,100)

    data = { 'soil_moisture' : soil ,'temp': temp, 'Humid': Humid }
    #print (data)
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Soil Moisture = %s %%" % soil ,  "to IBM Watson")
    success = deviceCli.publishEvent ("IoTSensor", "json", data, qos=0, on publish=myOnPublishCallback)
```

Ln: 21 Col: 0

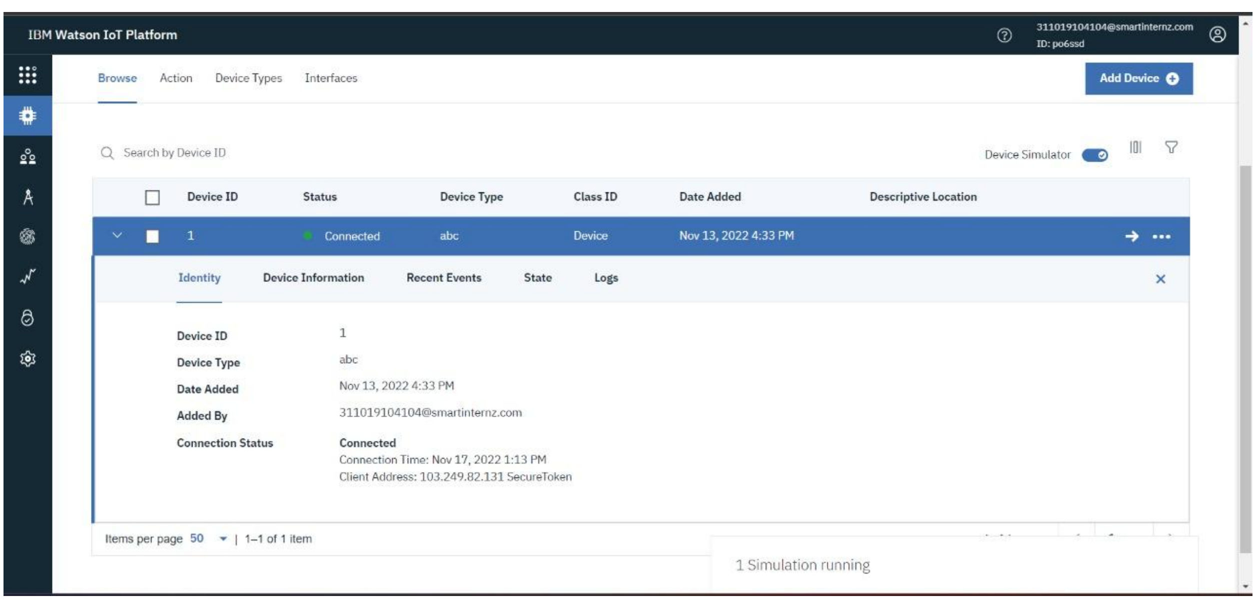
- In python code we provide the IBM Watson Device credentials and initialize general purpose input and output by getting sensor data.

7.2 Feature 2



- Node-RED makes it simple to connect flows using the variety of nodes available in the spectrum. Then, flows can be instantly deployed to the runtime.So here is the connection which shows the input from the Wokwi's simulated output or the Python code's Sensor .

7.3 *Database Schema (if Applicable)*



- In IBM Watson IoT platform

8. TESTING

8.1 Test Cases

Test case ID	Feature Type	Component	Component	Steps To Execute	Test Data	Expected Result	Actual Result	Status	TC for Automation(Y/N)	Executed By
LoginPage_TC_OO1	Functional	Registration Page	As a new user, I want to first register using my email and create a password for the account.	1.Enter URL and click go 2.Register using registration form.	https://github.com/IBM-EPBL/IBM-Project-17368-1659636434/tree/main/Develope%20a%20Mobile%20Application	Login/Signup popup should display	Working as expected	Pass	yes	Vishnu.B
Loginpage_TC_OO2			As a user, I can register for the application by entering my email, password, and	1.Enter URL(https://github.com/IBM-EPBL/IBM-Project-17368-1659636434/tree/main/Develope%20a%20Mobile%20Application) and click go.		Application should show below UI elements: a. Username text box b. Password text box c. Login button with green color		Pass	yes	Yuvanchandru M

			confirming my password.							
LoginPage_TC_OO3	UI	Gmail	As a user, I will receive confirmation email once I have registered for the application	2.Click on SmartFarmer.apk 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button				pass	yes	Rahul RR
LoginPage_TC_OO4		Home page	As a user, I can register for the application through Facebook		Username: Team Marvel password: 1234	User should navigate to user account Login page		pass	yes	Yuvanchandru M
LoginPage_TC_OO5	Functional	Login page	As a user, I can register for the application through GMAIL		Username: Team DC password: 1316	Application should show 'Incorrect email or password ' validation message		pass	yes	Vishnu.B

LoginPage_TC_OO6			As a user, I can log into the application by entering email and password		Username: Team Jack password: 2419	Application should show 'Incorrect email or password ' validation message if credientials given wrong		pass	yes	Ranjan Joseph Arunkumar
LoginPage_TC_OO7			As a registered user, I need to easily login log into my registered account via the web page in minimum time		Username: ijoicy02@gmail.com password: 1136			pass	yes	Rahul RR

8.2 User Acceptance Testing

Purpose of Document

- The purpose of this document is to briefly explain the test coverage and open issues of the [Smart Farmer Application] project at the time of the release to User Acceptance Testing (UAT).

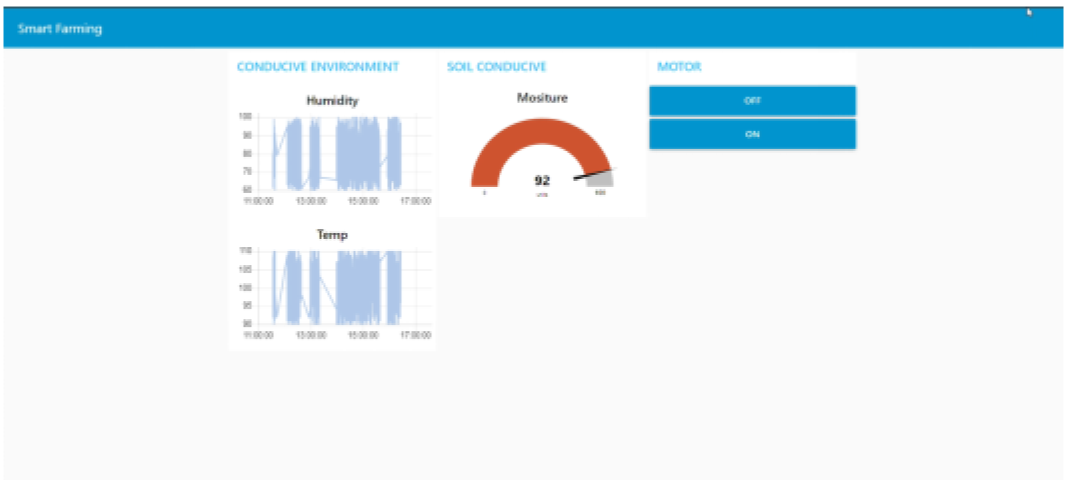
Defect Analysis

- This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	1	0	0	2
Duplicate	0	0	0	0	0
External	1	1	0	0	2
Fixed	1	1	1	0	3
Not Reproduced	0	0	0	0	0
Skipped	0	1	0	0	1
Won't Fix	0	0	0	0	0
Totals	3	4	1	0	8

9. RESULTS

9.1 Performance Metrics



10. ADVANTAGES & DISADVANTAGES

10.1 ADVANTAGES

- The ability to use soil sensing is one of the very great things about this area of farming. This component of intelligent farming allows you as a farmer to test your soil for information and measure it for a variety of significant and nutritious constituents required in ensuring the health of your farm products.

- In order to properly control the use of real-time variable rate equipment, soil sensing is also used. This enables you to comprehend the size of your property, enabling you to devise efficient methods of saving essential farming resources like water, fertilizer, and so forth. In order to avoid harming your plants, you only need to use fertilizers and insecticides where they are necessary. Additionally, you get to minimize waste of seeds, fertilizer, water, etc. while still achieving maximum harvests. Additionally, you receive access to crucial information on the volume and intensity of the air in your area as well as its levels of sound, humidity, and temperature.
- If you want to reduce your electricity costs, smart farming is a fantastic choice. It enables the use of cost-effective solar-powered equipment like pumps. It is economical because it somewhat lowers the maintenance costs that farmers typically incur for their expensive equipment.
- AI is used in smart agriculture to enhance the wireless monitoring, control, and data collection processes. With these inputs on your farm, you can be certain of high-quality crop production and delivery thanks to smart farming.

10.2 DISADVANTAGES

- The fact that smart farming necessitates an unrestricted or ongoing internet connection for success is a major drawback. This means that using this agricultural method in rural areas, especially in developing nations where we produce large quantities of crops, is utterly unfeasible. Smart farming won't be possible in locations with excruciatingly slow internet connections.

- As was already said, smart farming uses high-tech tools that require technical know-how and accuracy to be successful. It calls for knowledge of ICT and robotics. Many farmers, meanwhile, lack these abilities. Finding someone with this level of technical proficiency is at best challenging or expensive. This might be disappointing given the advantages and disadvantages of smart farming.

11. CONCLUSION

- ***Agriculture's future is bright:*** Smart farming is built on the Internet of Things. Farmers may aim their knowledge for a greater harvest thanks to remote access and minimal access security.
- As a result, the gap between quantity and quality is reduced. Real-time data and sophisticated sensor technology enable quicker response. For an efficient and seamless end-to-end operation, these are essential.

12. FUTURE SCOPE

- Smart farming and IoT technology are the foundation of the "Third Green Revolution." Precision farming and data-driven analytics are combined in this new development.
- It is clearly clear that this is a progressive course of action: Smart Farming will be utilized to decrease pesticides and fertilizers while raising yields.

- The technology also makes it possible for increased traceability and greater food safety. Thus, smart farming is viewed as a revolution that helps the environment and humankind.

13. APPENDIX

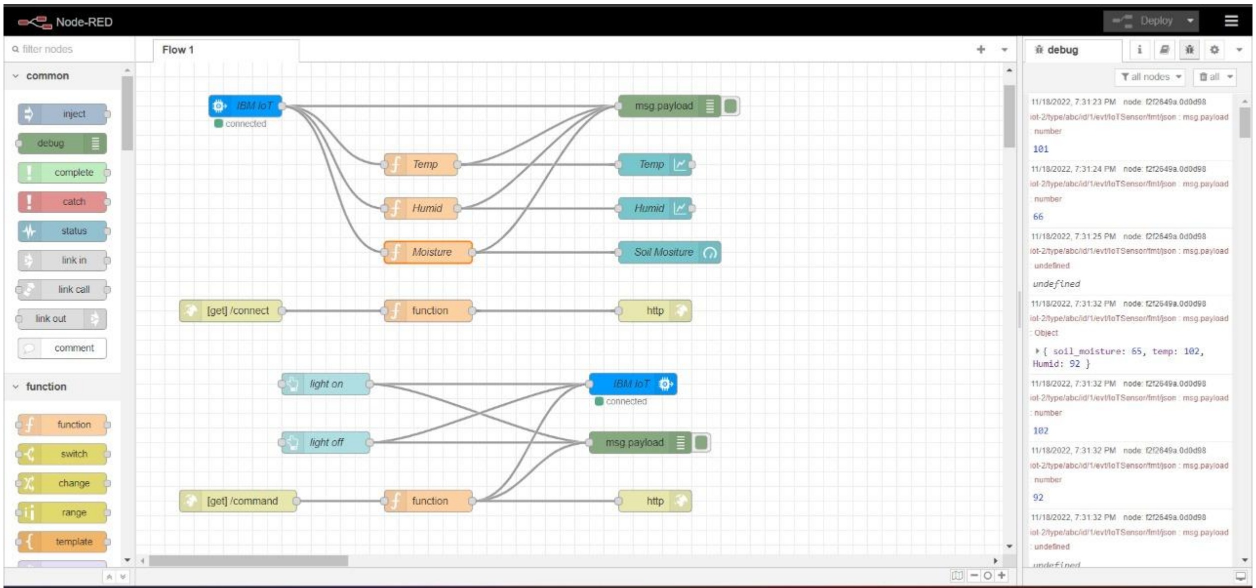
13.1 Source Code

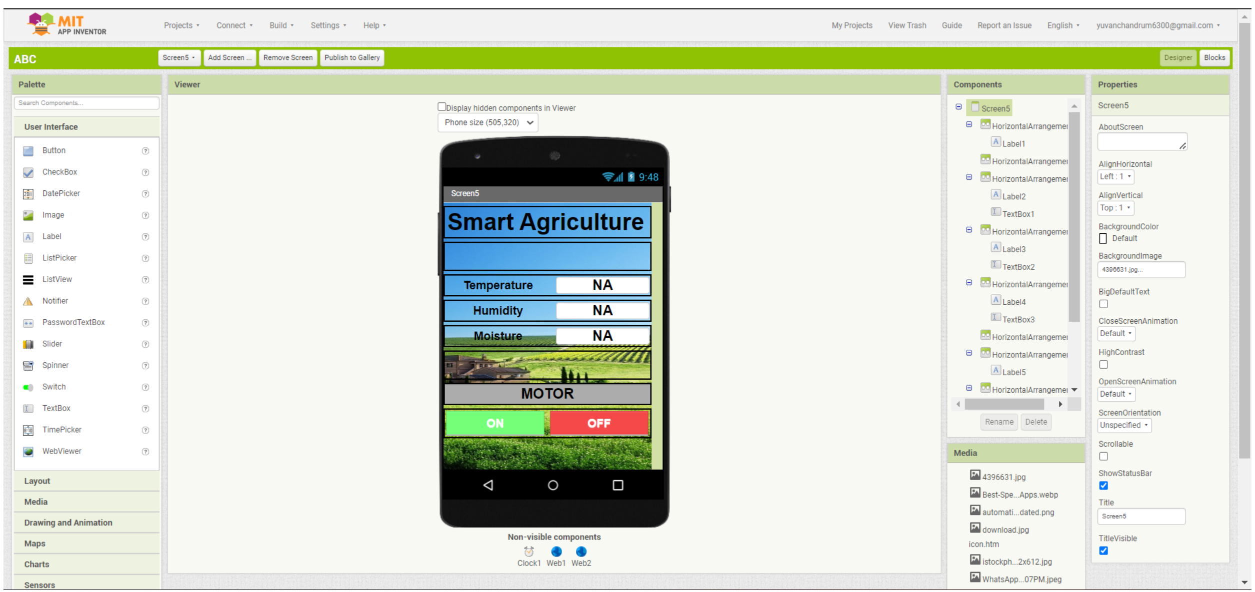
ibmfinal.py - D:\IBM Project\ibmfinal.py (3.7.0)

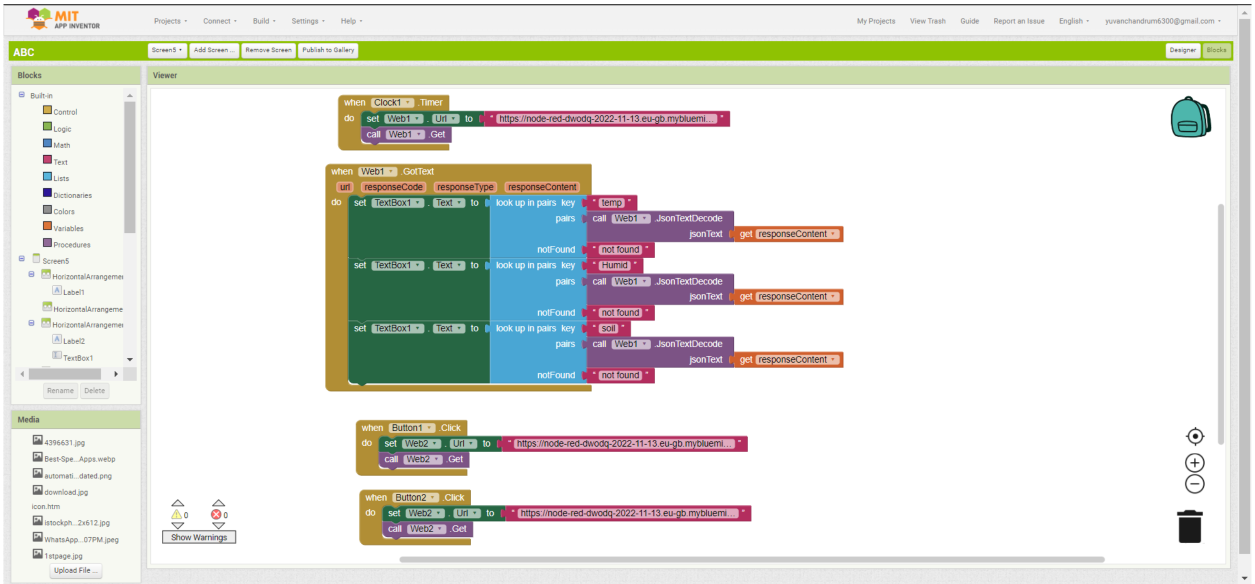
File Edit Format Run Options Window Help

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

#Provide your IBM Watson Device Credentials

organization = "po6ssd"
deviceType = "abc"
deviceId = "1"
authMethod = "token"
authToken = "12345678"

#Initialize GPIO
def myCommandCallback (cmd):
    print ("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print ("led is on")
    elif status == "lightoff":
        print ("led is off")
    else:
        print ("please send proper command")

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
    soil=random.randint(60,100)
    temp=random.randint(90,110)
    Humid=random.randint(60,100)

    data = { 'soil_moisture' : soil ,'temp': temp, 'Humid': Humid }
    #print (data)
    def myOnPublishCallback():
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid,"Soil Moisture = %s %%" % soil ,  "to IBM Watson")
    success = deviceCli.publishEvent ("IoTSensor", "json", data, qos=0, on publish=myOnPublishCallback)
```

Ln: 21 Col: 0


Node-RED
Deploy
Flow 1
debug
common
inject
debug
complete
catch
status
link in
link call
link out
comment
function
function
switch
change
range
template
IBM IoT
connected
msg.payload
Temp
Humid
Moisture
Temp
Humid
Soil Mositure
[get] /connect
function
http
light on
light off
IBM IoT
connected
msg.payload
[get] /command
function
http


MIT APP INVENTOR
ABC
Palette
Viewer
Components
Properties
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
Switch
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Maps
Charts
Sensors
Smart Agriculture
Temperature
NA
Humidity
NA
Moisture
NA
MOTOR
ON
OFF
Non-visible components



13.2 *GitHub & Project Demo Link*

Github Link :

https://github.com/IBM-EPBL/IBM-Project-17368-1659636434

Project Demo Link :

https://drive.google.com/drive/folders/1GxVK8Y894Nx_ovNsiDe1laiXOz2ixxZY?usp=share_link