|  |  |
| --- | --- |
| PROJECT DOCUMENTATION | |
| TEAM ID | PNT2022TMID07016 |
| PROJECT NAME | Smart Farmer - IoT Enabled Smart Farming Monitoring Application |

1. **INTRODUCTION**
   1. Project Overview

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and 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. 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.

* 1. Purpose

India’s major source of income is from agriculture sector and 70% of farmers and general people depend on the agriculture. In India most of the irrigation systems are operated manually. These outmoded techniques are replaced with semi-automated and automated techniques. The available traditional techniques are like ditch irrigation, terraced irrigation, drip irrigation, sprinkler system. The global irrigation scenario is categorized by increased demand for higher agricultural productivity, poor performance and decreased availability of water for agriculture. These problems can be appropriately rectified if we use automated system for irrigation.

NEED FOR AUTOMATIC IRRIGATION:

➢Simple and easy to install and configure.

➢ Saving energy and resources, so that it can be utilized in proper way and amount.

➢ Farmers would be able to smear the right amount of water at the right time by automating farm or nursery irrigation.

➢ Avoiding irrigation at the wrong time of day, reduce runoff from overwatering saturated soils which will improve crop performance.

➢ Automated irrigation system uses valves to turn motor ON and OFF. Motors can be automated easily by using controllers and no need of labour to turn motor ON and OFF.

➢ It is precise method for irrigation and a valuable tool for accurate soil moisture control in highly specialized greenhouse vegetable production.

➢ It is time saving, the human error elimination in adjusting available soil moisture levels

1. **LITERATURE SURVEY**
   1. Existing problem

IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and 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. 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.

* 1. References

[1] R.Suresh, S.Gopinath, K.Govindaraju, T.Devika, N.SuthanthiraVanitha, “GSM based Automated Irrigation Control using Raingun Irrigation System”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 2, February 2014.

[2] Pavithra D.S, M. S .Srinath, “GSM based Automatic Irrigation Control System for Efficient Use of Resources and Crop Planning by Using an Android Mobile”, IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE) Vol 11, Issue I, Jul-Aug 2014, pp 49-55.

[3] LaxmiShabadi, NandiniPatil, Nikita. M, Shruti. J, Smitha. P&Swati. C, “Irrigation Control System Using Android and GSM for Efficient Use of Water and Power”,International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 7, July 2014.

[4] Shiraz Pasha B.R., Dr. B Yogesha, “Microcontroller Based Automated Irrigation System”, The International Journal Of Engineering And Science (IJES), Volume3, Issue 7, pp 06-09, June 2014.

[5] S. R. Kumbhar, Arjun P. Ghatule, “Microcontroller based Controlled Irrigation System for Plantation”, Proceedings of the International MultiConference of Engineers and Computer Scientists 2013Volume II, March 2013.

[6] Yunseop (James) Kim, Member, IEEE, Robert G. Evans, and William M. Iversen, “Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor

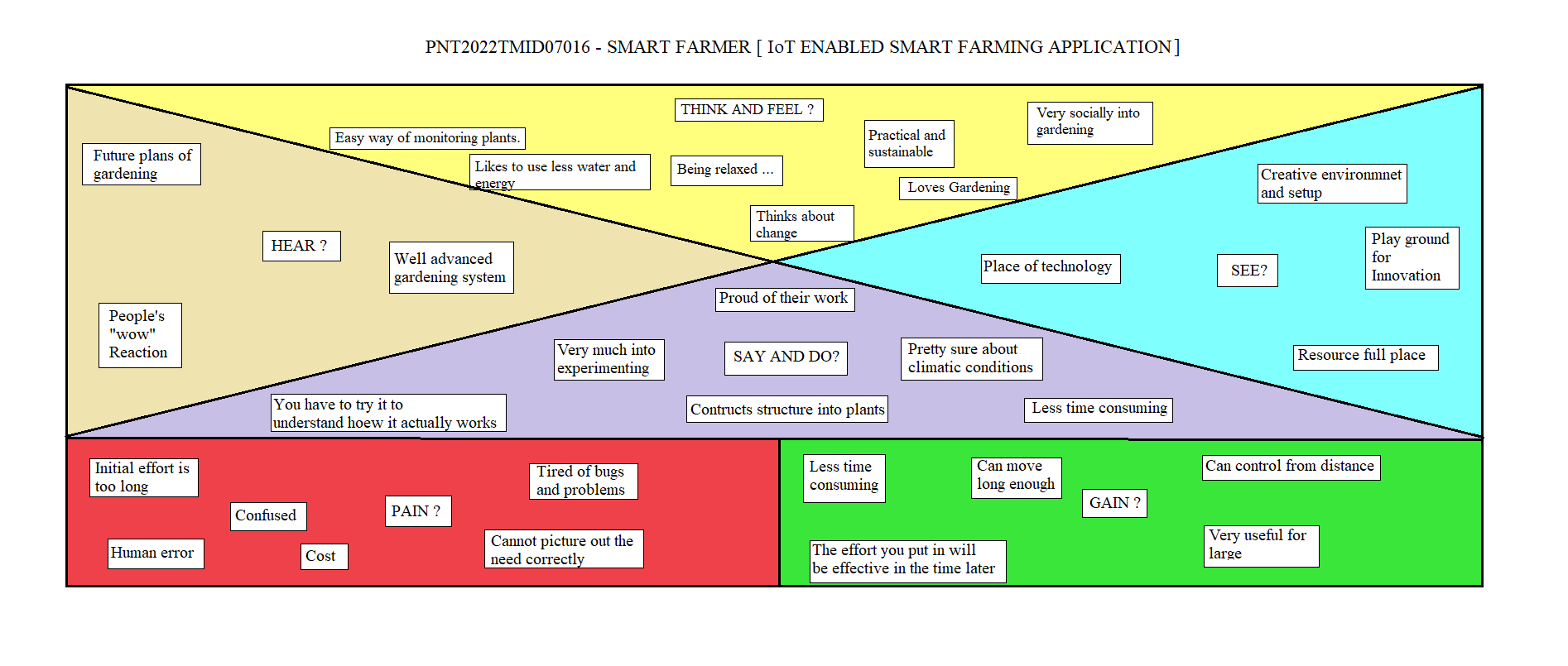
* 1. Problem Statement Definition

• IoT-based agriculture system helps the farmer in monitoring different parameters of his field like soil moisture, temperature, and 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. 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.

1. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas



* 1. Ideation & Brainstorming

Idea 1

The sensors to sense the soil moisture, humidity and temperature should be placed in the field The readings from the sensors are updated to the cloud storage.

The readings can be monitored using the mobile application.

User can analyse the situation and the data of the sensors

Based on the analysis, user can made decision whether to irrigate the field or not using the mobile application itself.

And from the same user can stop irrigating the field by analysing the sensor values

Idea 2

The sensors to sense the soil moisture, humidity and temperature should be placed in the field.

The readings from the sensors are updated to the cloud storage.

The readings are evaluated by the program with the minimum value and with the maximum value.

Based on the evaluation, Alert message can be thrown to the user when the sensor value reaches to the maximum range or it reaches down the minimum range.

When the user gets the alert message, he/she can verify and can make decision for the situation.

In addition, the level of water in the well can be monitored and also updated to the cloud. Now the user can also get the idea about the water level in the field.

Automation can be done

When the water level reaches the minimum level or the field reaches the maximum moisture level, the motor can be turned off automatically.

When the soil moisture reaches to minimum level and the water level is needed enough , the motor can be automatically turned on.

Idea 3

The thermal camera can be placed in a drown.

By analysing and processing the thermal footage of the field, we can get the humidity level, temperature level., etc.

The data after processing the footage are updated to the cloud storage.

The readings are evaluated by the program with the minimum range and with the maximum range.

Based on the evaluation, Alert message can be thrown to the user when the sensor value reaches to the maximum range or it reaches down the minimum range.

When the user gets the alert message, he/she can verify and can make decision for the situation.

* 1. Proposed Solution

The sensors to sense the soil moisture, humidity and temperature should be placed in the field.

The readings from the sensors are updated to the cloud storage.

The readings are evaluated by the program with the minimum value and with the maximum value.

Based on the evaluation, Alert message can be thrown to the user when the sensor value reaches to the maximum range or it reaches down the minimum range.

When the user gets the alert message, he/she can verify and can make decision for the situation.

In addition, the level of water in the well can be monitored and also updated to the cloud. Now the user can also get the idea about the water level in the field.

Automation can be done

When the water level reaches the minimum level or the field reaches the maximum moisture level, the motor can be turned off automatically.

When the soil moisture reaches to minimum level and the water level is needed enough , the motor can be automatically turned on.

* 1. Problem Solution fit

|  |  |  |
| --- | --- | --- |
| 1.CUSTOME SEGMENT(s)  Customers involve in this project are  Large scale farmers  Remote farmers | 6.CUSTOMER LIMITATIONS  The customer needs a solution which will solve the problems in farming when he is in a remote location and that solution should fulfil the following needs.  • Cost efficient  • Low power consumption  • Time efficient | 5.AVAILABLE SOLUTIONS  We can give solutions to this problem by using the Smart Farming Application which collects the Moisture level data from the field and operate in the basis of that moisture level |
| 2.JOBS TO BE DONE  The Customers want to automate the irrigation process, reduce cost of manual workers and minimize the power consumption | 9.PROBLEM ROOT/CAUSE  The route cause for Smart farming Application is farmer’s need to be feel comfortable. | 7.BEHAVIOUR  The customer needs to make a revolutionary change in farming by means of modern technologies. |
| 3.TRIGGERS  Farmers are facing many problems while farming in traditional manner. This triggers the Smart Farming Applications. | 10.SOLUTION  Our solution for this project is to give environment sustainable Product for the farming in modern era with reduced cost and with best efficiency. | 8.CHANNELS OF BEHAVIOUR  The channels of behaviour recombine the ration of the following  • Online  • Offline |
| 4.EMOTIONS  Farmers feel very relaxed and feel stressless while working in field. |

Functional requirement

Following are the functional requirements of the proposed solution.

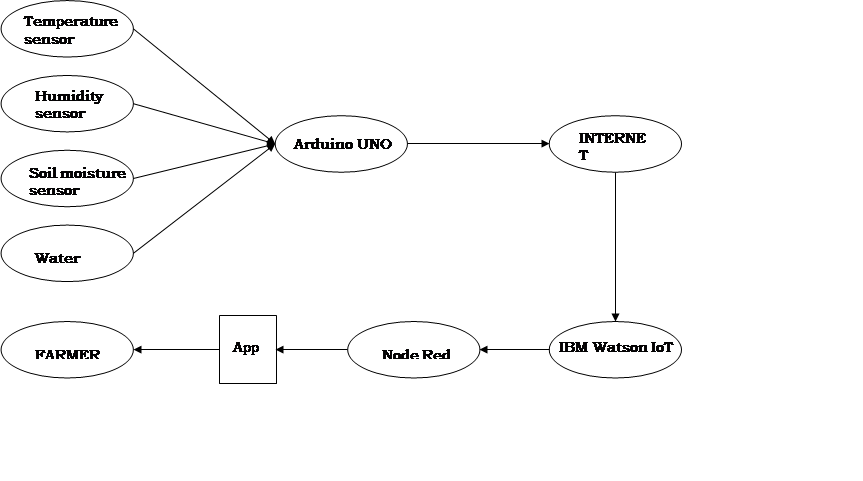
|  |  |  |
| --- | --- | --- |
| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
| Online | | |
| FR-1 | User Registration | Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email  Confirmation via OTP |
| FR-3 | Cloud Account | Creating an IBM cloud account  Sign in and confirmation via OTP/Mail |
| FR-4 | MIT App Account | Download MIT App  Sign up/Sign in MIT App  Confirmation via OTP/Mail |
| Offline | | |
| FR-1 | Sensor Setup | Setting up of required sensors in required places Connecting the main controller to the IBM cloud platform |

Non-Functional requirements

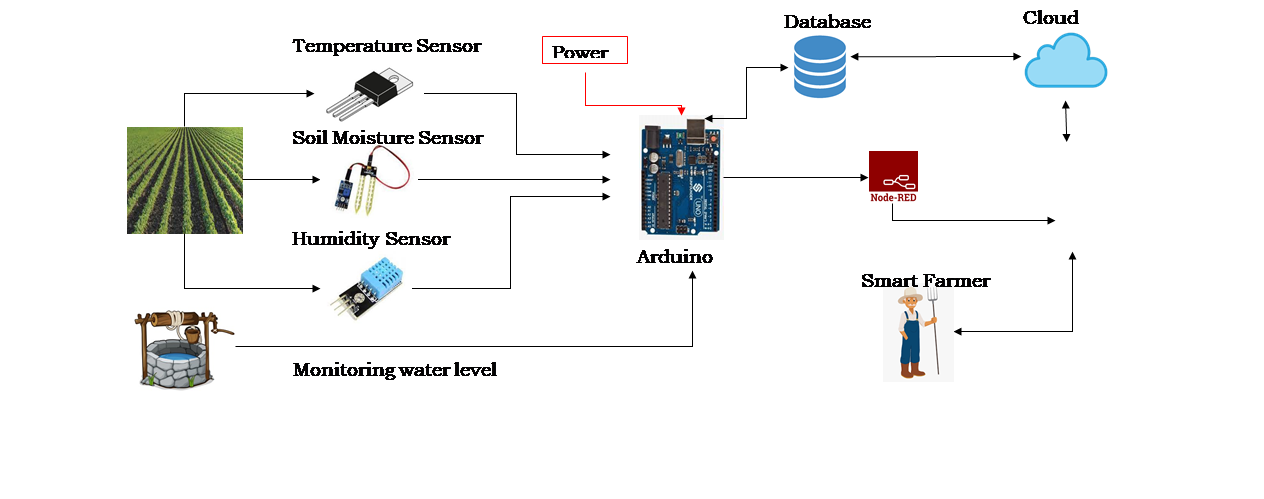
|  |  |  |
| --- | --- | --- |
| FR No. | Non-Functional Requirement | Description |
| NFR-1 | Usability | Usability includes easy learnability, efficiency in use, remembering, and subjective pleasure. |
| NFR-2 | Security | Data will be protected from their production until  the decision-making and storage stages. |
| NFR-3 | Reliability | By using a share protection scheme we can provide better security at optimal cost |
| NFR-4 | Performance | The idea of implementing integrated sensors in the  field will be more efficient for overall monitoring. |
| NFR-5 | Availability | Data is will stored in the cloud and so will be available globally. |
| dNFR-6 | Scalability | Since cloud technology has a variety of scalability  options we can scale based on the needs in real-time |

1. **PROJECT DESIGN**
   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. Solution & Technical Architecture



* 1. User Stories

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
| Customer (Mobile user) | Registration | USN-1 | As a user, I can register for the application  by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation 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 | I can register & access the dashboard with Gmail Login | Medium | Sprint-1 |
|  | Login | USN-5 | As a user, I can log into the application by entering email & password | I can access dashboard with email login | High | Sprint-1 |
|  | Dashboard | USN-6 | As a user I can enter into dashboard by using navigation panel | I can access the dashboard by using navigation panel | High | Sprint-1 |
| Customer (Web user) | Registration | USN-1 | As a user, I can register for the web 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 web application | I can receive confirmation email & click confirm | High | Sprint-1 |
|  | Login | USN-3 | As a user, I can log into the web application by entering email & password | I can access dashboard with email login | High | Sprint-1 |
|  | Dashboard | USN-4 | As a user I can enter into web dashboard by using navigation panel | I can access into dashboard by using navigation panel | High | Sprint-1 |
| Customer  Care Executive | Registration | USN-1 | As a user I can contact the customer care service through phone or mail medium | I can receive confirmation SMS or email | High | Sprint-1 |
|  |  | USN-2 | As a user I want customer care to answer the questions related to product and services | I can get the problem solved within a day | High | Sprint-1 |
|  |  | USN-3 | As a user I want customer care to register my complaints | I can receive a confirmation message stating my complaint is registered | High | Sprint-1 |
|  |  | USN-4 | As a user I want customer care to collect and analyses consumer feedback | I can get the status of my feedback | High | Sprint-1 |
|  |  | USN-5 | As a user I want customer care to troubleshoot technical problems | I can get the problem solved within a day | High | Sprint-1 |
| Administrator |  | USN-1 | As a user I want the administrator to use good working hardware | I can get a guarantee and warranty card | High | Sprint-1 |
|  |  | USN-2 | As a user I want the administrator to sell the product in a reasonable rate | I can get the cost of bill of materials | High | Sprint-1 |
|  |  | USN-3 | As a user I want the administrator to refund my amount if I am not satisfied with the product | I can get an assurance stating I will get my amount back | High | Sprint-1 |

1. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
| Sprint-1 | Objective | USN-1 | As a system, the soil moisture should detect the soil moisture level, humidity sensor should detect the humidity present in atmosphere, temperature sensor detects the  temperature | 8 | High | Manoj T Rohith R |
| Sprint-1 | Features | USN-2 | As a system, water level indicator tells us  about the level of water in the well | 2 | High | Sathishkumar S Vishnukumar D |
| Sprint-1 | Features | USN-3 | As a system, when water reaches the  minimum threshold value it alerts with LED | 5 | Low | Vishnukumar D Manoj T |
| Sprint-1 | Features | USN-4 | As a system, when soil moisture reaches minimum value or reaches to maximum value, it alerts with red and green LED  respectively | 5 | Low | Sathishkumar S Rohith R |
| Sprint-2 | Focus | USN-5 | As a system, when water reaches the  minimum threshold value it automatically turns off the motor | 10 | High | Sathishkumar S Manoj T |
| Sprint-2 | Focus | USN-6 | As a system, when soil moisture reaches  minimum value or reaches to maximum value, it alerts the user through notification | 10 | High | Vishnukumar D Rohith R |
| Sprint 2 | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | Priority | Team Members |
| Sprint-3 | Data Transfer | USN-6 | As a program, it should retrieve the API key of the IBM cloud to send the details of the system. | 2 | Low | Rohith R Vishnukumar D |
| Sprint-3 | Data Transfer | USN-7 | As a system, it should send the data of sensor values along with water level data to the IBM cloud | 5 | Medium | Manoj T Sathishkumar S |
| Sprint-3 | Data Transfer | USN-8 | As a cloud system, the IBM cloud should send the data to NodeRed | 2 | Medium | Vishnukumar D Sathishkumar S |
| Sprint-3 | Data Transfer | USN-9 | As a system, it should collect the data from the NodeRed and give it to the backend of the mit app. | 3 | Medium | Rohith R Manoj T |
| Sprint-3 | Data Transfer | USN-10 | As an application, it should display the details of the sensor values, water level value and other details to the user  through the frontend of the mit app. | 8 | High | Vishnukumar D Sathishkumar S |
| Sprint-4 | Registration | USN-11 | As a user, I must first register my email and mobile number in the website | 2 | High | Sathishkumar S Rohith R |

* 1. Sprint Delivery Schedule

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total Story Points |  | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
| Sprint-1 | 20 |  | 6 Days | 24 Oct 2022 | 29 Oct 2022 |  | 29 Oct 2022 |
| Sprint-2 | 20 |  | 6 Days | 31 Oct 2022 | 05 Nov 2022 |  | 05 Nov 2022 |
| Sprint-3 | 20 |  | 6 Days | 07 Nov 2022 | 12 Nov 2022 |  | 12 Nov 2022 |
| Sprint-4 | 20 |  | 6 Days | 14 Nov 2022 | 19 Nov 2022 |  | 19 Nov 2022 |

1. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. Feature 1

Make use of sensors and getting data from the sensor and sending the data to the IoT Watson platform

Coding

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "r3m467"

deviceType = "NalaiyaThiran"

deviceId = "NalaiyaThiran"

authMethod = "token"

authToken = "NalaiyaThiran"

# Initialize GPIO

def myCommandCallback(cmd):

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

    status=cmd.data['Command']

    if status=="Motor ON":

        print ("Motor turned ON")

    elif status == "Motor OFF":

        print ("Motor turned OFF")

    else :

        print ("Invalid Command")

try:

    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-method": authMethod,"auth-token":authToken}

    deviceCli = ibmiotf.device.Client(deviceOptions)

    deviceCli.connect()

#..............................................

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

while True:

    #Get Sensor Data from DHT11

    temp=random.randint(15,35)

    Mois,Humid=random.randint(200,60000)//1000,random.randint(200,60000)//1000

    Water = 60

    time.sleep(10)

    data = { 'Temperature' : temp, 'Humidity': Humid, 'Moisture' :Mois, 'Water':Water}

    #print data

    def myOnPublishCallback():

        print ("Published Temperature= %s C" % temp, "Humidity = %s%%" % Humid, "Moisture =%s%%" %Mois,"Water Level =%s%%" %Water, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

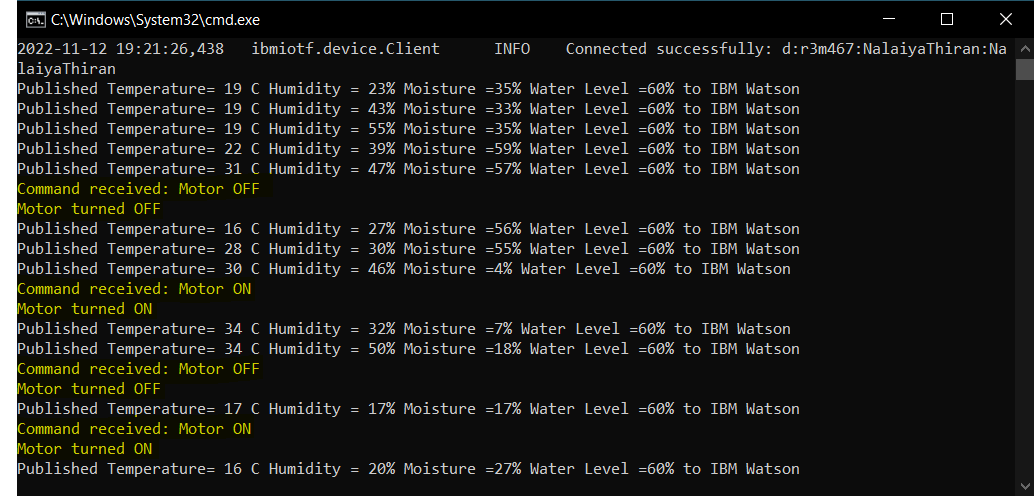
    deviceCli.commandCallback = myCommandCallback

    if not success:

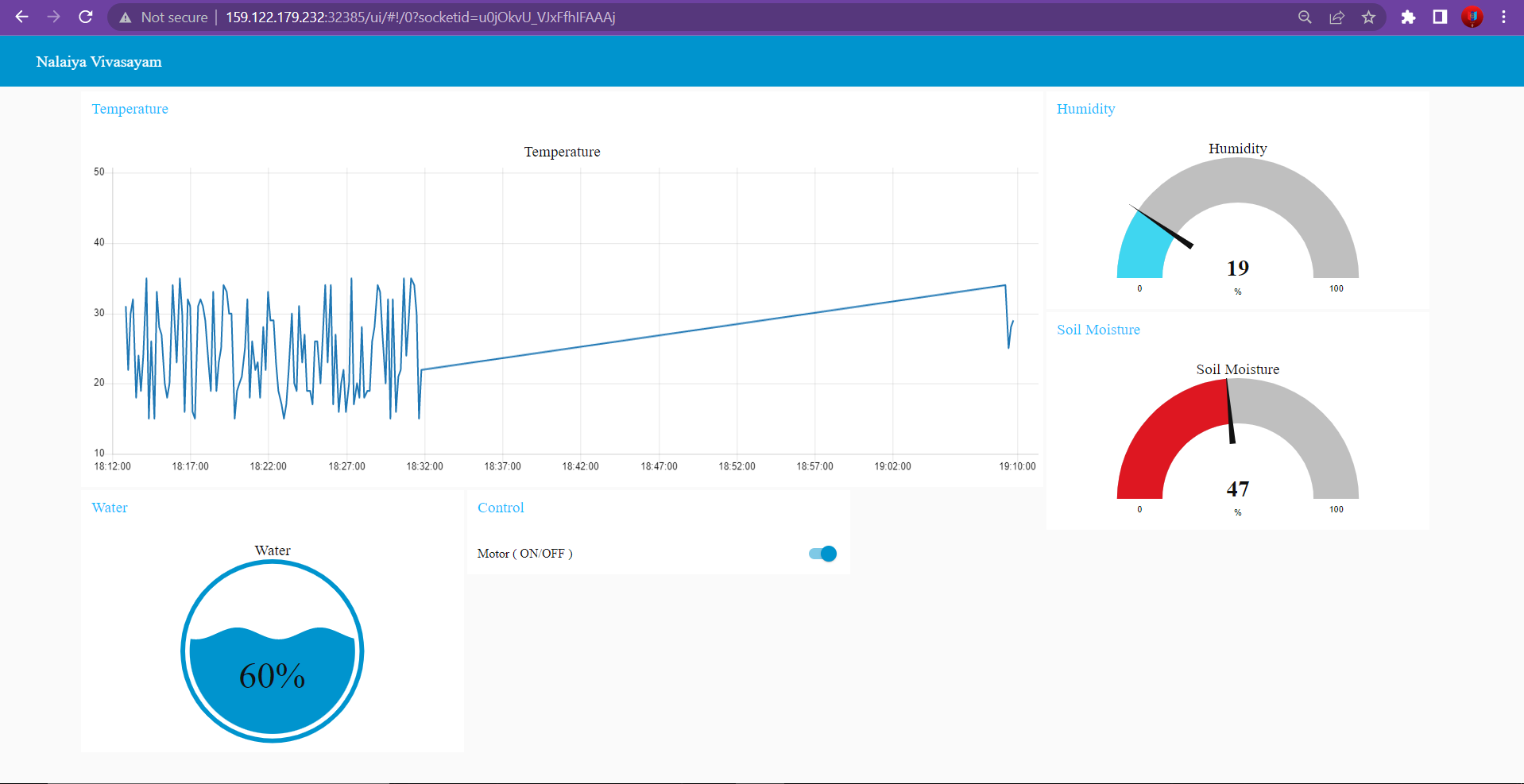
        print("Not connected to IoTF")

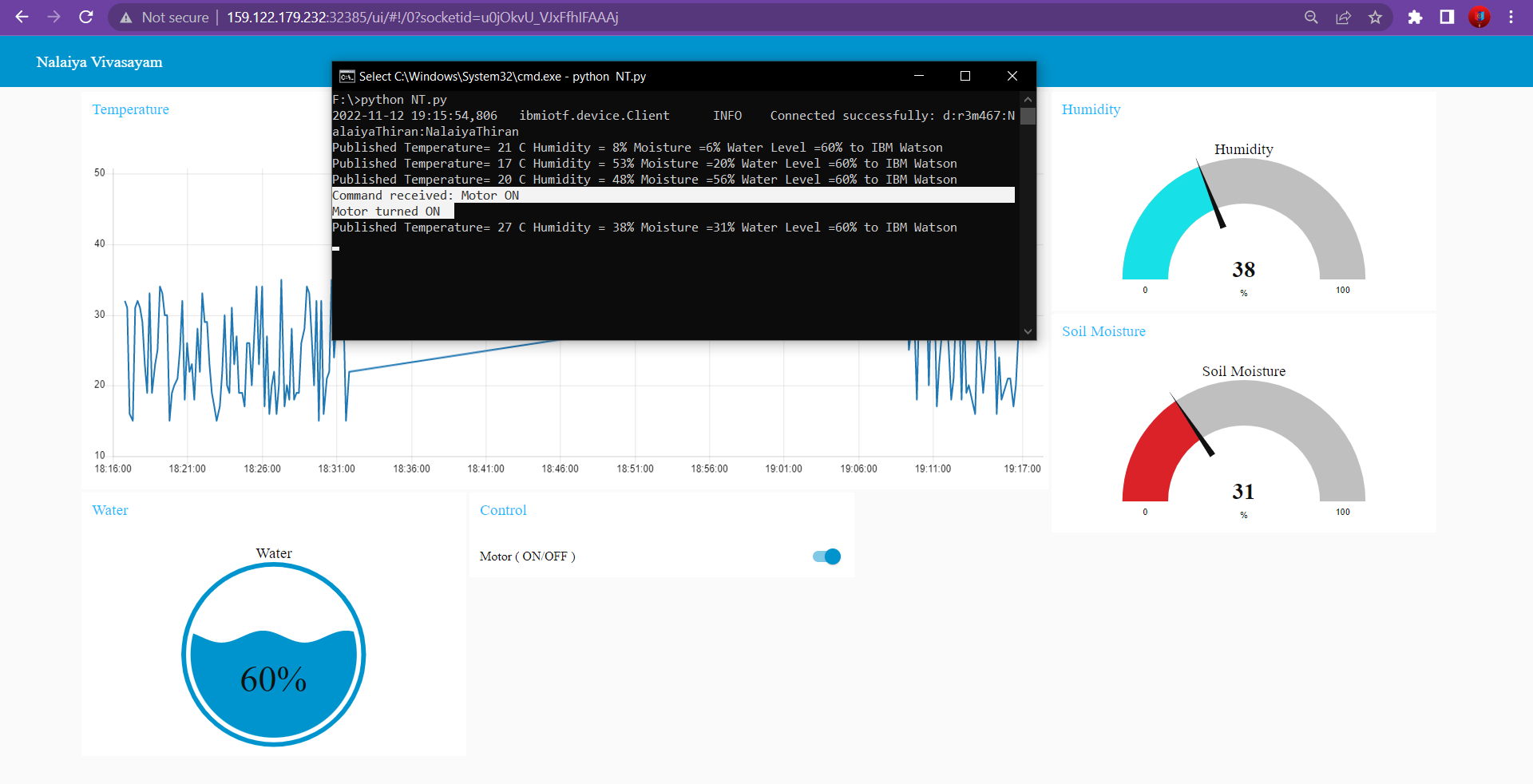
# Disconnect the device and application from the cloud

deviceCli.disconnect()

Output: 

Solution:





* 1. Feature 2

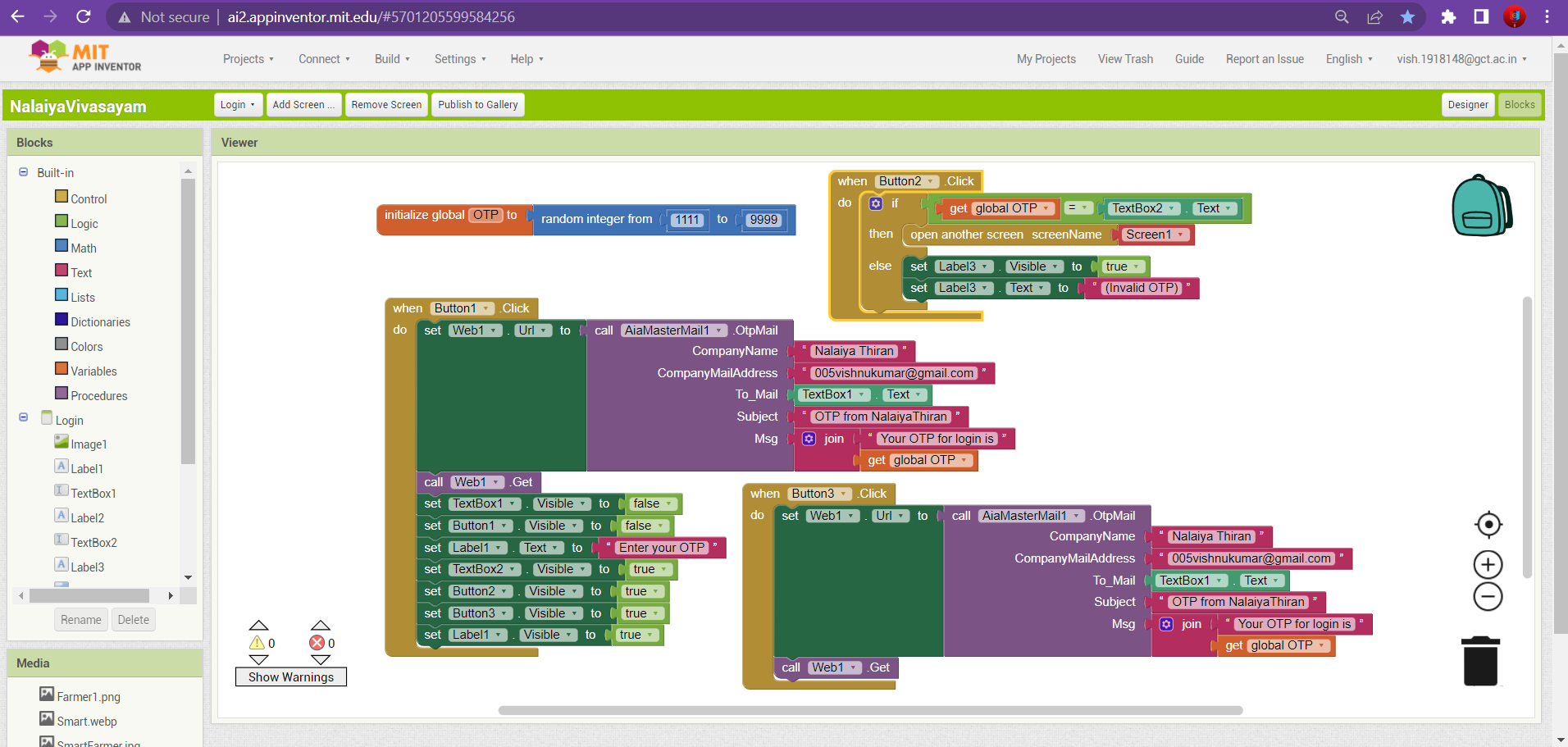
MOBILE APPLICATION:

* Login Page
* Main Screen

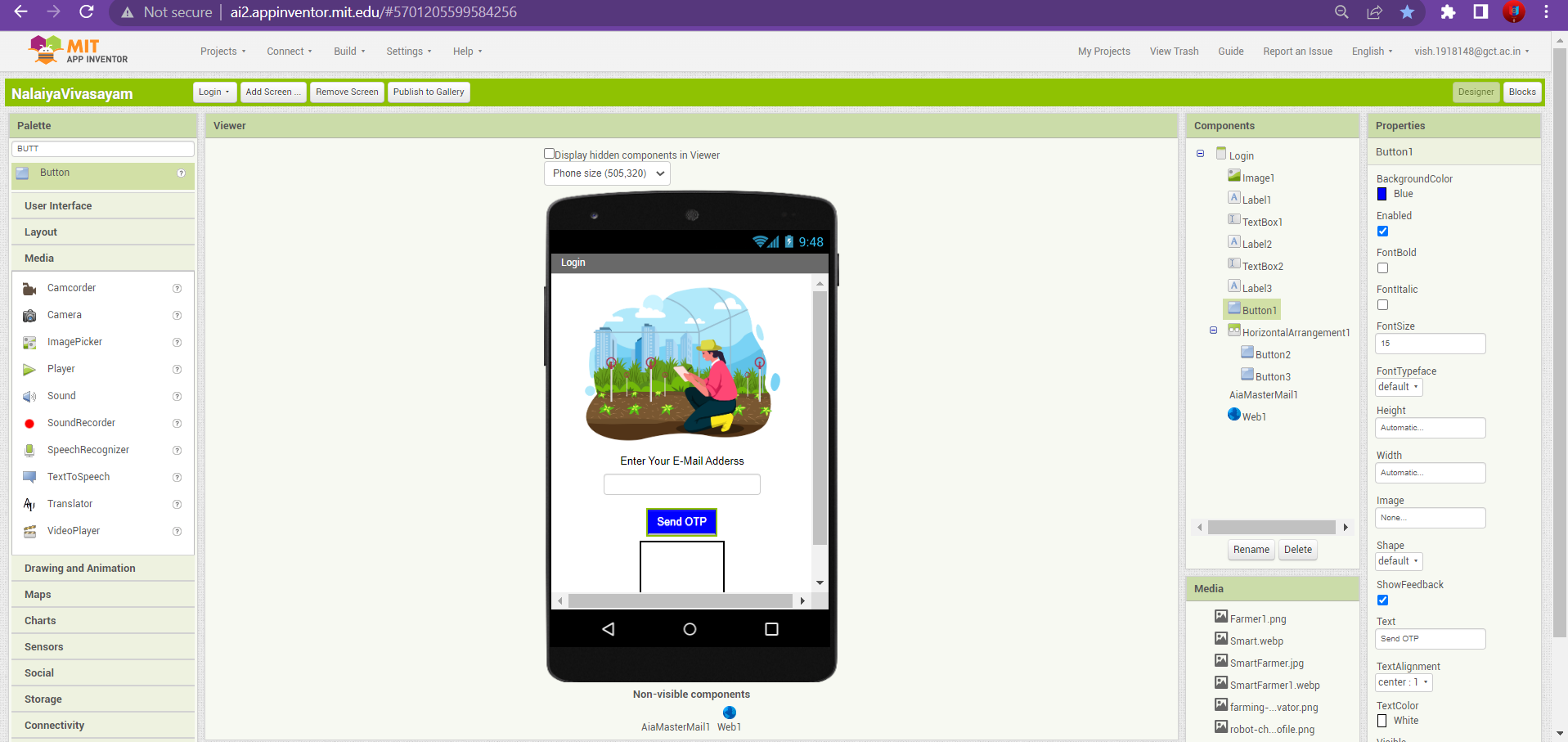
Login Page

* User can use their E-Mail address to login into the app and then they can access their resource.
* If user enters the e-mail, then OTP will be sent to the entered e-mail
* If user enters the wrong OTP, then it alerts with the Invalid OTP message
* If user forgets the OTP in short time, they can make use RESEND OTP button
* On successful OTP verification, User can navigate to the main screen

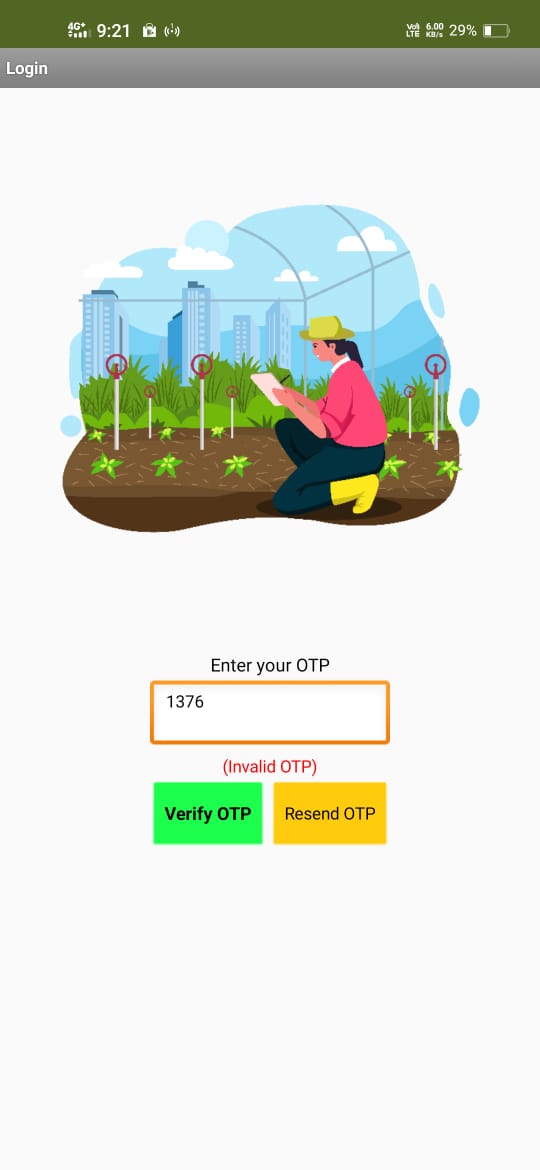
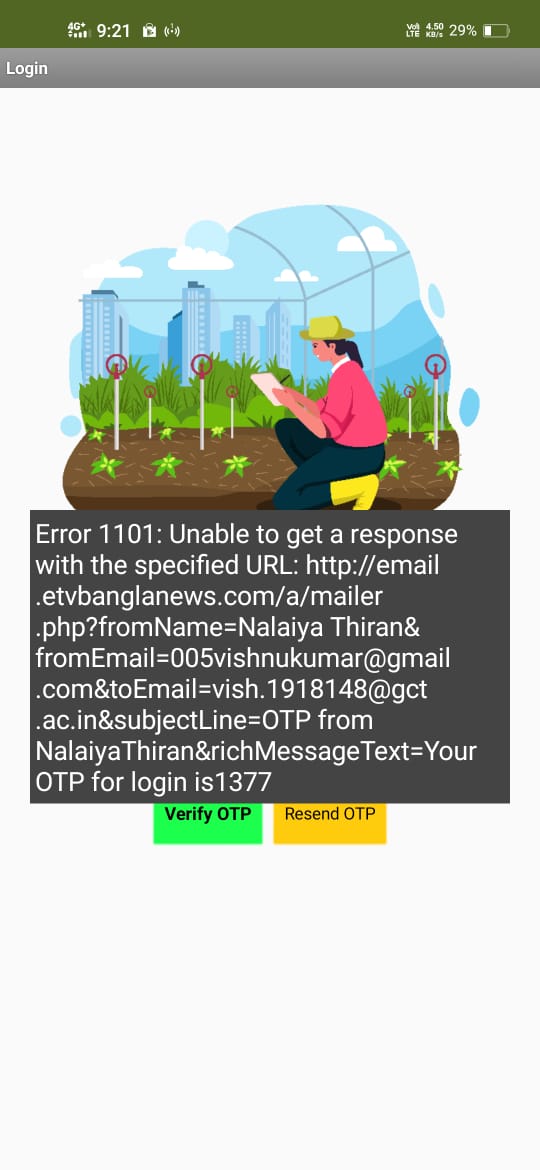
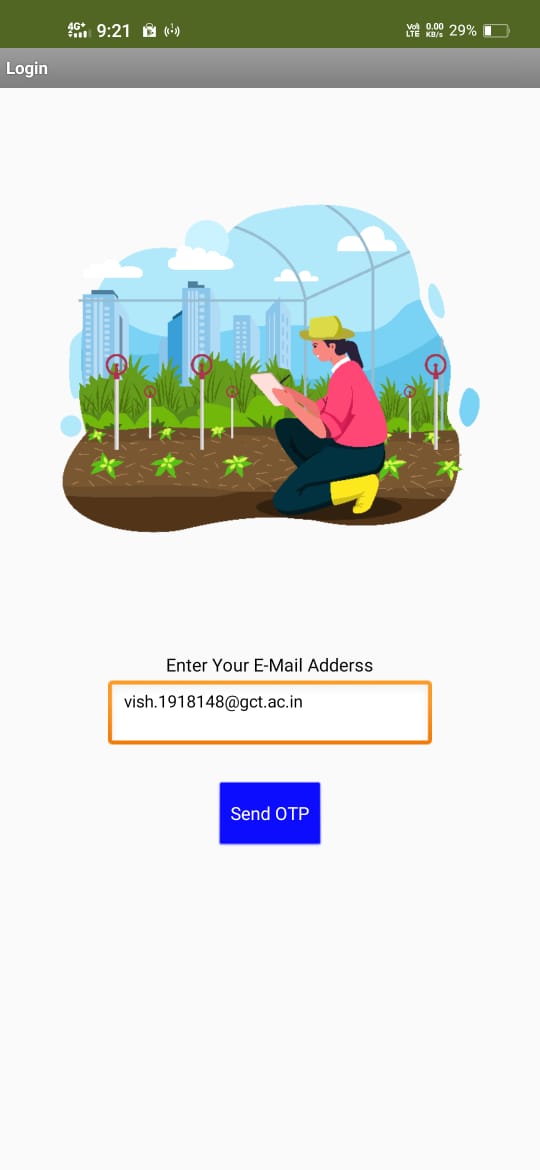
Block Coding



Design



Output



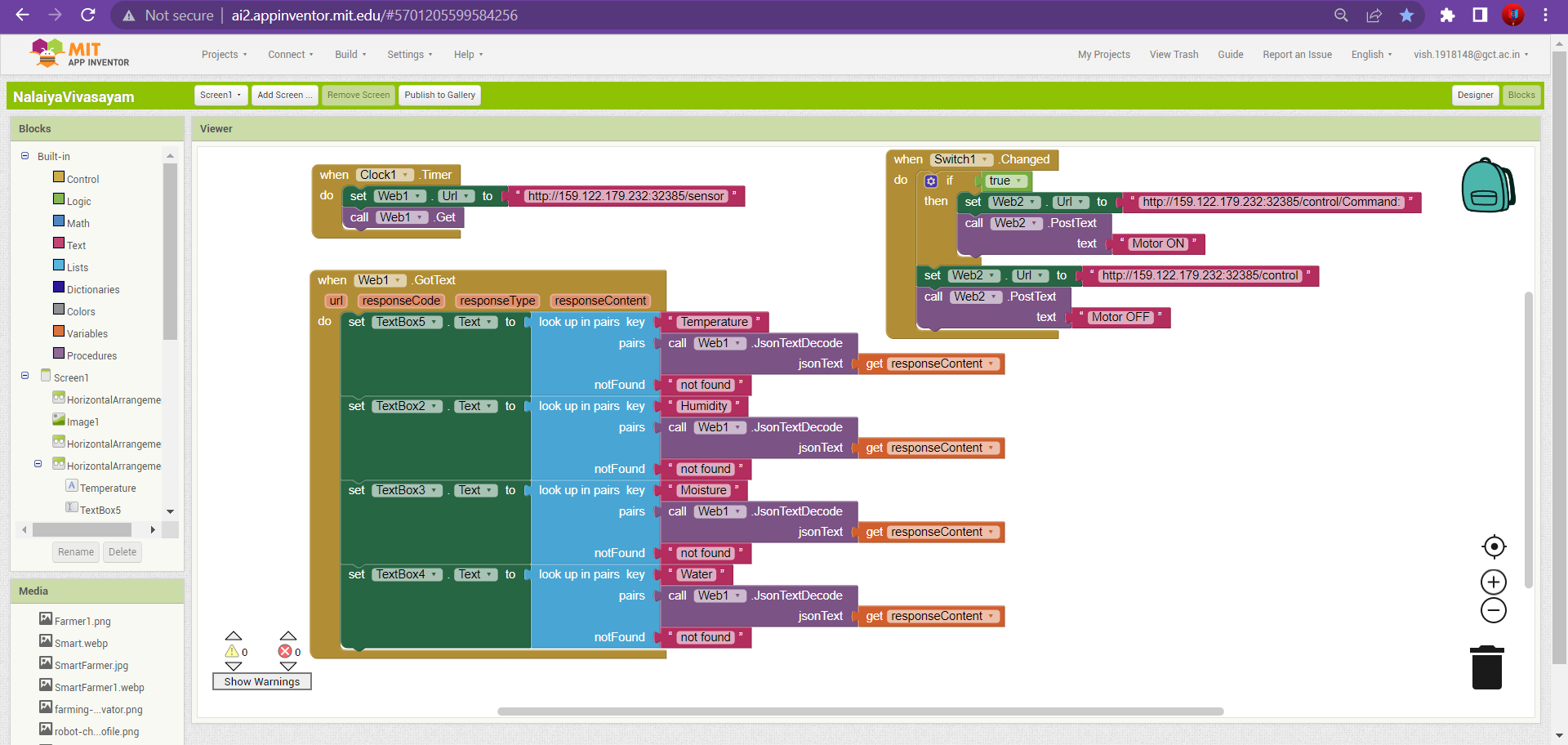
Main Screen

When user verification is successful then they will navigate to this main screen

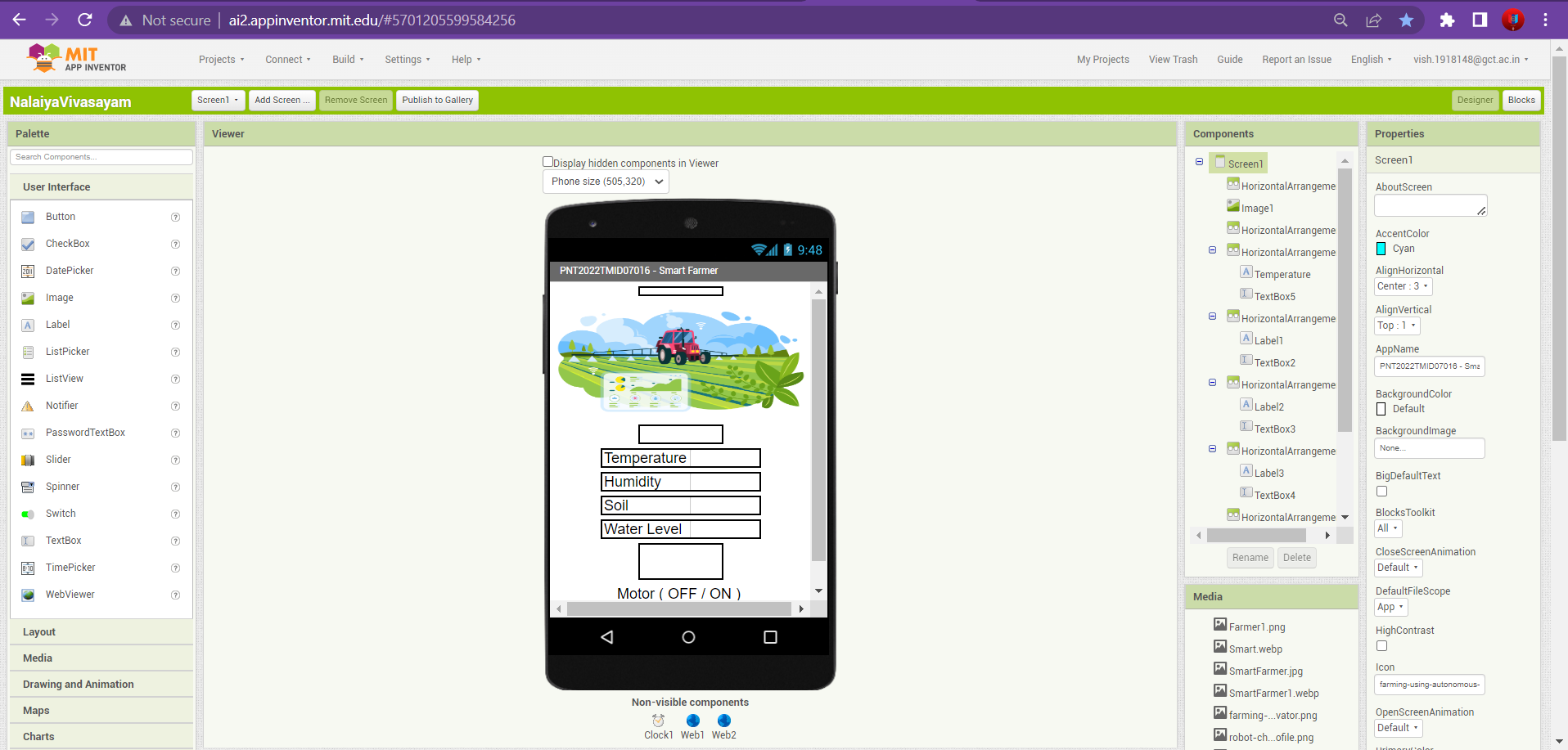
User can view the data of the sensor which are fetched from the IBM Watson IoT Platform

When user turns on the switch button, they can control the motor ON / OFF action

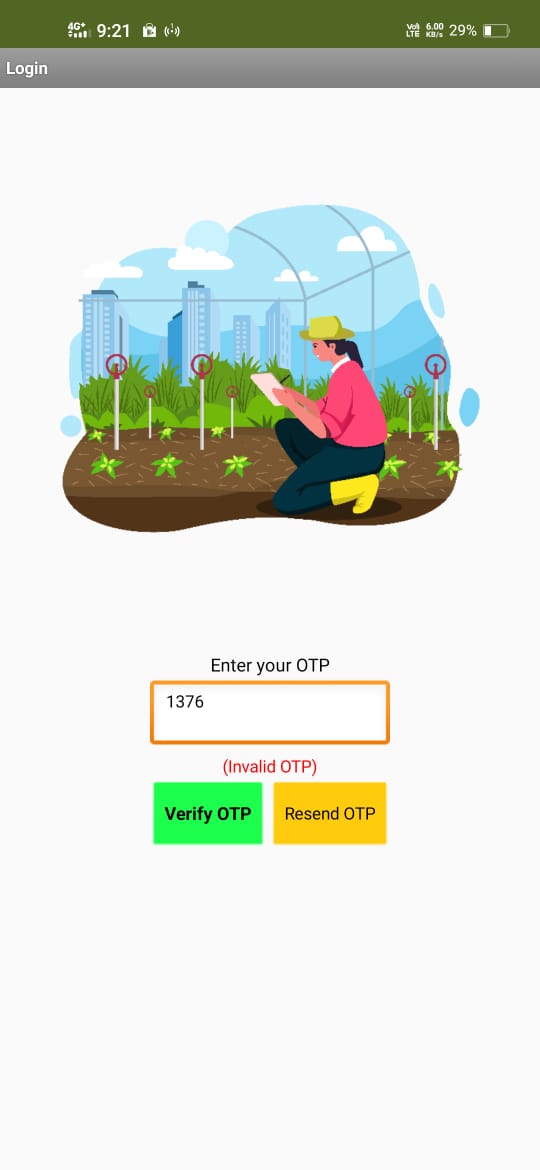
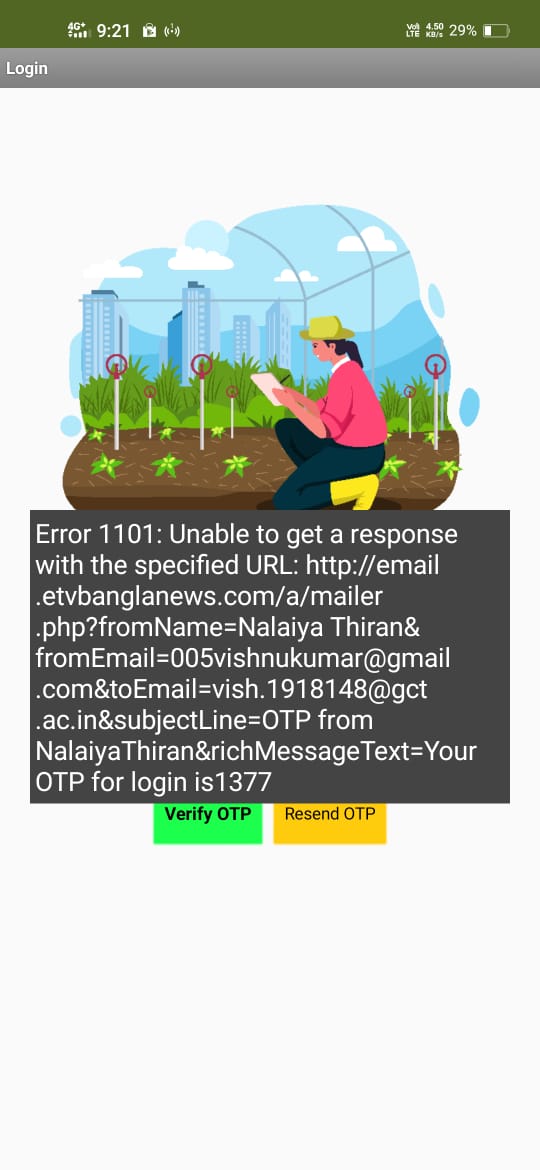
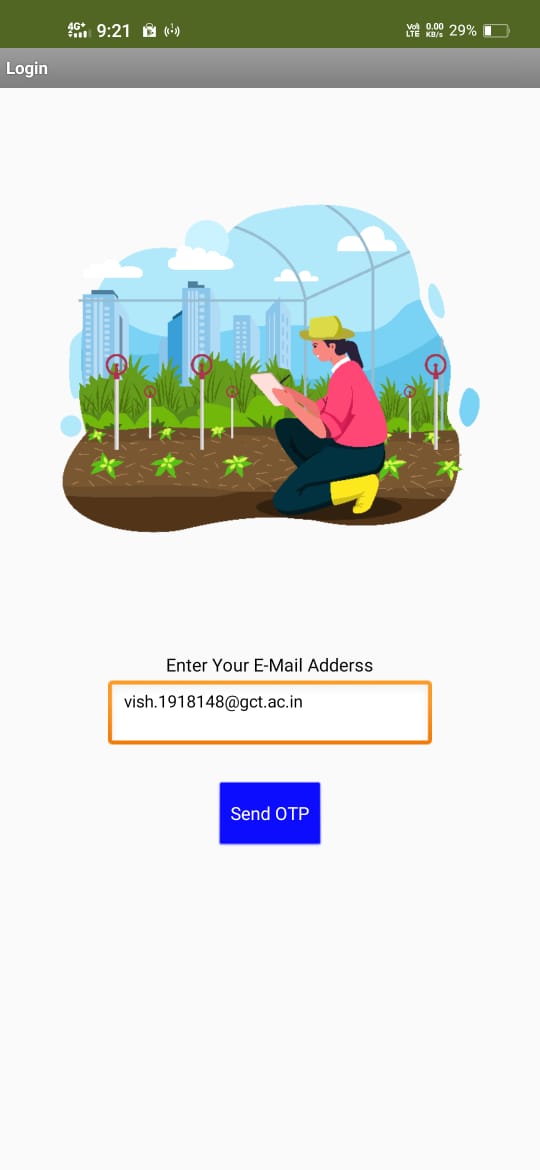
Blocks



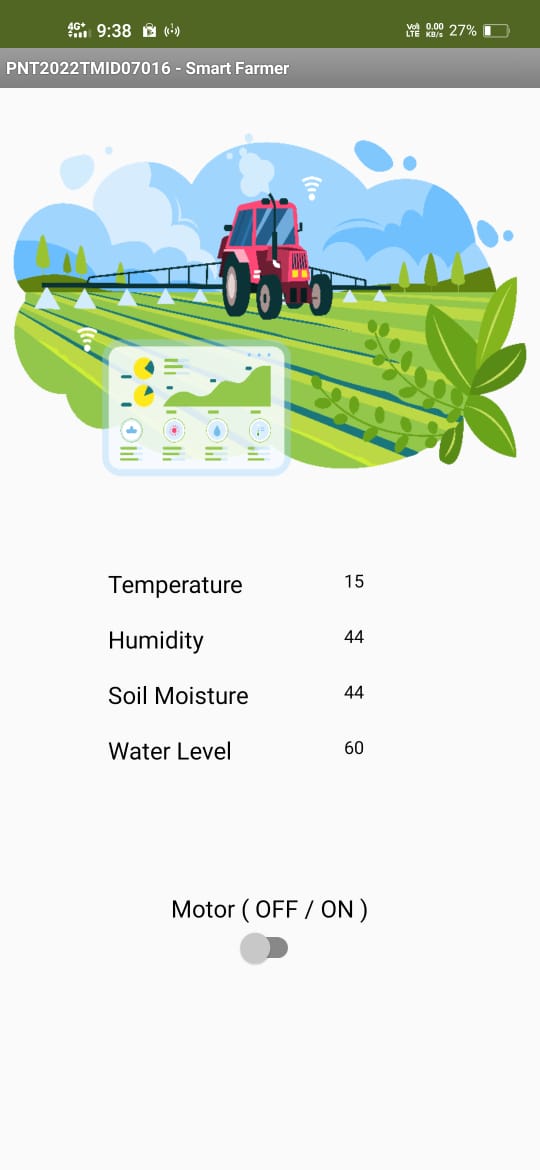
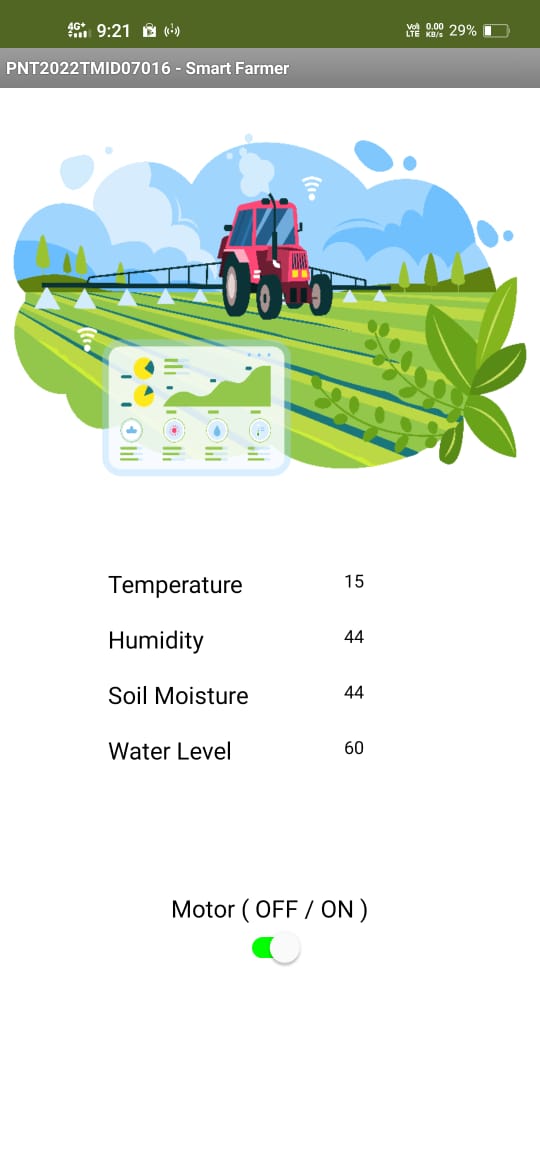
Design



1. **TESTING**
   1. Test Cases



* 1. User Acceptance Testing



1. **RESULTS**
   1. Performance Metrics

Performance metrices of the developed project is high

1. **ADVANTAGES & DISADVANTAGES**

Advantages:

Discourage weeds, saves water and time, statistical data can be used to control diseases and fungal growth, simplest model.

* Farms can be monitored and controlled remotely.
* Increase in convenience to farmers.
* Less labor cost.

Disadvantages:

This system is just limited to the automation of irrigation system and lacks in extra ordinary features

* Lack of internet/connectivity issues.
* Added cost of internet and internet gateway infrastructure.

1. **CONCLUSION**

Thus, the objective of the project to implement an IOT system in order to help farmers to control and monitor their farms has been implemented successfully.

The system supports water management decision, used for monitoring the whole system with GSM(RS-232) module.

● The system continuously monitors the water level (Water level Sensor) in the tank and provide accurate amount of water required to the plant or tree (crop).

● The system checks the temperature, and humidity of soil to retain the nutrient composition of the soil managed for proper growth of plant.

● Low cost and effective with less power consumption using sensors for remote monitoring and controlling devices which are controlled via SMS using a GSM using android mobile

1. **FUTURE SCOPE**

Irrigation is a process of providing the desire amounts of water to the agricultural land. This process is very beneficial in minimizing runoffs or drought situations for the croup’s cultivation.

Due to alarming changes in the climate, farmers cannot rely on natural rainwater.

Irrigation is important to yield good quality crops in the seasonable or non-seasonable period.

For [modern agriculture](https://dipslab.com/precision-farming/), a smart irrigation system is one of the best techniques that give more production in minimum duration. To many extend, this smart irrigation system is designed and fully automated to minimize manual handling in agriculture.

And one of the good things is that it is very comfortable for users (or farmers) to understand the concept of IoT and [sensors](https://dipslab.com/sensor-types/) for smart irrigation.

It can help you to learn how various sensors can be deployed and utilization of their data to generate events and control irrigation systems.

1. **APPENDIX**

Source Code

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "r3m467"

deviceType = "NalaiyaThiran"

deviceId = "NalaiyaThiran"

authMethod = "token"

authToken = "NalaiyaThiran"

# Initialize GPIO

def myCommandCallback(cmd):

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

    status=cmd.data['Command']

    if status=="Motor ON":

        print ("Motor turned ON")

    elif status == "Motor OFF":

        print ("Motor turned OFF")

    else :

        print ("Invalid Command")

try:

    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-method": authMethod,"auth-token":authToken}

    deviceCli = ibmiotf.device.Client(deviceOptions)

    deviceCli.connect()

#..............................................

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

while True:

    #Get Sensor Data from DHT11

    temp=random.randint(15,35)

    Mois,Humid=random.randint(200,60000)//1000,random.randint(200,60000)//1000

    Water = 60

    time.sleep(10)

    data = { 'Temperature' : temp, 'Humidity': Humid, 'Moisture' :Mois, 'Water':Water}

    #print data

    def myOnPublishCallback():

        print ("Published Temperature= %s C" % temp, "Humidity = %s%%" % Humid, "Moisture =%s%%" %Mois,"Water Level =%s%%" %Water, "to IBM Watson")

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on\_publish=myOnPublishCallback)

    deviceCli.commandCallback = myCommandCallback

    if not success:

        print("Not connected to IoTF")

# Disconnect the device and application from the cloud

deviceCli.disconnect()

GitHub & Project Demo Link

GitHub Link

<https://github.com/IBM-EPBL/IBM-Project-842-1658325386>

Project Demo Link

<https://drive.google.com/drive/folders/19mk1rxMbYHqfmXohwh9S9X9WhqmmGE4w?usp=share_link>