|  |  |
| --- | --- |
| PROJECT DEVELOPMENT PHASE | |
| SPRINT DELIVERY – 4 | |
| TEAM ID | PNT2022TMID07016 |
| PROJECT NAME | Smart Farmer - IoT Enabled Smart Farming Monitoring Application |

Python program connects with the IBM cloud

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)

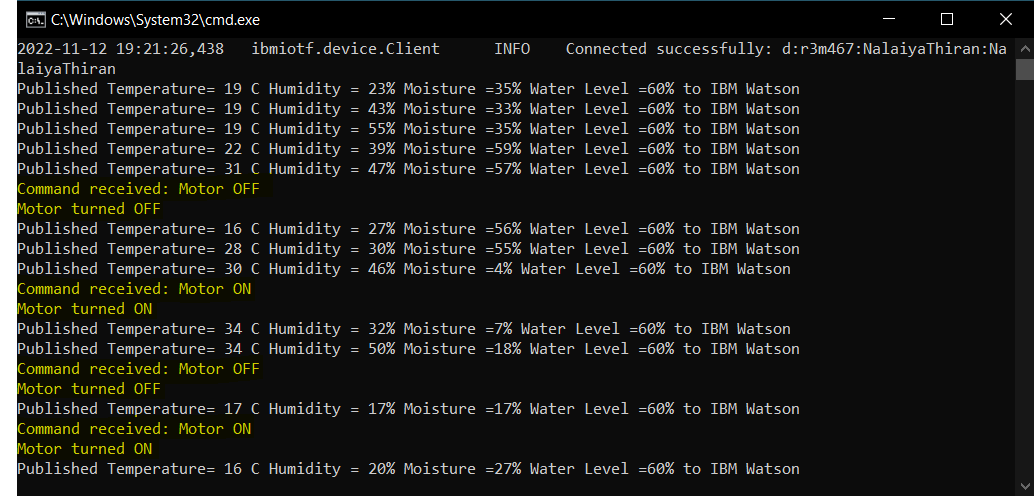
    deviceCli.commandCallback = myCommandCallback

    if not success:

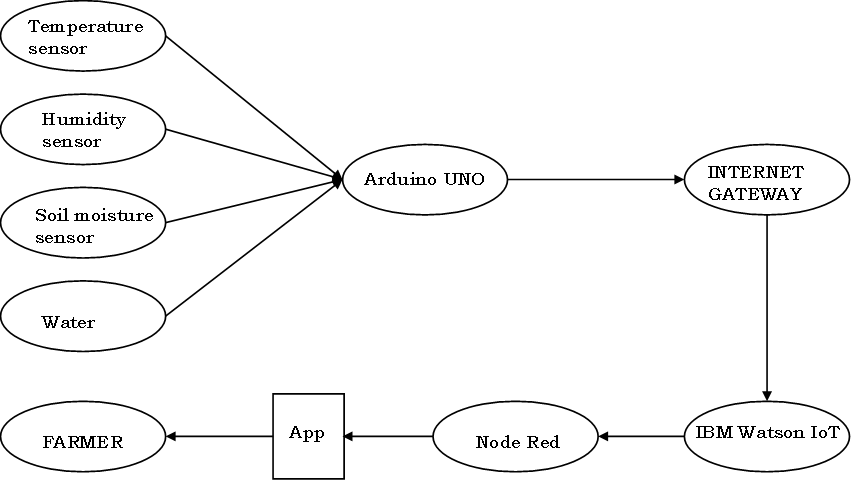
        print("Not connected to IoTF")

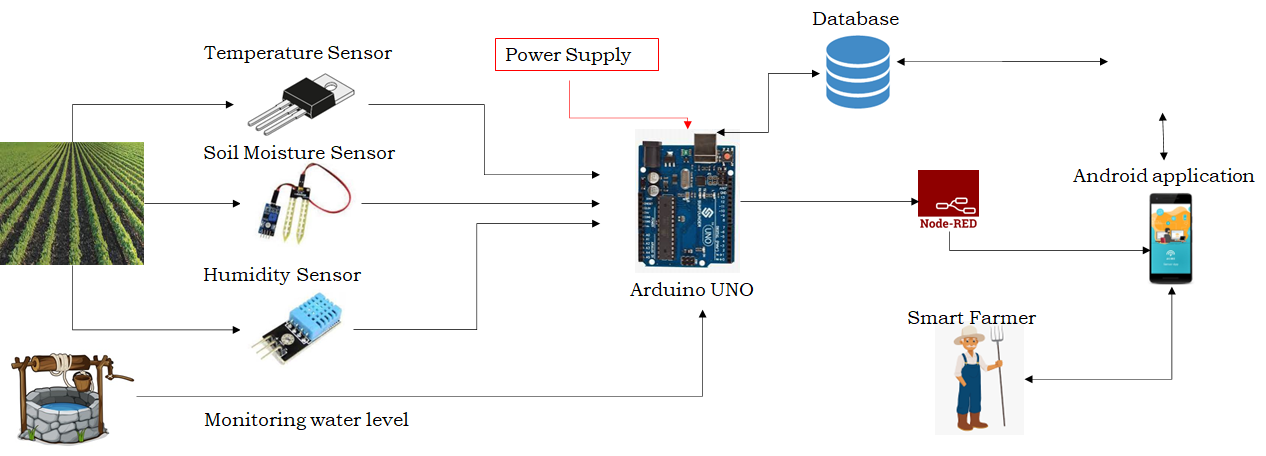
# Disconnect the device and application from the cloud

deviceCli.disconnect()

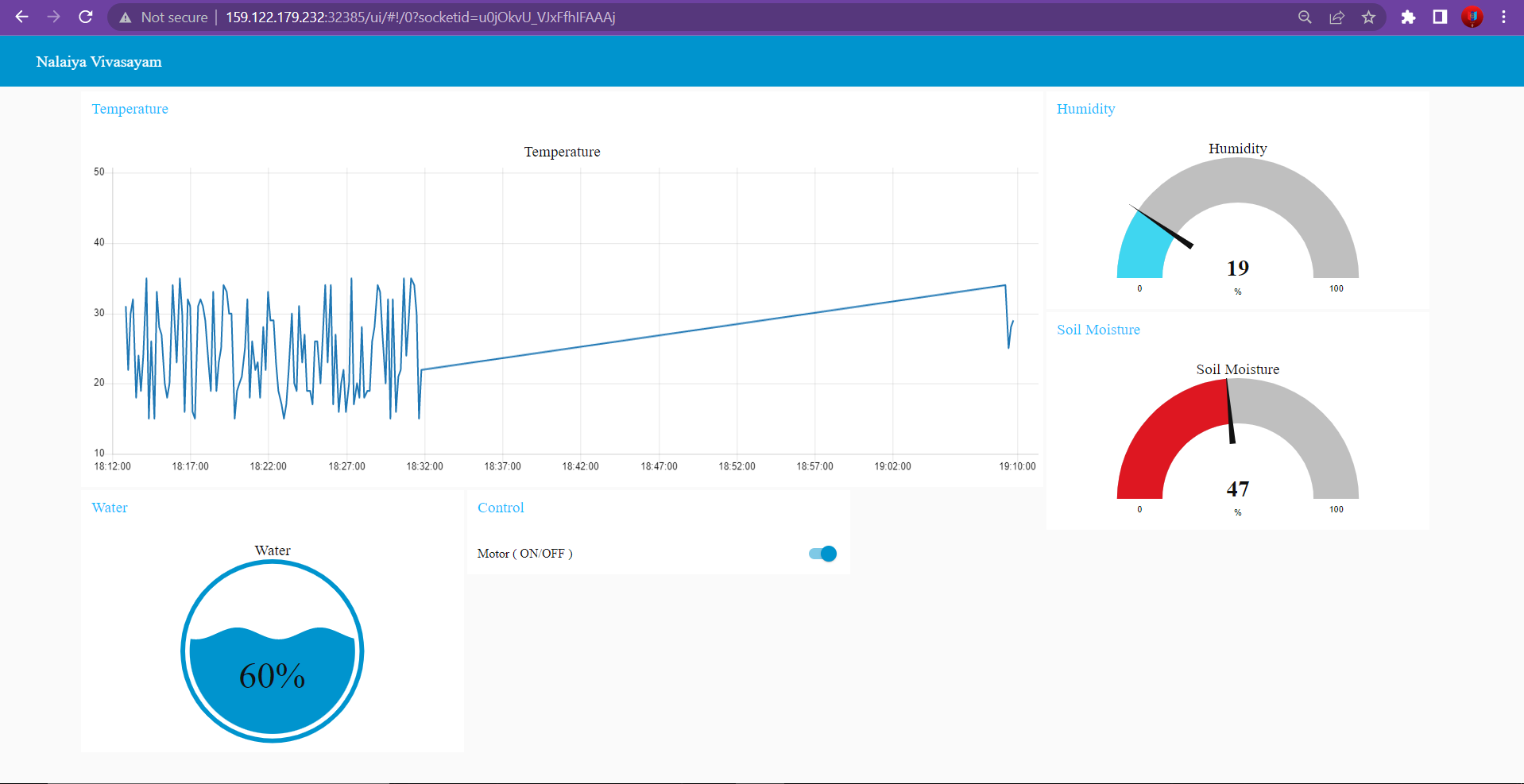


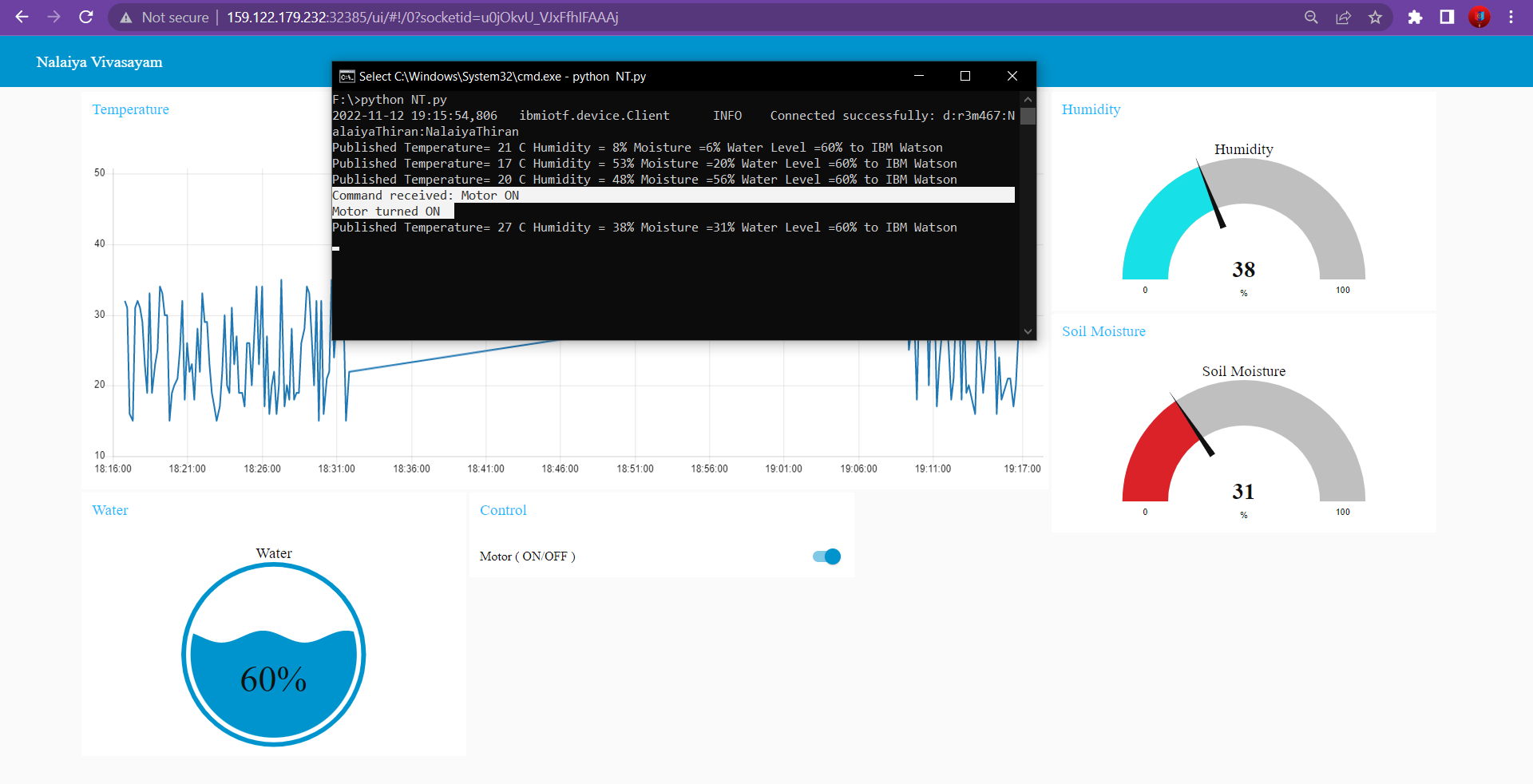
Flow Chart





Observations & Results





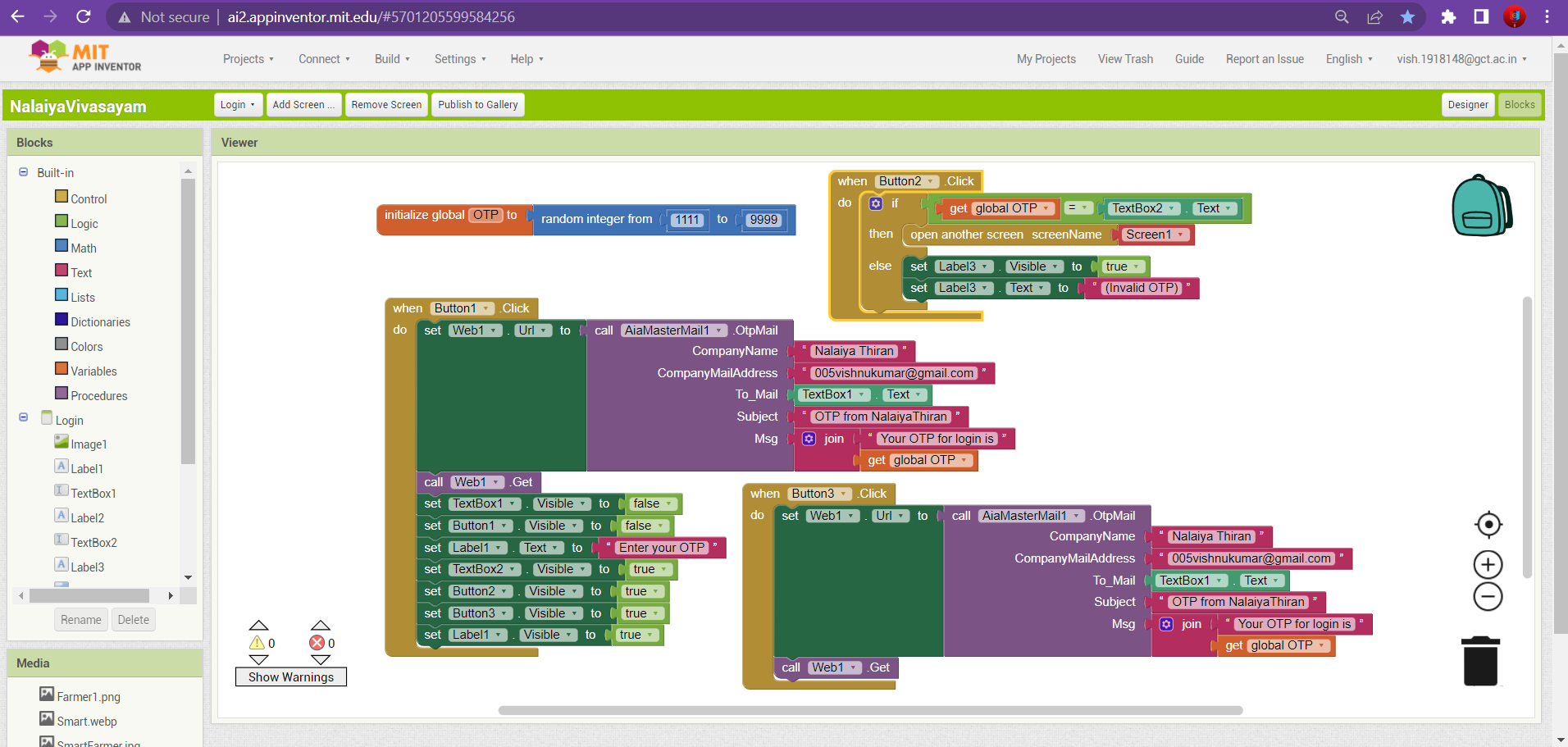
MOBILE APPLICATION:

* Login Page
* Main Screen

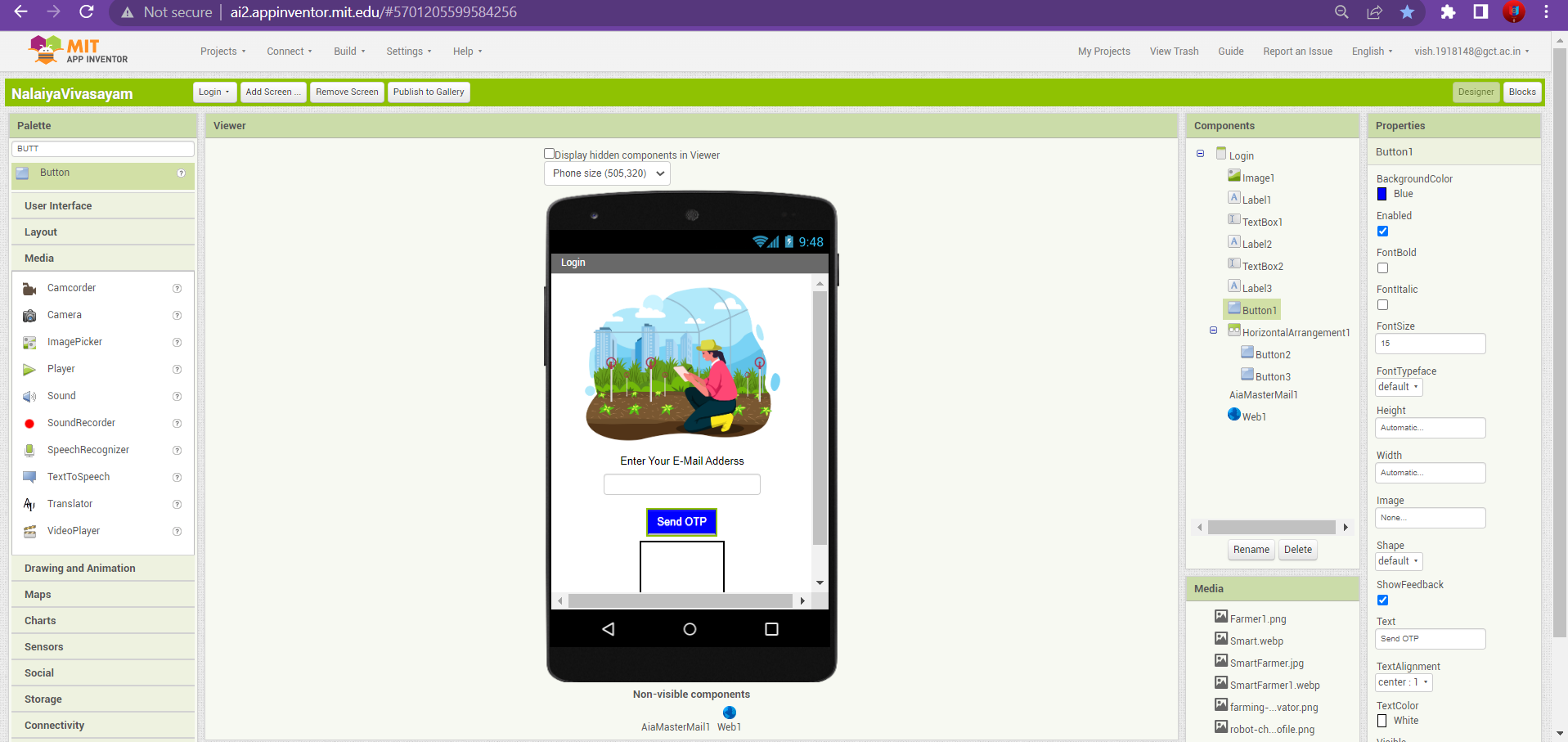
Login Page

* 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

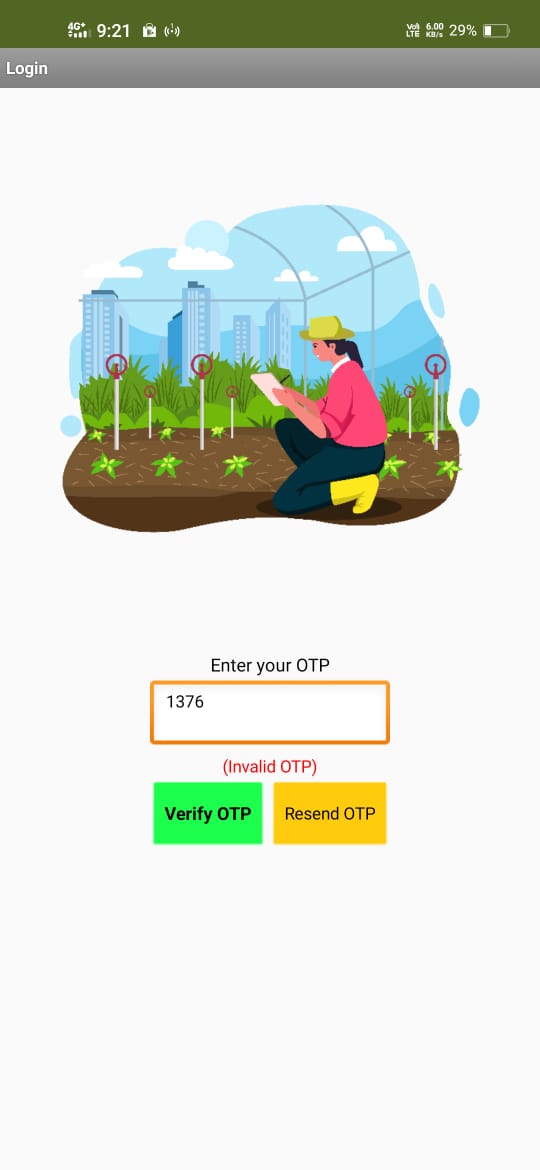
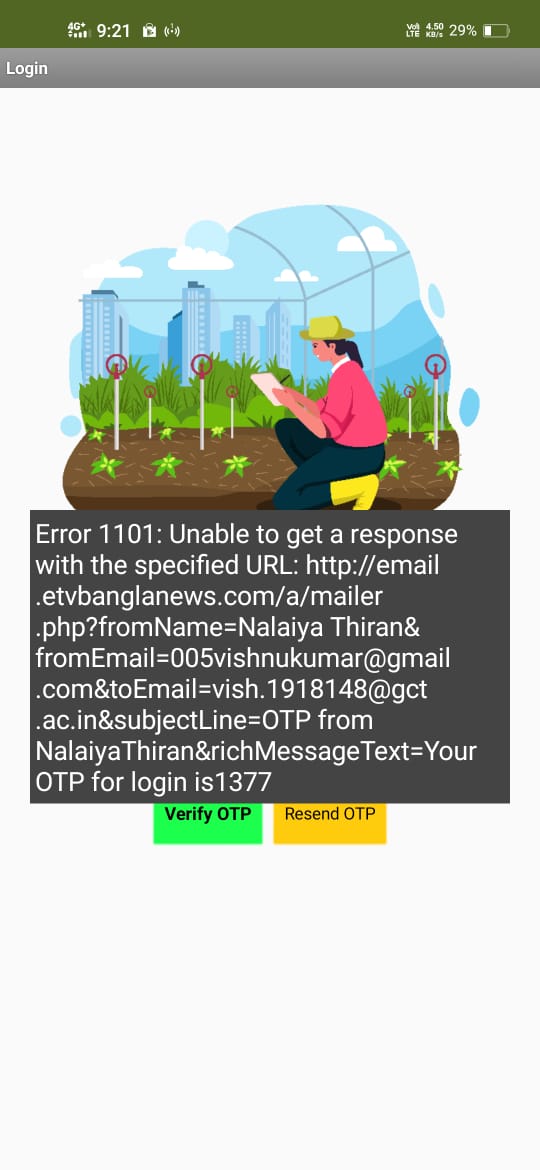
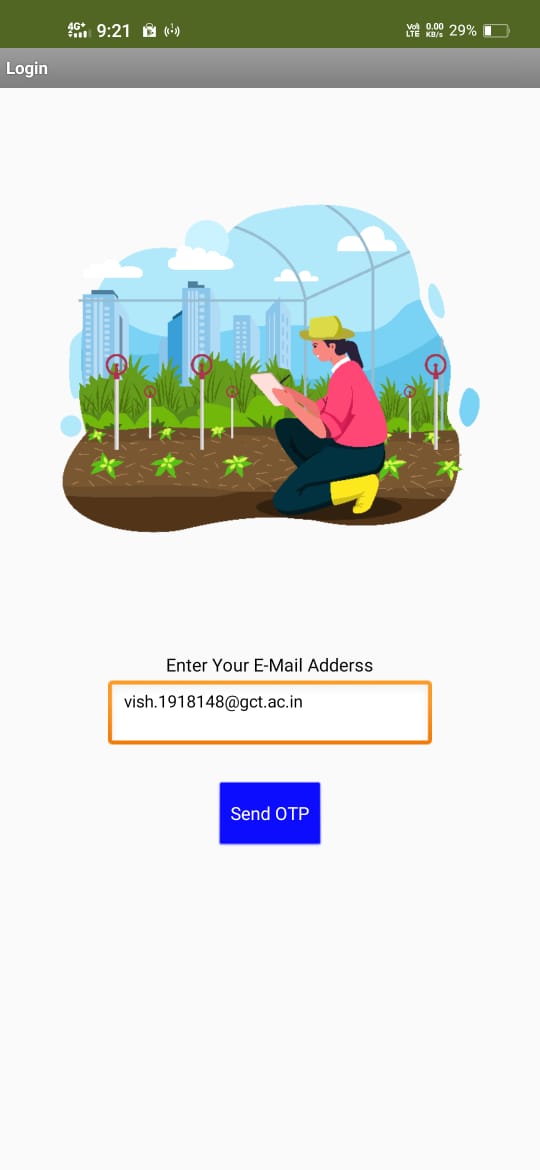
Blocks



Design



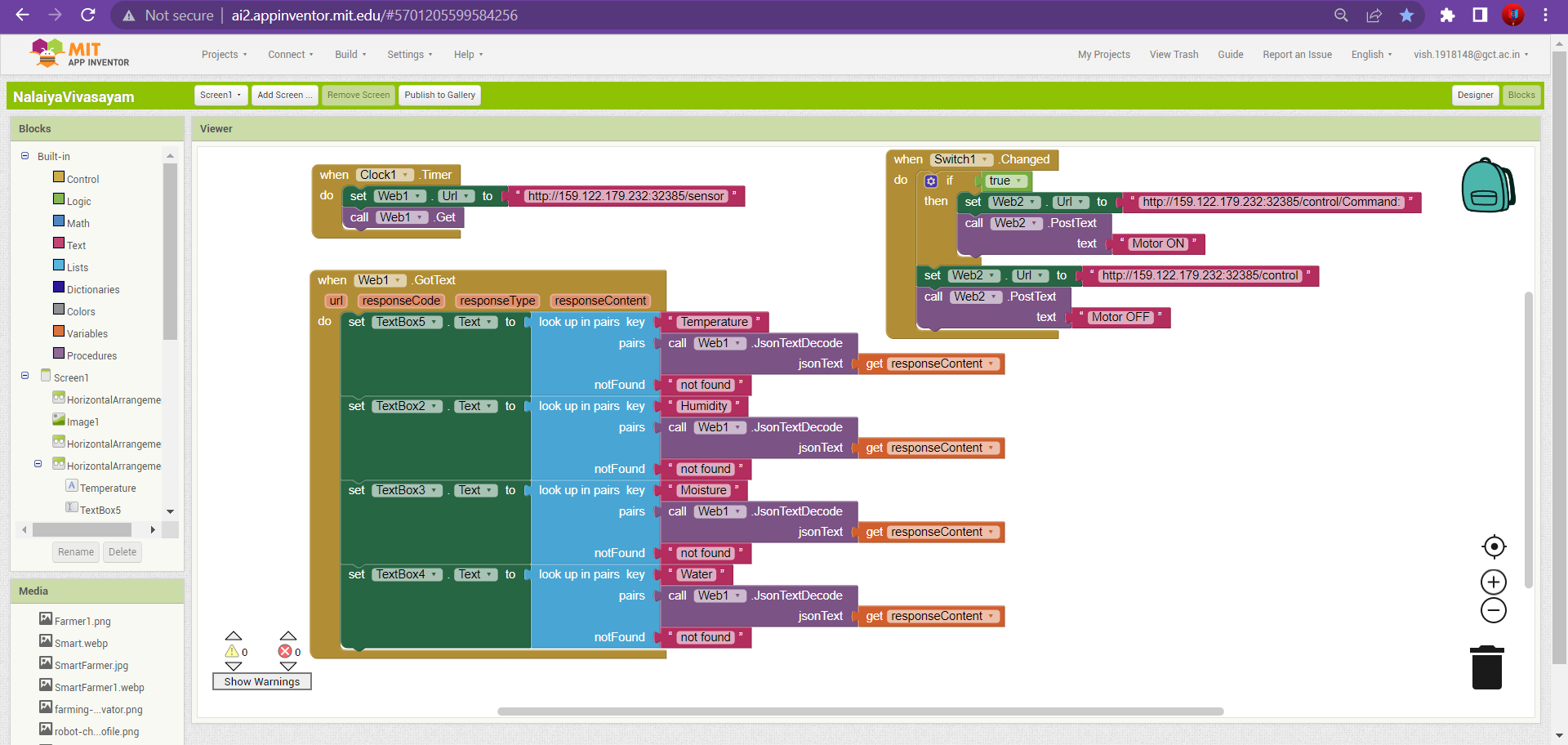
Output



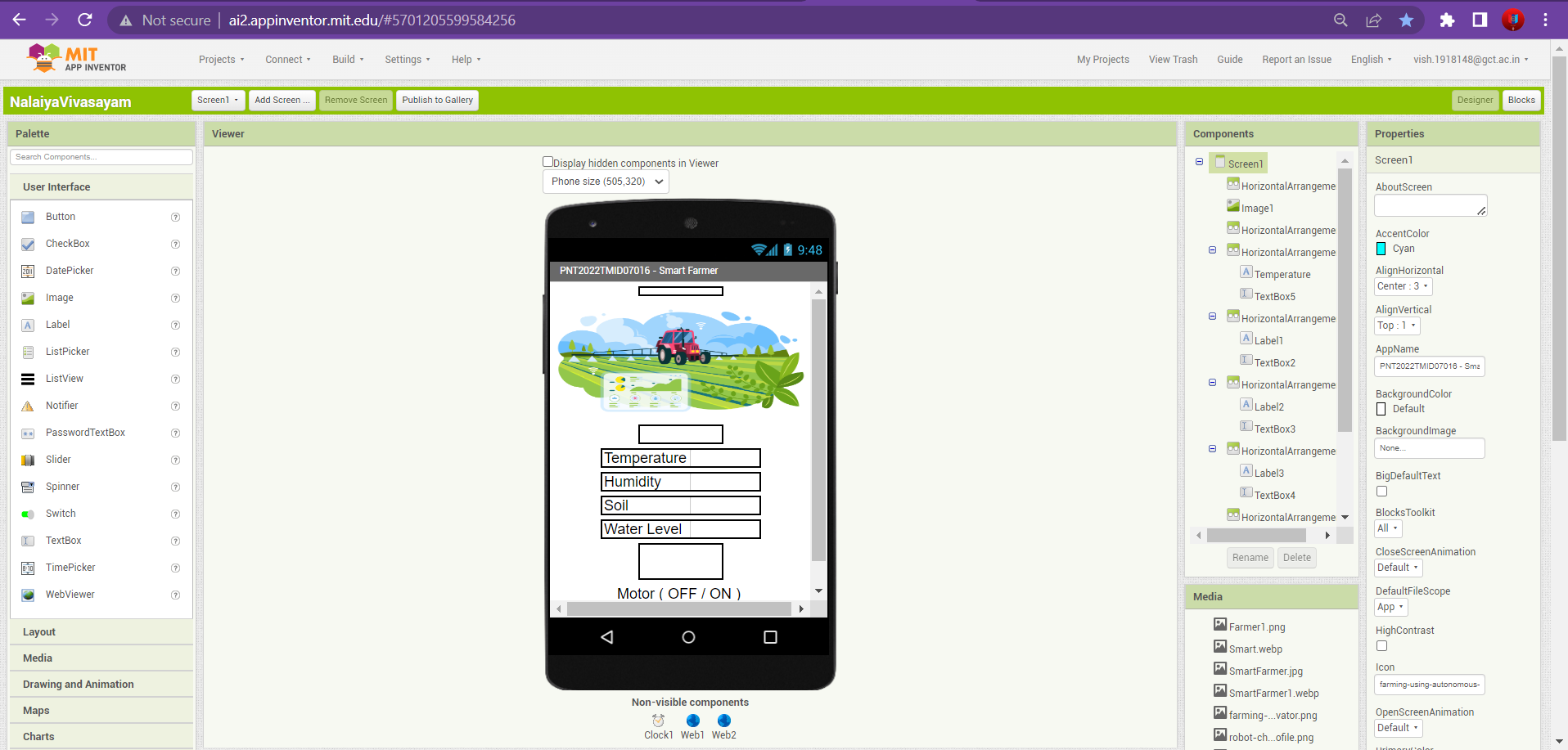
Main Screen

* 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

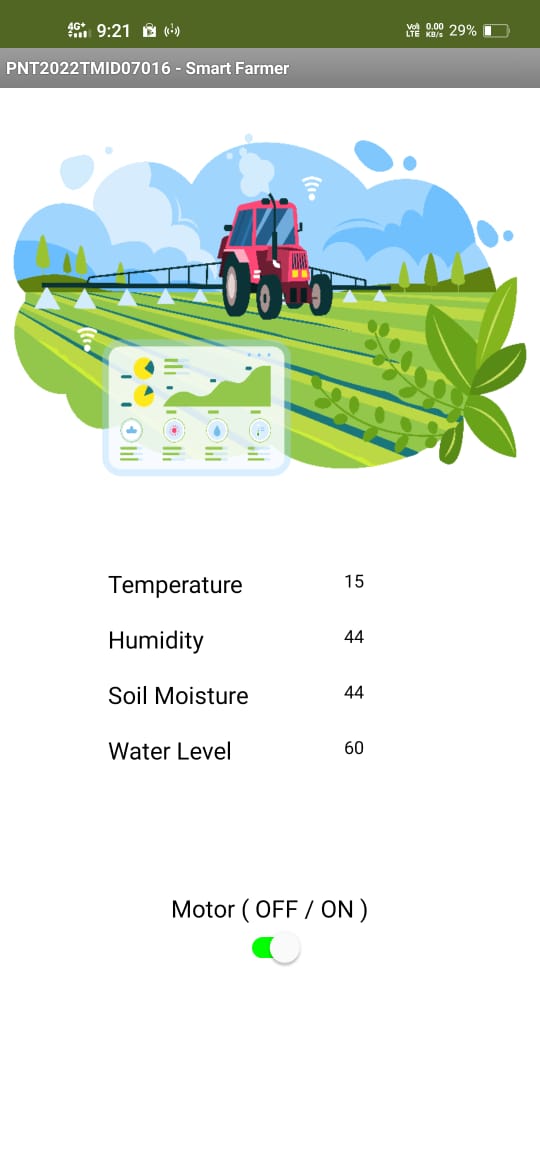
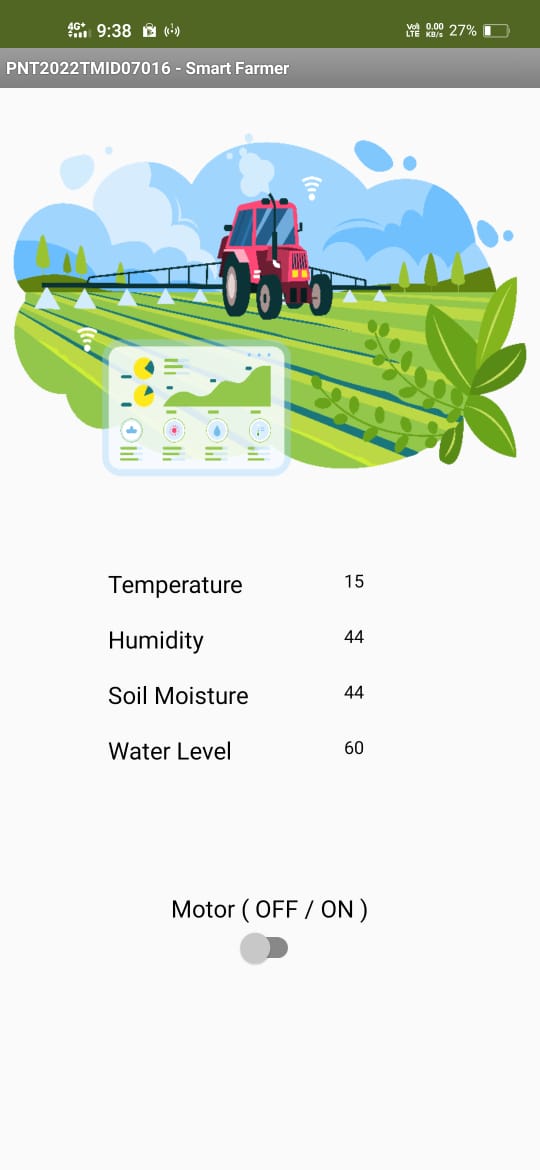
Blocks



Design



Output

Advantages:

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

Disadvantages:

* Lack of internet/connectivity issues.
* Added cost of internet and internet gateway infrastructure.
* Farmers wanted to adapt the use of Mobil App.

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.