SMART FARMING IOT ENABLED APPLICATION

IBM NALAIYA THIRAN

PROJECT REPORT

Submitted By

TEAM ID PNT2022TMID32108

TEAM LEAD	KARTHICK.J	(731619205021)
TEAM MEMBER 1	VIJAYARAGAVAN.A	(731619205060)
TEAM MEMBER 2	VEERASAKTHIVEL.T	(731619205057)
TEAM MEMBER 3	SRIDHARAN.S	(731619205047)
TEAM MEMBER 4	TAMILSELVAN.T	(731619205055)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

KSR INSTITUTE FOR ENGINEERING AND TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

CHAPTER-1

INTRODUCTION

1.1 PROJECT OVERVIEW

In recent times, attention to the Indoor plants in the houses has increased, which can be used to produce food or just for decoration and for health purposes. However, moving plants from their original place in nature to a closed place leads to negative effects due to changing environmental parameters around them. For instance, fluctuations in temperature, light, and soil moisture might affect indoor plant growth processes. This work investigates the possibility of using the smart mobile and web applications to monitor remotely the most of changing environmental parameters around plants. These parameters can give the user real-time information on air temperature and humidity, soil temperature and moisture, as well as amount of light. The Top-down method has been used to design a monitoring system to help the user keep informed of indoor climate changes. This system contains Raspberry and some sensors that are used in sensing various environmental conditions. It also includes the software component which defines services and actions to be taken on the data collected by sensing objects. The system is tested and evaluated in the indoor environment to prove the required concepts. The results indicated that web and mobile interfaces transfer data in real-time manner and send environmental information to the user. The data collected is visualized by different charts and figures to give a better understanding of the surrounding conditions in which the plant grows. It is concluded that the proposed system provides a user-friendly monitoring application to monitor the most indoor environmental parameters.

1.2 PURPOSE

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. Smart farming refers to managing farms using modern Information and communication technologies to increase the quantity and quality of products while optimizing the human labor required. Among the technologies available for present-day farmers are: Sensors: soil, water, light, humidity, temperature management.

Through smart farming though, larger food production is possible because the production can take place during the whole year without interruptions, while weather conditions in many cases have no impact on it. Apart from a possible future food crisis that may arise from poor environmental management, another issue that arises is that of high pollution caused by the use of fossil fuels due to the transportations; it is estimated that the produced goods travel thousands of kilometers on average before they reach our table.

Most importantly, smart farming systems do not require much prior expert knowledge in order to start your own cultivation; it requires resources like a backyard or a rooftop, and a hands-on approach combined with the right skills. Nevertheless, in most countries throughout Europe, they are still in an infantile stage, even though it is rapidly developing during the last few years.

Especially concerning the project’s participating countries (Bulgaria, Greece, Romania, Turkey and Cyprus) where farming and fishing are two very important – yet not as modernized – industries, new smart farming methods cannot be located in many curricula of VET organizations or HEIs, nor are there many large-scale establishments, either for profit or non-profit.

This is why the partners have come together to establish the “Smart Farming 4.0 All” consortium, in order to conduct extensive desk and field research addressed to a wide variety of relevant stakeholders and professionals, to create a Smart Farming Handbook and disseminate it freely to everybody interested, and to develop specific training curricula that will be utilized to train experts from the participating organizations as “carriers” of innovation, in order to be able to spread and main stream this novel method back to their regions.

CHAPTER-2

LITERATURE SURVEY

2.1 EXISTING PROBLEM

Arduino is an open source computer hardware and software company, protecting a user community that preassembled form, or as do-it- yourself (DOI) kits. Arduino board designs use a variety of micro process controllers. The boards are equipped with sets of digital and analog input/output(I/O) pins that may be interfaced to various expansion boards(shields) and other circuits.

The boards feature serial communication interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers.

The micro-controllers are typically programmed using a dialect of features from the programming languages c and c++.

In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the processing language project.

In an industry during certain hazards it will be very difficult to monitor the parameter through wires and analog devices such as transducers.

To overcome this problem, we use wireless devices to monitor the parameters so that ARDUINO UART ADC Temperature Sensor LDR SENSOR Contrast Soil Humidity Sensor Water Pump IOT can take certain steps even in the worst case. Few years' back the use of wireless devices was very less, but due the rapid development of technology now-a-days we use maximum of our data transfer through wireless like Wi-Fi, Bluetooth,WI-Max,etc.Project is designed as a plant monitoring system based on IOT. In this project we use different modules such as IOT,Arduinoas controller, Temperature sensor, Moisture sensor, Humidity sensor.

2.2 References

[1] **Smart Agriculture Using Internet of Things with Raspberry Pi** (Zuraida Muhammad, Muhammad Azri Asyraf Mohd Hafez, Nor Adni MatLeh, Zakiah Mohd Yusoff , Shabinar Abd Hamid),september 2020.

[2] **IoT based smart soil monitoring system for agricultural production** (Divya J., Divya M.,Janani V), February 2018.

[3] **Design and Optimization of IoT Based Smart Irrigation System in Sri Lanka**(H.G.C.R. Laksiri, H.A.C. Dharma Gunawardhana,J.V.Wijayakulasooriya),

April 2020.

[4] **A Literature Survey on Smart Agriculture Monitoring and Control System Using IOT**(Anushree Math, Layak Ali, Pruthviraj U), February 2022.

[5] **IoT based smart crop-field monitoring and automation irrigation system**(R. Nageswara Rao, B.Sridhar), June 2018.

2.3 Problem Statement Definition

The traditional agriculture and allied sector cannot meet the requirements of modern agriculture which requires high-yield, high quality and efficient output. Thus, it is very important to turn towards modernization of existing methods and using the information technology and data over a certain period to predict the best possible productivity and crop suitable on the very particular land.

The adoption of access to high-speed internet, mobile devices, and reliable, low-cost satellites (for imagery and positioning) are few key technologies characterizing the precision agriculture trend.

Precision agriculture is one of the most famous applications of IoT in the agricultural sector and numerous organizations are leveraging this technique around the world.

Some products and services in use are VRI optimization, soil moisture probes, virtual optimizer PRO, and so on. VRI (Variable Rate Irrigation) optimization maximizes profitability on irrigated crop fields with topography or soil variability, improves yields, and increases water use efficiency.

IoT has been making deep inroads into sectors such as manufacturing, health-care and automotive. When it comes to food production, transport and storage, it offers a breadth of options that can improve India's per capita food availability. Sensors that offer information on soil nutrient status, pest infestation, moisture conditions etc. which can be used to improve crop yields over time.

Some of the sample problem statements related to Agriculture & allied sectors where IoT application will be beneficial are given below

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

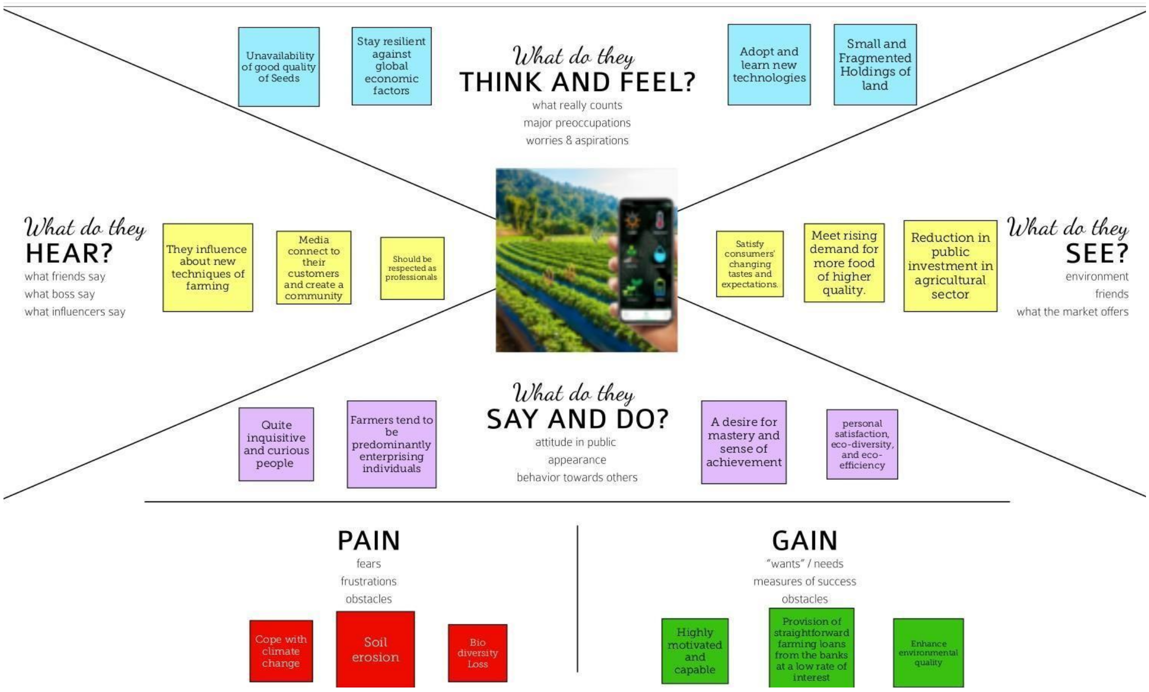

FIG 3.1.1

3.2 IDEATION & BRAINSTORMING

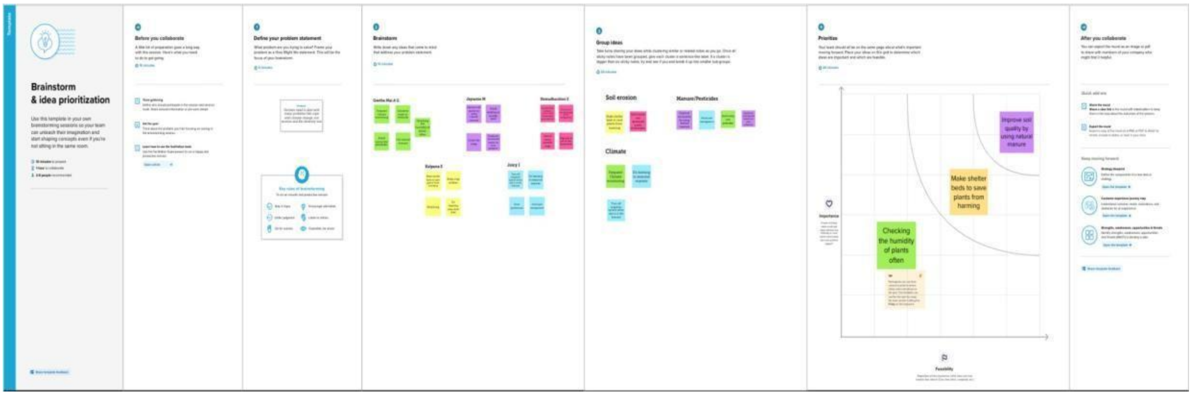

FIG 3.2.1

3.3 PROPOSED SOLUTION

S.No	Parameter	Description
1	Problem Statement (Problem to be solved)	Difficulty in selecting suitable crops for the soil.
2	Idea / Solution description	The crops strength will be maintained and the water level in the soil will be maintained.
3	Novelty / Uniqueness	The water level will be reduced in the soil means it will automatically pour the water.
4	Social Impact / Customer Satisfaction	It improves soil quality and soil fertility. It is a very time efficient process and gives more yields for the farmers.
5	Business Model (Revenue Model)	The project involves a NodeMCU and SOIL MOISTURE SENSOR which is cheaper than the existing ideas.
6	Scalability of the Solution	The sensor is responsible for checking the water level of the soil and the there is no overlap between area of various sensor.

3.4 PROBLEM SOLUTION

FIT CUSTOMER SEGMENT:

Public people

Farmer people

Industrialist people

CUSTOMER NEEDS:

User friendly device

Easy to operate

Need good service

CUSTOMER USE:

The customer will use this technology in a smart way and it will help to maintain the crops in a technological way and also maintain the water level of the crops and it will be user friendly to the farmer. It will reduce the work for farmers.

PROBLEM ROOT CAUSE:

The crop cannot maintain the water level regularly by the humans

The insufficient water will spoil the crops and affecting the farming

BEHAVIOR:

The crops can be maintained regularly.

JOBS TO BE DONE:

The farmer will notify the crops regularly.

The water level will low means it automatically pour the water.

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User login	Login the user ID and password for the use the application
FR-2	sensors	Using DHT11 and soil moisture sensor for maintaining the crops
FR-3	Monitoring details	This process gives the brief description about the crops.
FR-4	Settings	Change settings for the convenience

4.2 NON - FUNCTIONAL REQUIREMENTS

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

NFR No.	Non-Functional Requirement (Epic)	Description
NFR-1	Usability	Login the user ID and password for the use the application
NFR-2	Security	Using DHT11 and soil moisture sensor for maintaining the crops
NFR-3	Reliability	This process gives a brief description about the crops.
NFR-4	Performance	Change settings for the convenience
NFR-5	Availability	The user can easily access the analyzed data from the sensors connected with IoT devices which are placed in the farming land and the sensor analyzed data are stored in a cloud storage for future references.
NFR-6	Scalability	The application features are upgraded randomly for easy access and better user experience.

CHAPTER-5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

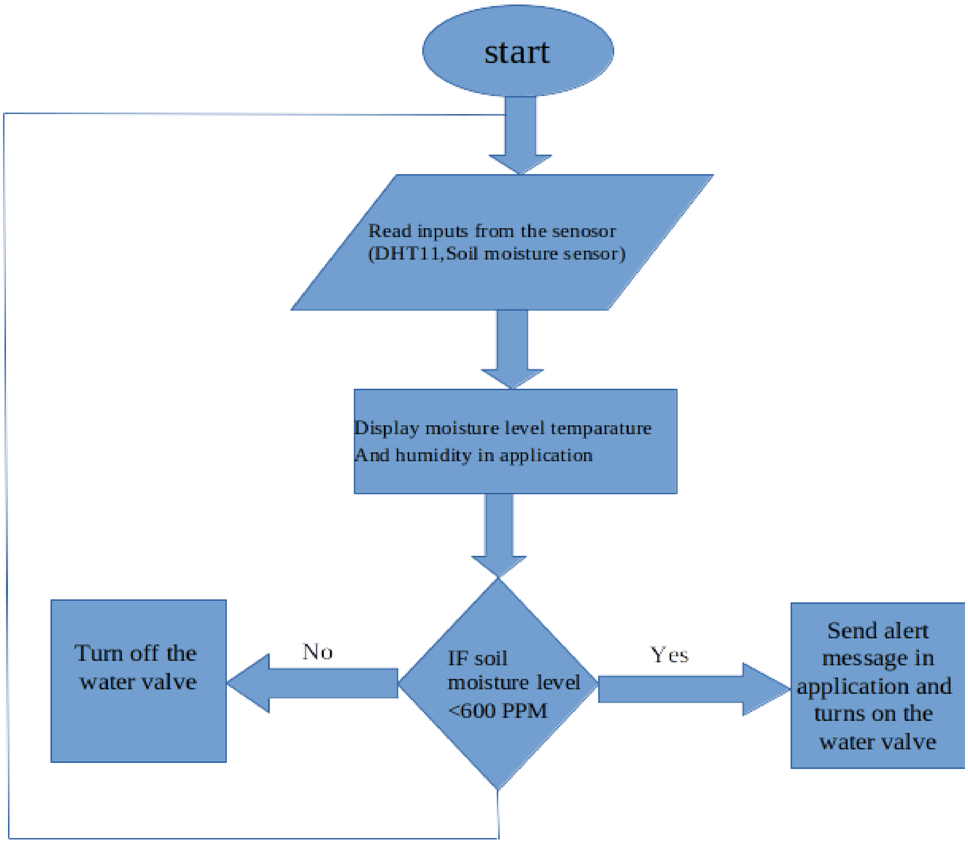

FIG 5.1.1

5.2 SOLUTION & TECHNICAL ARCHITECTURE

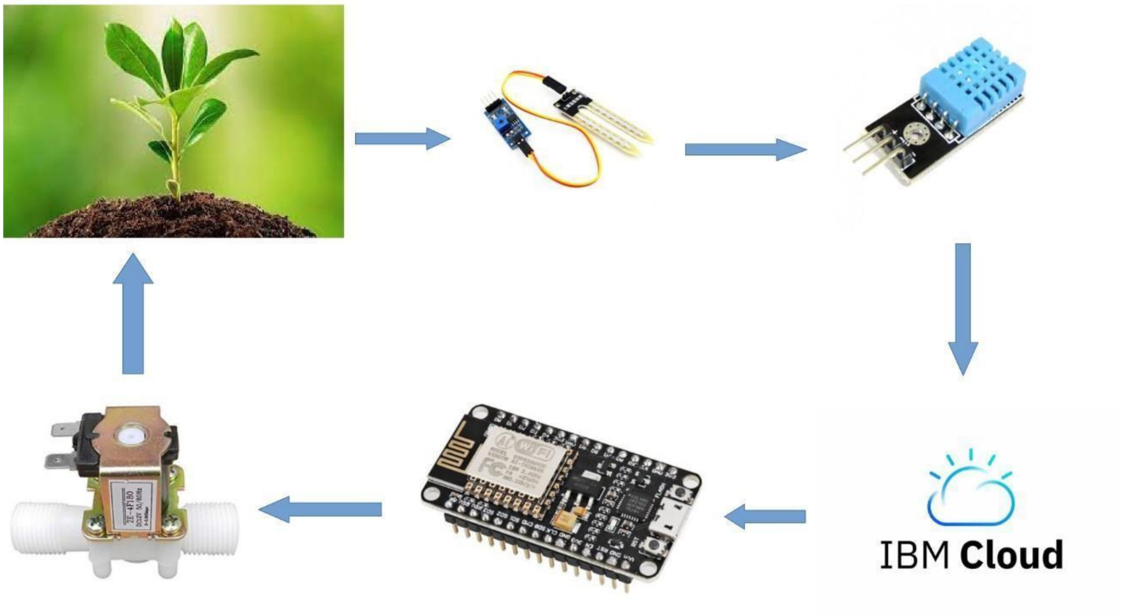

FIG 5.2.1

5.3 USER STORIES

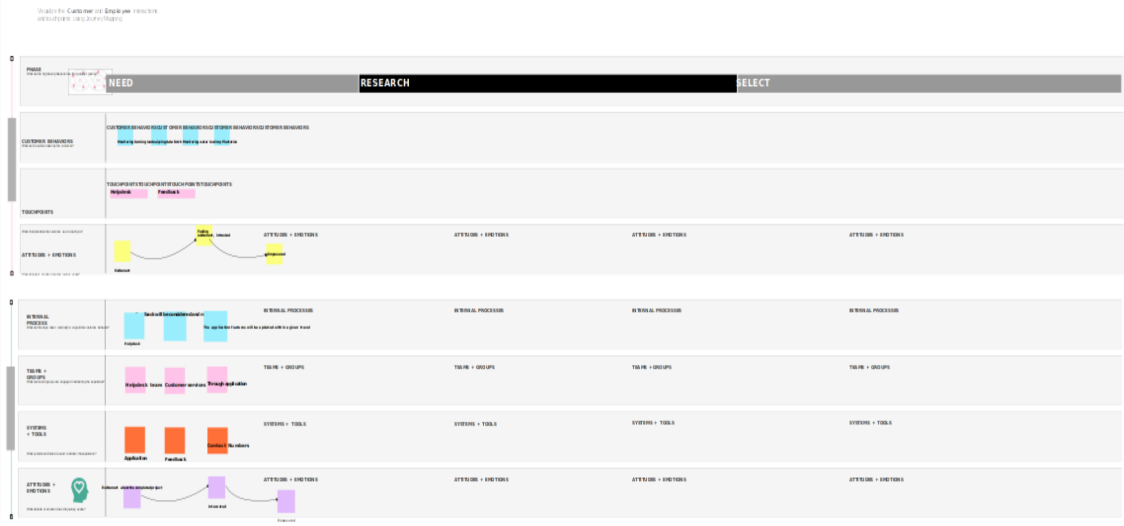

FIG 5.3.1

FIG 5.3.2

CHAPTER-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	Creating And Connecting IBM cloud for Project	USN-1	As a user, I want to know about the parameters of my field and control motors from anywhere	2	High	j.karthick, P.Tamilselvan
Sprint-2	Creating Node-Red service and connect with IBM cloud and WebUI	USN-2	As a user, I want to View my parameters on WebUI	1	High	A.vijayaragavan
Sprint-3	Preparing User interface on MIT app Invertor	USN-3	As a user, I want to know the parameters on Mobile app	2	Low	S.Sridharan
Sprint-4	Connecting and Configuring the services and debug the errors	USN-4		2	Medium	T.veerasakthivel

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	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	30	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	40	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	50	16 Nov 2022

CHAPTER-7

CODING AND SOLUTIONS

7.1 FEATURE 1

7.1.1 Node MCU

I have used NodeMCU esp8266 and if you are using any other vendor wifi chips or generic wifi module please check with the esp8266 Pin mapping which is very essential to make things work.

The reason why I used the D7 pin for this example is , I uploaded the basic blink program that comes with the example program in the arduino IDE which is connected with 13 pins of the arduino. The 13th pin is mapped into the D7 pin of NodeMCU.

go to board and select the type of esp8266 you are using. and select the correct COM port to run the program on your esp8266 device.

Source code:

```
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage
  LOW delay(1000);    // wait for a second
}
```

7.2 FEATURE 2

7.2.1 DHT11 Sensor

The DHT11 is a commonly used Temperature and humidity sensor that comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data.

Source code:

```
#include <dht.h>
#define dht_apin A0 // Analog Pin sensor is connected to

dht DHT;
void setup(){
  Serial.begin(9600);
  delay(500);//Delay to let system boot
  Serial.println("DHT11 Humidity & temperature Sensor\n\n");
  delay(1000);//Wait before accessing Sensor
}//end "setup()"


void loop(){
  //Start of Program
    DHT.read11(dht_apin);
    Serial.print("Current humidity =
    "); Serial.print(DHT.humidity);
    Serial.print("% ");
    Serial.print("temperature = ");
    Serial.print(DHT.temperature);
    Serial.println("C ");
    delay(5000);//Wait 5 seconds before accessing the sensor again.
  //Fastest should be once every two seconds.
}// end loop(
```

CHAPTER-8

TESTING

8.1 TEST CASE

Web application using Node Red

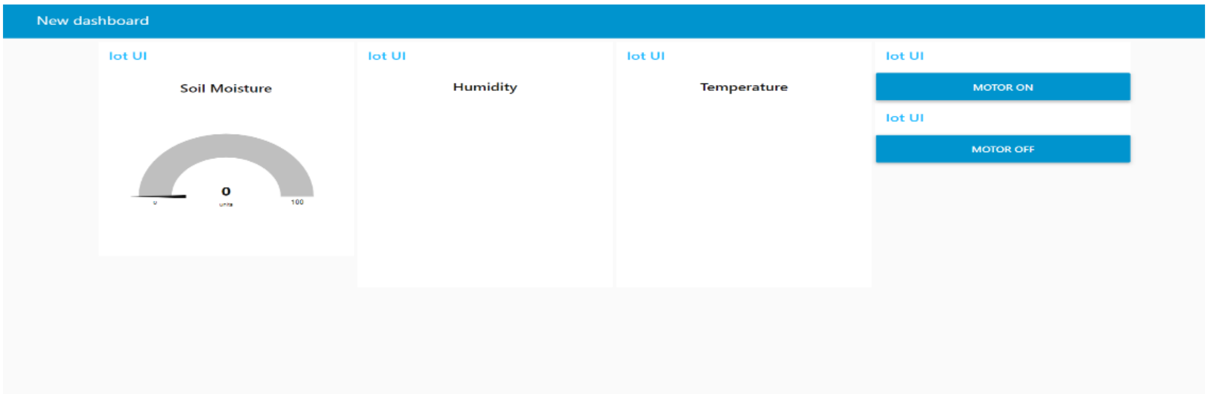

FIG 8.1.1

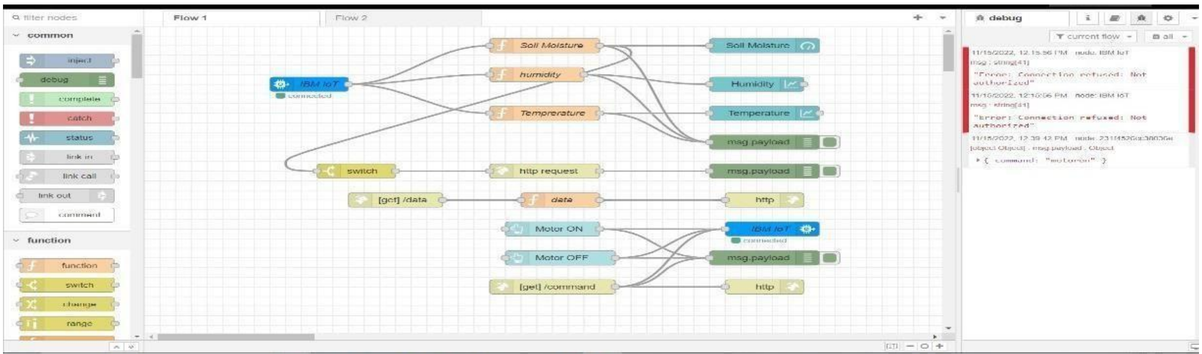

FIG 8.1.2

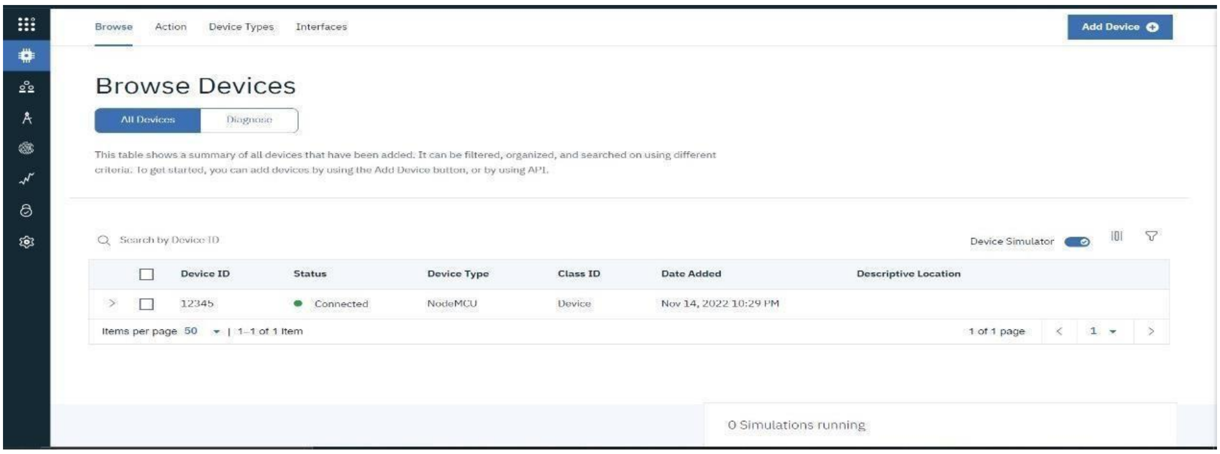

FIG 8.1.3

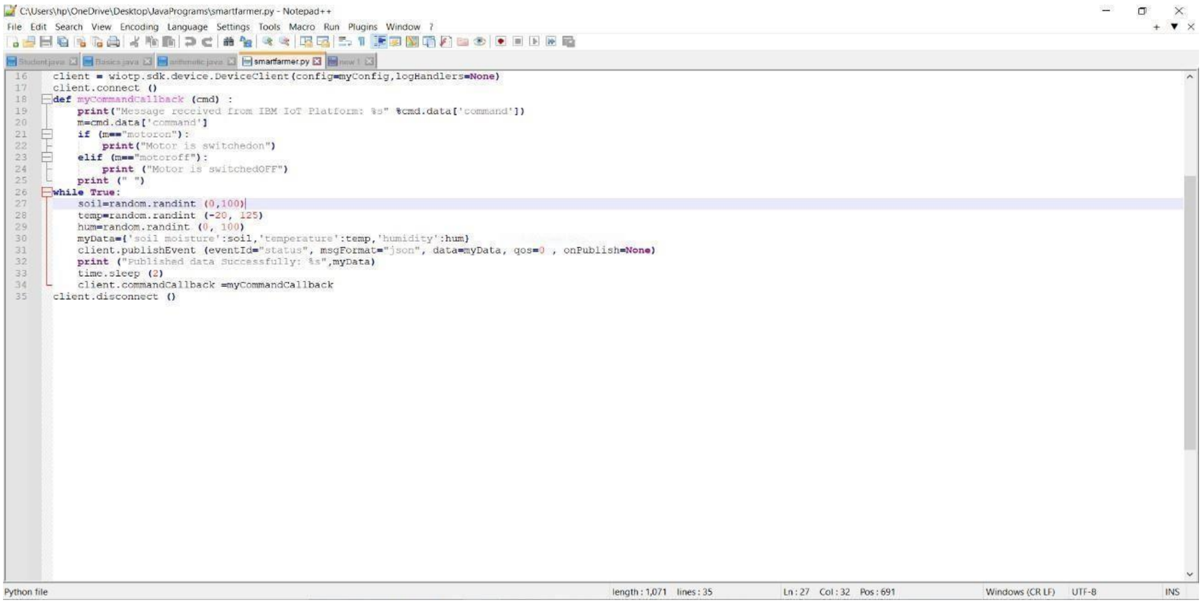
C:\Users\hp\OneDrive\Desktop\JavaPrograms\smartfarmer.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
smartfarmer.py
client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=None)
client.connect ()
def myCommandCallback (cmd) :
print("Message received from IBM IoT Platform: %s" %cmd.data['command'])
m=cmd.data['command']
if (m=="motoron"):
print("Motor is switchedon")
elif (m=="motoroff"):
print ("Motor is switchedOFF")
print (" ")
while True:
soil=random.randint (0,100)
temp=random.randint (-20, 125)
hum=random.randint (0, 100)
myData={'soil moisture':soil,'temperature':temp,'humidity':hum}
client.publishEvent (eventId="status", msgFormat="json", data=myData, qos=0 , onPublish=None)
print ("Published data Successfully: %s",myData)
time.sleep (2)
client.commandCallback =myCommandCallback
client.disconnect ()
Python file
length : 1,071 lines : 35
Ln : 27 Col : 32 Pos : 691
Windows (CR LF) UTF-8 INS

FIG 8.1.4

8.2 USER ACCEPTANCE TESTING

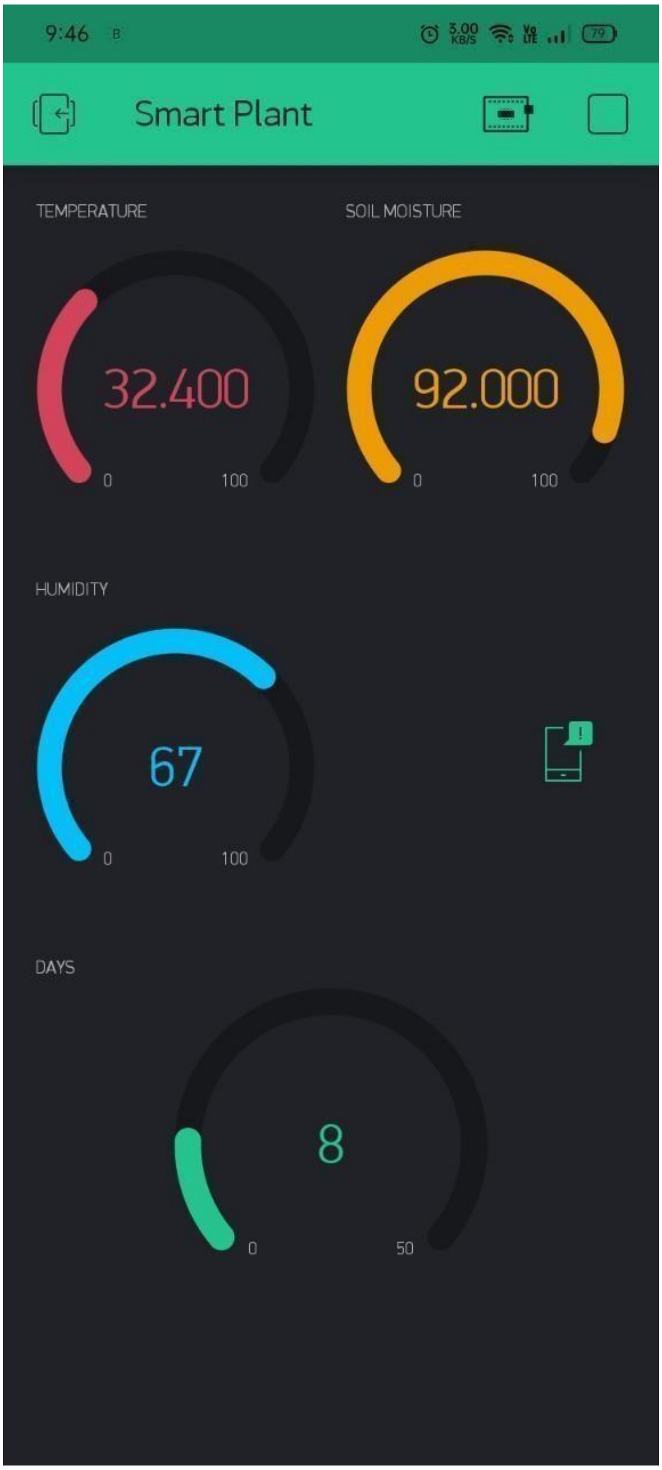

FIG 8.2.1

CHAPTER-9

RESULTS

9.1 PERFORMANCE METRICS

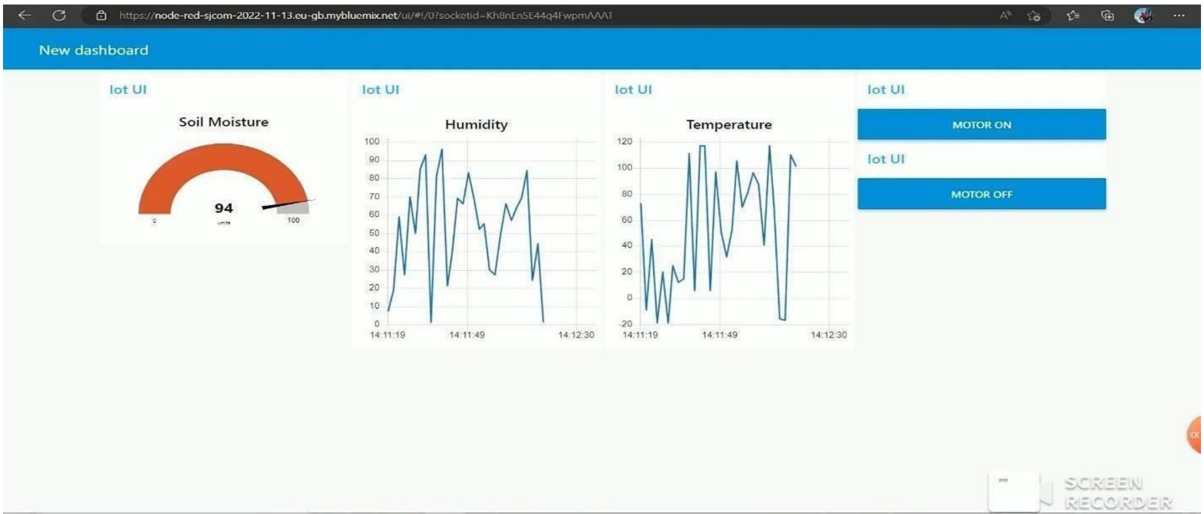

FIG 9.1.1

CHAPTER-10

ADVANTAGES AND 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.

DISADVANTAGES:

- The smart agriculture needs availability of internet continuously. Rural parts of most developing countries do not fulfill this requirement. Moreover, internet connection is slower.

- The smart farming based equipment require farmers to understand and learn the use of technology. This is a major challenge in adopting smart agriculture farming at large scale across the countries.

CHAPTER-11
CONCLUSION

Farmers can benefit greatly from an IoT-based smart agriculture system. As a result of the lack of irrigation, agriculture suffers. Climate factors such as humidity, temperature, and moisture can be adjusted depending on the local environmental variables. Thistechnology 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 do 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 units as an alternative to mobile apps. By GSM, SMS can be sent to farmers' phones.

CHAPTER-12

FUTURE SCOPE

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

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

- We can update this project by using solar power mechanisms. 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 for 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 are internet issues.

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

CHAPTER-13

APPENDIX

13.1 SOURCE CODE

```
#define BLYNK_PRINT Serial
#include <OneWire.h> #include
<SPI.h>
#include <BlynkSimpleEsp8266.h> #include
<DHT.h>
#include <DallasTemperature.h> #define
ONE_WIRE_BUS D2 OneWire
oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
char auth[] =”BOsoqmyvsdJSenk51n6EplHN0jqela_aajwi”; char ssid[]
= “Iqbal”;
char pass[] =
“Password”; #define
DHTPIN 2
#define DHTTYPE DHT11 DHT
dht(DHTPIN, DHTTYPE);
SimpleTimer timer; void
sendSensor()
{
float h = dht.readHumidity(); float t =
dht.readTemperature();

if (isnan(h) || isnan(t)) {
Serial.println(“Failed to read from DHT sensor!”); return;
}
```

```
Blynk.virtualWrite(V5, h); //V5 is for Humidity Blynk.virtualWrite(V6,
t); //V6 is for Temperature
}
void setup()
{
Serial.begin(9600); dht.begin();

timer.setInterval(1000L, sendSensor);
Blynk.begin(auth, ssid, pass); sensors.begin();
}
int sensor=0;
int output=0;
void sendTemps()
{
sensor=analogRead(A0);
output=(145-map(sensor,0,1023,0,100)); //in place 145 there is 100(it change with the change in
sensor)
delay(1000);
sensors.requestTemperatures();
float temp = sensors.getTempCByIndex(0);
Serial.println(temp);
Serial.print("moisture = ");
Serial.print(output);
Serial.println("%");
Blynk.virtualWrite(V1, temp);
Blynk.virtualWrite(V2,output);
delay(1000);

}
void loop()
```

```
{

Blynk.run();

timer.run();
sendTemps();
}

#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the BlynkApp.

// Go to the Project Settings (nuticon). Char auth [] ="YourAuthToken";

// Your WiFi credentials.

// Set password to "" for opennetworks. Char ssid [] ="YourNetworkName"; Char pass []
="YourPassword"; Void setup()

{

// Debug console Serial.begin(9600);Blynk.begin(auth, ssid,pass);

}

Void loop ()

{

Blynk.run();

}

#include <SPI.> #include
```

```
<Ethernet.h> #include"DHT.h"

#define DHTPIN 2

 #defineDHTTYPEDHT11

String tempVarId = "575475df7625423fd9da9c36"; String humVarId

= "575475f1762542406cb10c43"; String lightVarId = "575475fc762542410358a0c3"; String
soilVarId = "5754760576254241593d4d47";
String token = "xxxxx";

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED}; char
        server[]="things.ubidots.com"; EthernetClient client;

IPAddress ip(192, 168, 1, 40); // Arduino IP Add  IPAddressmyDns(8,8,8,8); IPAddress
myGateway(192,168,1,1);
int moisturePin = 0; intlightPin = 3; DHT
dht(DHTPIN,DHTTYPE);
void setup() {

  Serial.begin(9600);

  Serial.print("Starting...");

// Net connection...

}

void loop() {

  float soilHum =analogRead(moisturePin);

soilHum = (1023 - soilHum) * 100/1023;
```

```
  Serial.println("Soil Humidty: " +String(soilHum));
// Read light

  float volts = analogRead(lightPin) * 5.0 /1024.0;

  float amps = volts /10000.0;

  float microamps = amps *1000000;

  float lux = microamps * 2.0;
  Serial.println("Lux: " +String(lux));
  float h =dht.readHumidity();

  float temp =dht.readTemperature();
  Serial.println("Temp: " +String(temp,2));

  Serial.println("Hum: " +String(h,2));

  sendValue(temp, h, lux,soilHum);

  delay (60000);

}
void sendValue(float tempValue, float humValue, floatlux, float soil)

{

  Serial.println("Sending data...");

// if you get a connection, report back viaserial:
```

```
int bodySize = 0;

delay(2000);

// Post single value to singlevar

String varString = "[{\"variable\": \"" + tempVarId +"\",

\"value\":"+ String(tempValue) +"}";

// Add other variables Serial.println("Connecting...");

if (client.connect(server,80)) { client.println("POST/api/v1.6/collections/values HTTP/
1.1");
Serial.println("POST/api/v1.6/collections/values HTTP/1.1");
client.println("Content-Type: application/json"); Serial.println("Content-
Type: application/json"); client.println("Content-Length:
"+String(bodySize));
 Serial.println("Content-Length:     "+String(bodySize));

client.println("X-Auth-Token: "+token);

Serial.println("X-Auth-Token: "+token); client.println("Host:
things.ubidots.com\n"); Serial.println("Host:things.ubidots.com\n");
client.print(varString);

}

else {

// if you didn't get a connection to theserver:

Serial.println("connection failed");

}
```

```
boolean sta =client.connected();

 Serial.println("Connection["+String(sta)+"]");

if (!client.connected()) {

 Serial.println();

 Serial.println("disconnecting.");

 client.stop();

 }
 Serial.println("Reading..");

while (client.available()){

char c =client.read();

 Serial.print(c);

  } client.flush();

  client.stop();

}
```

13.2 GIT HUB & PROJECT DEMO LINK

CONTENT	LINK
GITHUB	https://drive.google.com/file/d/1Avmh1x0t-SOcw4Nx_j5X6_jjgkDR3LX2/view?usp=drivesdk