

Data Analysis on Home Forestation Using IoT

BY

Tanveer Haque

ID: 153-15-634

This Report Presented in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Supervised By

Ms. Zerin Nasrin Tumpa

Lecturer

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

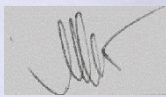
DHAKA, BANGLADESH

NOVEMBER 2019

APPROVAL

This Project/internship titled **“DATA MINING ON HOME FORESTATION”** submitted by **Tanveer Hoque**, ID No: 153-15-6334 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **12/07/2019**.

BOARD OF EXAMINERS



Dr. Syed Akhter Hossain
Professor and Head

Chairman

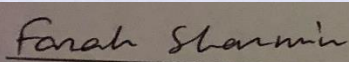
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Abdus Sattar
Assistant Professor

Internal Examiner

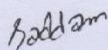
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Farah Sharmin
Senior Lecturer

Internal Examiner

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Dr. Md. Saddam Hossain
Assistant Professor

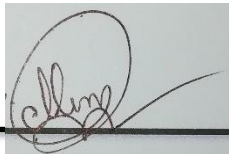
External Examiner

Department of Computer Science and Engineering
United International University

DECLARATION

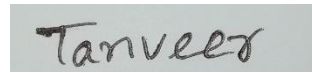
We hereby declare that, this project has been done by us under the supervision of **Ms. ZERIN NASRIN TUMPA**, Lecturer, **Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Ms. ZERIN NASRIN TUMPA
Lecturer
Department of CSE
Daffodil International University

Submitted by:



TANVEER HOQUE
ID: 153-15-6334
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

I want to express my gratefulness to the almighty Allah for his blessings that has made me possible to complete the research project successfully.

I want to thank my thesis advisor **Ms. Zerín Nasrín Tumpa**, Lecturer, Department of CSE Daffodil International University, Dhaka. The door was always open for me to reach her with all the problems I faced. Struggling through her hardest times of life, she had always been with me and supported me more like my elder sister. I lost my directions so many times during my research, so was she for the loss of her brother. But she remained strong to instruct me through the very best direction to the end.

I would conjointly like to convey my heartfelt gratitude to **Dr. Syed Akhter Hossain**, Head, Department of CSE, for his participation in the validation survey for this research project. Without his passionate participation and input, the validation survey could not have been successfully completed.

Finally, I must express my very earnest indebtedness to my parents. They provided me with unfailing support and continuous encouragement throughout these years of study and through the process of my research. Without their support, I could not have prepared myself to fulfill my dreams. All the hardships and sacrifices they made; I can't ever repay. I want to end my acknowledgment with a famous quote of *Robert Herjavee*-

"I had to be successful to justify my parents' sacrifice for me"

ABSTRACT

This paper automates plant monitoring, data collection and smart gardening using IoT. Home forestation (such as gardening) in rooftop, balcony, and living room has now a days become common in urban areas, because there are less land resources to plant trees. Also, it increases interior beauty of a house. In urban areas trees are decreasing day by day at an alarming rate. At the same time, the population is increasing inversely proportional to trees. As a result, the air is polluted at a massive rate in urban areas. Home forestation or gardening is a smart solution to this life-threatening problem. In urban areas, people lead busy life. They have less time to do gardening with proper knowledge. Even if they try to do gardening with less information, most of the times they fail. A plant needs 4 basic elements to grow such as: Light, Humidity, Temperature, Soil moisture. Every plant requires different environment variables. The motive of this research is to conduct data mining on these four elements in home environment to analyze the environment requirements for the survival of each plants based on Plant taxonomy. In future, it will help me to make a smart platform “Eco Friend” which will suggest users the plants that are best for his home environment. The data will be collected using IoT including the device NodeMCU, Sensors and later analyzed with Data mining tools.

TABLE OF CONTENTS

CONTENTS	PAGE
BOARD OF EXAMINERS	Error! Bookmark not defined.
DECLARATION	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
LIST OF FIGURES	vii
LIST OF TABLES	ix
Chapter 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Motivation	2
1.3 Rationale of the Study	2
1.4 Research Questions	3
1.5 Expected Output	3
1.6 Repot Layout	4
Chapter 2: BACKGROUND	5
2.1 Introduction	5
2.2 Related Works	5
2.3 Research Summary	6
2.4 Scope of the Problem	8
2.5 Challenges	8
2.6 Hardware Requirements	9
Chapter 3: RESEARCH METHODOLOGY	15
3.1 Introduction	15
3.2 Research Subject and Instrumentation	15
3.3 Data Collection Procedure	16
3.4 Implementation Requirements	24
Chapter 4: EXPERIMENTAL RESULTS AND DISCUSSION	25

4.1	Introduction	25
4.2	Experimental Results	25
4.3	Descriptive Analysis	37
4.4	Summary	44
Chapter 5: SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH		45
5.1	Summary of the Study	45
5.2	Conclusions	45
5.3	Recommendations	45
5.4	Implication for Further Study	46
REFERENCES		47
Appendices		48
	Appendix A: Research Reflection	48
PLUGIARISM REPORT		50

LIST OF FIGURES

FIGURE 1.1: GARDENING AT HOME	1
FIGURE 2.1: WORKFLOW OF ECO FRIEND	6
FIGURE 2.2: RAW DATA OF THE IOT DEVICE	7
FIGURE 2.3: UPLOADED SERVER DATA	7
FIGURE 2.4: FILTERED OUTPUT OF RANGED VALUES	8
FIGURE 2.5: NODEMCU 1.0 (ESP 12E)	9
FIGURE 2.6: DHT11 TEMPERATURE & HUMIDITY SENSOR	9
FIGURE 2.7: CAPACITIVE SOIL MOISTURE SENSOR	10
FIGURE 2.8: BH1750 LIGHT INTENSITY SENSOR	11
FIGURE 2.9: NOKIA 5110 LCD MODULE	11
FIGURE 2.10: PUSH BUTTON	12
FIGURE 2.11: JUMPER WIRES	12
FIGURE 2.12: BILL OF MATERIALS	13
FIGURE 3.1: DATA COLLECTION IN DEPTH	16
FIGURE 3.2: DIAGRAM OF THE DEVICE & SERVER	17
FIGURE 3.3: PIN DIAGRAM OF THE DEVICE	18
FIGURE 3.4: LINE GRAPH OF LIGHT INTENSITY	19
FIGURE 3.5: SCATTER GRAPH OF LIGHT INTENSITY	19
FIGURE 3.6: LINE GRAPH OF TEMPERATURE DATA IN CELSIUS	20
FIGURE 3.7: SCATTER GRAPH OF TEMPERATURE DATA IN CELSIUS	20
FIGURE 3.8: LINE GRAPH OF HUMIDITY IN %	21
FIGURE 3.9: SCATTER GRAPH OF HUMIDITY DATA IN %	21
FIGURE 3.10: LINE GRAPH OF SOIL MOISTURE SENSOR VALUE	22
FIGURE 3.11: SCATTER GRAPH OF SOIL MOISTURE SENSOR VALUE	22
FIGURE 3.12: BAR CHART FOR SOIL STATE COUNT	23
FIGURE 3.13: PIE CHART FOR SOIL STATE COUNT	23
FIGURE 3.14: BAR CHART FOR ALIVE & DEAD STATE	24
FIGURE 3.15: PIE CHART FOR ALIVE & DEAD STATE	24
FIGURE 4.1: EXPERIMENT PLANT TAXONOMY	25
FIGURE 4.2: ARDUINO UNO DIAGRAM	26
FIGURE 4.3: ESP8266 WI-FI MODULE	26
FIGURE 4.4: NODEMCU PIN DIAGRAM	27
FIGURE 4.5: SERIAL COMMUNICATION BETWEEN ARDUINO UNO & NODEMCU	27
FIGURE 4.6: DATA MISSING DUE TO SERIAL COMMUNICATION	27
FIGURE 4.7: BH1750 SENSOR MODULE	28
FIGURE 4.8: CODE UPLOAD ISSUE	28
FIGURE 4.9: PORT ISSUE	29
FIGURE 4.10: DHT11 SENSOR FAULT READING	29
FIGURE 4.11: SOIL MOISTURE SENSOR CORROSION	30
FIGURE 4.12: CAPACITIVE SOIL MOISTURE SENSOR AS A REPLACEMENT	30
FIGURE 4.13: NOKIA 5110 LCD AS A REPLACEMENT FOR 16X2 LCD	31
FIGURE 4.14: LCD WITHOUT BACKLIGHT	31
FIGURE 4.15: LCD WITHOUT RESISTOR	32

FIGURE 4.16: SERIAL DATA VIEWER AS ALTERNATIVE TO LCD	32
FIGURE 4.17: DEAD PLANT	33
FIGURE 4.18: REPLACEMENT	33
FIGURE 4.19: DJANGO ARCHITECTURE	34
FIGURE 4.20: LARAVEL ARCHITECTURE	35
FIGURE 4.21: FLASK ARCHITECTURE	35
FIGURE 4.22: CROSS SERVER COMMUNICATION	36
FIGURE 4.23: CIRCUIT DIAGRAM	37
FIGURE 4.24: DHT11 PIN DIAGRAM	37
FIGURE 4.25: BH1750 PIN DIAGRAM	38
FIGURE 4.26: CAPACITIVE SOIL MOISTURE SENSOR DEPTH RECOMMENDATION	38
FIGURE 4.27: CAPACITIVE SOIL SENSOR CALCULATION PROCEDURE	40
FIGURE 4.28: NOKIA 5110 LCD PINOUT	41
FIGURE 4.29: PUSH BUTTON DIAGRAM	42
FIGURE 4.30: FINAL DATAMINED OUTPUT	44

LIST OF TABLES

TABLE 4.1: PLANT TAXONOMY	34
TABLE 4.2: DHT11 INTERFACING TABLE	38
TABLE 4.3: BH1750 INTERFACING TABLE	38
TABLE 4.4: CAPACITIVE SOIL MOISTURE SENSOR INTERFACING TABLE	39
TABLE 4.5: NOKIA 5110 INTERFACING TABLE	41
TABLE 4.6: PUSH BUTTON INTERFACING TABLE	42

Chapter 1:

INTRODUCTION

1.1 Introduction

An indoor garden is often either sophisticated or straightforward as somebody wants. One can pay hundreds of thousands of moneys to create a garden, or even pay a bit. Horticulture or gardening could be a fun and relaxing way to get connected with nature. Gardening is an activity that's smart for both the mind and body at a similar time. It may be enjoyed and practiced by individuals of any ages. It's easy, grab your tools and get in the dirt! Is it really that much easy? In most cases, it's not.

“IoT (Internet of Things) is usually stated as internet of Objects, since IoT can rework anything-including ourselves.” This bold statement is given by considering the impact of web on education, business, communication, science and humanity, etc. Kelvin Ahston used the term IoT first in 1998, that currently has additional and additional development. All tale of human history the foremost powerful and vital creation is that the internet. The integrated and a part of future web is IoT. Within the field of business, human process, info and communication, the things are expected to become active participants by using IoT. They have to be enabled so as to be able to move with the surroundings themselves by reworking and exchanging the data and knowledge detected regarding the environment. It reacts mechanically to the real-world events and is influenced by the processes that make services and trigger actions with or without human intervention.



Figure 1.1: Gardening at Home

The planned system is intended and enforced employing a low-priced MasterCard sized NodeMCU that is controlled through internet under the windows environment. The smart device collects home environment data and sends the info to server. It logs the info and stores them as data. Later they're classified using data processing tools. The complete platform is developed to gather the desired information of a future device. The device can observe home environment, collect and analyze information, send this to server, the server can respond according the matched criteria of the surroundings and return range of variables for a specific plant and environment.

All garden parameters like humidity, temperature, soil moisture, light intensity is tracked by the system and this data is uploaded in the cloud.

Eco Friend (the device itself) incessantly monitors the conditions of the garden and collects each amendment of information that need immediate actions for the garden and advise the user.

1.2 Motivation

This was a long and hard procedure to work on this research-based project. This was initially proposed in the campus final of Hult Prize Competition 2017. The judges simply said it's a necessary and innovative step to encourage gardening. But this was an imaginary project and nobody would care about gardening. So, he simply refused it for the regional final event. That was when I realized I should not only keep it as an idea, rather make a real project on this.

Plants are precondition for the existence of life in earth. None of our basic functions from respiration to eating to drinking would even be possible without plants. Plants not solely act as food sources however they also unleash oxygen and help to maintain the water table. The straightforward fact of the matter is that without plants we cannot survive.

In today's busy world, we always have little time to know every detailed information. But lack of information could spoil a perfectly planned decision. My roommate has 3 plants. 2 of them are already dead and another is about to die. Why that might have happened? Because he went to a local nursery plant shop, there he found three trees that had cool color flowers, leaves were beautiful. He asked himself how much would they suite the beauty of his balcony, and he found out they would quite an impression. They really were beautiful at first. He used to care for them, nurture them, and water them. But after some time, he got himself busy with daily life. About a matter of time, these plants faced their destiny, and gets rotten. What if he knew, which plants was best for his environment? Which plants needed less nurture and attention? Maybe that could save his money and time. So, this device helps to learn more about the environment that a tree can survive, and then generate a database. Using the database, a user can learn which plants suits the criteria of his home environment and necessity (based on purpose, e.g. vegetable plants, flower plants, medicinal herbs, and leaflet plants).

1.3 Rationale of the Study

There are so many projects on IoT is undergoing right now to make smart garden, which involves in smart watering and lighting systems. They use advanced technique and method using IoT devices and android platform to make user friendly gardening.

By this research, the gardening is taken to a whole new level by implying data mining techniques on home environment.

1.4 Research Questions

These are few research questions regarding my research topic

- Does gardening need computer supervision rather than basic gardening knowledge?
- Can smart gardening reduce tree death rate?
- Accuracy of the data collected is up to the standard?
- How much adaption is possible in the variation of plants?
- How technology can save trees, whereas it is a major reason of environment threat?
- Can a garden at home can produce enough oxygen for a family?
- Does a garden release that much carbon di oxide which may harm someone?

1.5 Expected Output

I am working on a specific plant and my home environment. I tried to create a massive data set of environment logs consisting of the below information

1. Analyze **Experimental environment**
 - a. Gather the information based on **Room Environment**
 - i. Light intensity
 - ii. Temperature
 - iii. Humidity
 - b. Gather information based on **Plant**
 - i. Soil moisture
 - c. **Collect** and **Format** data
 - d. Send Data to the **Server**
 - e. The server will do the following
 - i. Save the requested data
 - ii. Notice any suspicious occurrences
 - f. Scrapping the database
 - i. Get plants **upper and lower threshold** limit of environment variables
2. Analyze **User environment**
 - a. The device will get the information based on room environment
 - i. Light intensity
 - ii. Temperature
 - iii. Humidity
3. Interact with user
 - a. Maintain
 - i. Automatic water system
 - ii. Automatic lighting system
 - iii. Notify user about the tree

1.6 Repot Layout

Chapter 1: Introduction

In this chapter, we have discussed the motivation, objectives and the expected outcome of the project. Why plants are necessary for human life. How plants are getting threatened day by day. Moreover, research questions and expected outcomes also been discussed based on the project. Later followed by the report layout.

Chapter 2: Background

I discussed the background circumstances and hardware details of our project. I also mentioned the related work, comparative studies, scope of the problem and challenges of the project.

Chapter 3: Research method

In this chapter, we discussed the requirements like the use case model of the project and their descriptions, the logical data model and the design requirements.

Chapter 4: Experimental results and discussion in this chapter, we discussed all the designs of the research with proper descriptions. We also discussed about individual result descriptions.

Chapter 5: Conclusion and Future Scope We discussed the conclusion and the scope for further developments which can make a vast sector for this system.

Chapter 2: BACKGROUND

2.1 Introduction

As per the growing rate of population with spontaneous consumption of resources, creates in the need for the managing the available resources at its best. So, it was necessary to manage the outflow of the two major resources i.e. water and electricity and to formulate out, that's how it can be protected from getting wasted and could be utilized at its best.

As during the survey study, it was found that (Dhaka, in Bangladesh) the practices were manual and a major portion of resources was wasted due to slothful and unconcerned behavior leading to the death of trees and unwanted operation of the lights.

So, using technology, and modern statistical, survey-based study it was discovered that major portion of the resources (Electricity and Water) could be managed out and preserved by managing their controlled flow in an allotted area. [2]

2.2 Related Works

There are lots of works on IoT devices regarding smart plant monitoring system. There are logs to classify the environment required to plant

Plant Monitoring System by Ryan Gill

Real time plant monitoring system is utilized in his project to get temperature, light exposure and moisture. He started off by loading the MKR1000 with the Standard Firmata Wi-Fi sketch. This allowed him to communicate with the board using Johnny-Five. [3]

Plant Monitoring System by STEMpedia

This project demonstrates a way to track parameters like moisture of the soil, sunlight, humidity, and temperature of the plant atmosphere and show it on evive display.

Monitoring plant health is incredibly necessary for their quick growth. in this busy world, people sometimes forget to water their plants that ends up in unhealthy growth and health of their plants. For guaranteeing complete development of plants it's necessary to develop correct encompassing conditions in which plants grow. he tends to aim at building a little DIY Plant observance system whose information are displayed on the evive screen. The evive screen can show the subsequent data-

Automated Plant monitoring System by TUPT

He decided to make a project which would help people see the importance of technology on our environment and with their lives. He came up with researching the projects that students have done using the Arduino. But he thought of making something that is existing but not most students have done improving it, so he decided to make an Automated Plant Monitoring System.

His major components for this project was the Gizduino and Raspberry Pi 3. Since plant owners are still doing the manual process of monitoring their plants for it to grow, by applying technology.

2.3 Research Summary

To get the desired output the research is conducted following these steps-

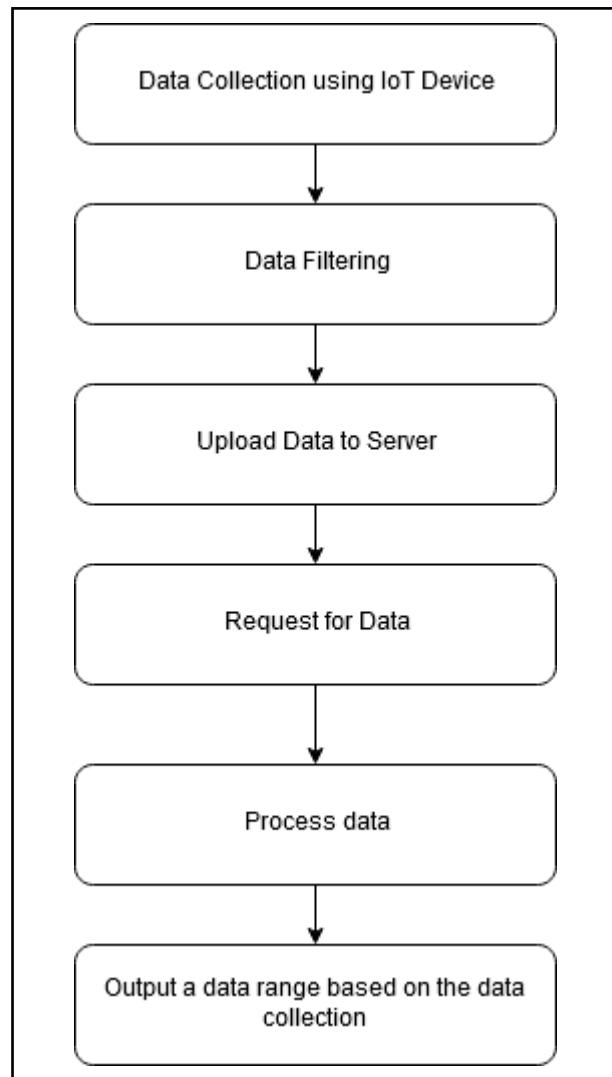


Figure 2.1: Workflow of Eco Friend

I have collected 1000's of data from the IoT device directly. It continuously monitored the environment.

```
Request Check:
Host: localhost:8000/api
light=10&temperature=29.00&humidity=70.00&soil=Wet&moisture=747&remarks=null&alive=false/-11

closing connection
Srvr: 0
Lite: 10 Lx
Temp: 29.00C
Humd: 70.00 %
Soil: Wet
Mois: 747
live: 0
```

Figure 2.2: Raw Data of the IoT Device

The above figure is captured from the Serial communication output of NodeMCU. It is the raw data after posting it to server.

```
{
  {
    "id": 3,
    "url": "http://djangpi.herokuapp.com/api/datalogs/3/?format=json",
    "plant": "http://djangpi.herokuapp.com/api/plants/1/?format=json",
    "light": 1,
    "temperature": 30,
    "humidity": null,
    "soil": null,
    "moisture": null,
    "remarks": "6",
    "alive": 1
  },
  {
    "id": 4,
    "url": "http://djangpi.herokuapp.com/api/datalogs/4/?format=json",
    "plant": "http://djangpi.herokuapp.com/api/plants/1/?format=json",
    "light": 1,
    "temperature": 2,
    "humidity": null,
    "soil": null,
    "moisture": null,
    "remarks": "6",
    "alive": 1
  },
  {
    "id": 5,
    "url": "http://djangpi.herokuapp.com/api/datalogs/5/?format=json",
    "plant": "http://djangpi.herokuapp.com/api/plants/1/?format=json",
    "light": 1,
    "temperature": 2,
    "humidity": null,
    "soil": null,
    "moisture": null,
    "remarks": "6",
    "alive": 1
  },
  {
    "id": 6,
    "url": "http://djangpi.herokuapp.com/api/datalogs/6/?format=json",
    "plant": "http://djangpi.herokuapp.com/api/plants/1/?format=json",
    "light": 1,
    "temperature": 2,
    "humidity": null,
    "soil": null,
    "moisture": null,
    "remarks": "6",
  }
```

Figure 2.3: Uploaded Server Data

The above figure describes the json data fetched from the server, that was posted earlier with the help of the Eco Friend IoT Device.

The Data is processed and later using filtering and data mining techniques we get the desired range values

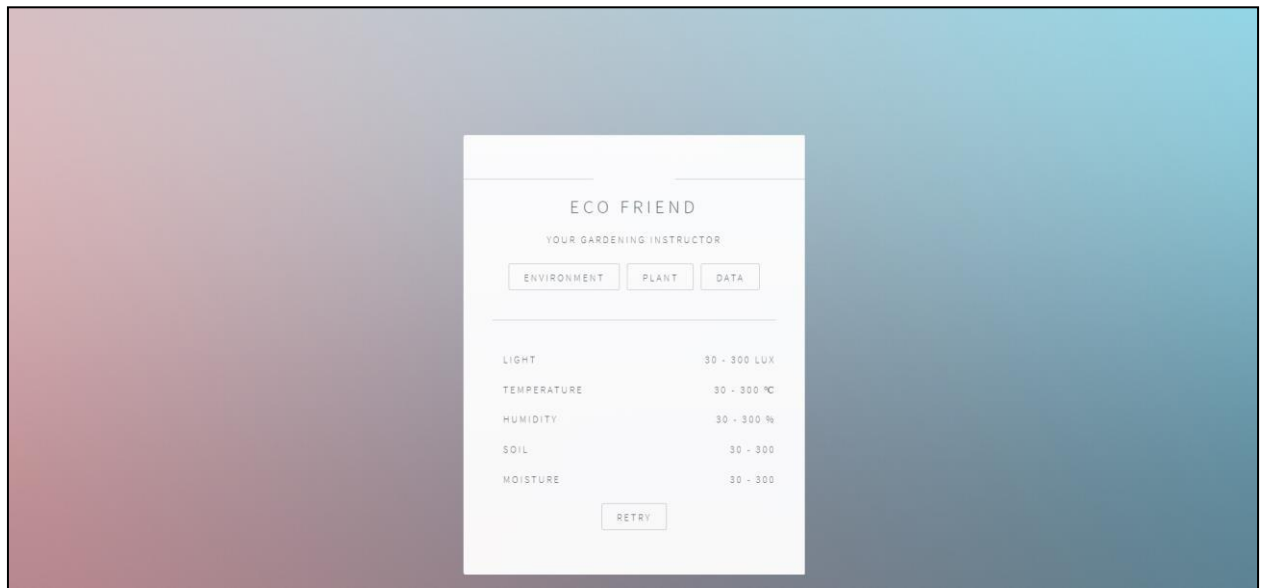


Figure 2.4: Filtered Output of Ranged Values

This is the final output of the dataset. That illustrates the range of values for the desired plant.

2.4 Scope of the Problem

This research-based project has tremendous scope in practical fields. Some examples are given below:

- Plant environment analysis
- Genetic transfusion of plants based on research data
- Predict plant's survival based on environment
- Extend research on plants

2.5 Challenges

The proposed device should have the following challenges to overcome

- To operate in friendly environment
- To work with cheap device that is prone to error
- Irrelevant and unwanted data due to improper utilization of programming
- Working with massive datasets and options
- Server response issues
- Heavy dataset leads to no result at all
- Needs premium materials
- Work on cross platform servers
- Low level programming for IoT leads to more errors

2.6 Hardware Requirements

There were lots of parts and devices used to complete the device. Some of them finally came out with limitations. These were replaced or removed later. The final parts that makes the final device is as follows-

2.6.1 NodeMCU 1.0 (ESP 12E)



Figure 2.5: NodeMCU 1.0 (ESP 12E)

The NodeMCU is a development board that includes the favored ESP8266 Wi-Fi chip. As it seems, you'll be able to program the ESP8266 a bit like the other microcontroller. Its obvious advantage over the Arduino or PIC is that it will promptly connect with the web via Wi-Fi. However, the ESP8266 prison-breaking board has restricted pins though the chip itself contains a heap of output ports. The NodeMCU solves this drawback by that includes ten GPIO pins each capable of using PWM, I2C and 1-wire interface. [4]

2.6.2 DHT11

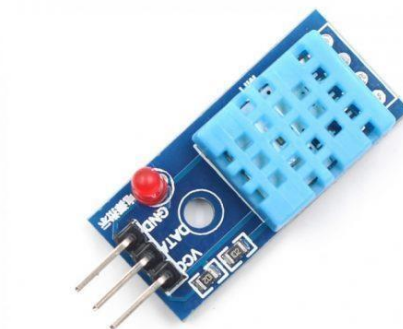


Figure 2.6: DHT11 Temperature & Humidity Sensor

DHT11 is a humidity and Temperature detecting sensor that generates calibrated digital output. DHT11 are often interface with any microcontroller like Arduino, Raspberry Pi, etc. and find fast results. DHT11 may be an affordable humidity and temperature detector that provides high reliability and long stability. [5]

It uses a capacitive humidity detector and a semiconductor to measure the encircling air, and outputs a digital signal on the data pin (no analog input pins needed). It's terribly straightforward to use, and libraries and sample codes are accessible for Arduino and Raspberry Pi.

This module makes it simple to attach the DHT11 detector to an Arduino or microcontroller as it includes the pull up resistance required to use the detector. Solely 3 connections are needed to be created to use the detector – Vcc, Gnd and Output.

It has high reliability and glorious long stability, because of the exclusive digital signal acquisition technique and temperature sensing technology.

Specifications-

Power Supply : 3.3~5.5V DC

Output : 4 pin single row

Measurement Range : Humidity 20-90%RH, Temperature 0~50°C

Accuracy : Humidity +-5%RH, Temperature +-2°C

Resolution : Humidity 1%RH, Temperature 1°C

Interchangeability : Fully Interchangeable

Long-Term Stability : <±1%RH/year

Pin Description: -

Pin 1: Power +ve (3.3VDC to 5.5VDC Max wrt. GND)

Pin 2: Serial Data Output

Pin 3: Power Ground or Power –ve

2.6.3 Capacitive Soil Moisture Sensor v1.2



Figure 2.7: Capacitive Soil Moisture Sensor

Soil moisture device measures soil moisture levels by electrical phenomenon sensing instead of resistive sensing like alternative sensors on the market. It's made from corrosion resistant material which provides it a wonderful service life. Insert it in to the soil around your plants and impress your friends with real-time soil moisture data! This module includes an on-board transformer which provides it an operative voltage vary of 3.3 ~ 5.5V. It's excellent for low-voltage MCUs, both 3.3V and 5V. For compatibility with a Raspberry Pi it'll need an ADC convertor. This soil moisture device is compatible with our 3-pin "Gravity" interface, which might be directly connected to the Gravity I/O enlargement shield. [6]

2.6.4 Digital Light intensity sensor module

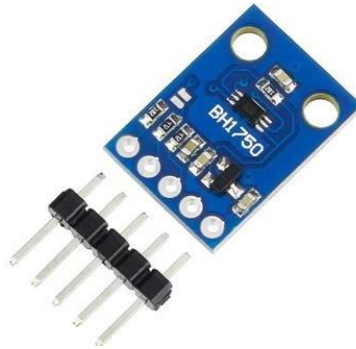


Figure 2.8: BH1750 Light Intensity Sensor

BH1750FVI is a digital ambient light sensing element IC for I2C bus interface. This IC is the best suited to get the ambient light information for adjusting liquid crystal display and keyboard backlight power of Mobile Phones. It's attainable to find wide selection at High resolution. Its Spectral responsibility is roughly human eye response. This has terribly low source of illumination dependency like electric lamp, lamp, halogen Lamp, White LED. There is no need of any external elements and comes with 50Hz or 60Hz light noise reject function. It's attainable to pick two kind of I2C slave-address. [7]

2.6.5 Nokia 5110 LCD Display



Figure 2.9: Nokia 5110 LCD Module

The Nokia 5110 is a basic graphic LCD screen for ample applications. It was originally supposed to be used as a cellular phone screen. This one is mounted on a straightforward to solder PCB. [8]

It uses the PCD8544 controller that is the same utilized in the Nokia 3310 liquid crystal display. The PCD8544 could be a low power CMOS LCD controller/driver, designed to drive a graphic display of forty-eight rows and eighty-four columns. All necessary functions for the display are provided in an exceedingly} very single chip, in conjunction with on-chip generation of liquid crystal display offer and bias voltages, resulting in a minimum of external elements and low power consumption.

2.6.6 Push Button



Figure 2.10: Push Button

A six Pin Push Switch additionally referred to as mini DPDT Push Switch, is nothing however a mix of 2 push switches placed together within one package. not like short switches that connect the wires of the switch just for a second, this switch retains its ON-OFF state until pushed afterward. for instance, if I push it once so it's turned on, it'll stay in ON state until it's pushed once more. That's why this switch is helpful in dominant power connections most of the time.

2.6.7 Jumper wires



Figure 2.11: Jumper Wires

Jumper wires are merely wires that have connection pins at every end, permitting them to be used to connect 2 points to every other without soldering. Jumper wires are usually used with breadboards and alternative prototyping tools so as to create it simple to change a circuit as required. Fairly easy. In fact, it doesn't get way more basic than jumper wires.

2.6.8 Parts Bill

Bill of Materials: Eco Friend					
Tuesday, August 6 2019, 04:52:06					
Assembly List					
#	Unit	Label	Part Type	Description	Price
1	1	NodeMCU	ESP8266 WiFi Board	NodeMCU V3 ESP8266 ESP-12E is WiFi development board that helps you to prototype your IoT product with few Lua script lines, or through Arduino IDE. The board is based on ESP8266 ESP-12E variant, unlike other ESP-12E, you won't need to buy a separate breakout board, usb to serial adapter, or even solder it to a PCB to get started, you will only need a usb cable (Micro USB).	350
2	1	Nokia 5110 LCD Module	Nokia 5110 LCD	The Nokia 5110 is a basic graphic LCD screen for lots of applications. It was originally intended for cell phone screen. This one is mounted on an easy to solder PCB. It uses the PCD8544 controller, which is the same used in the Nokia 3310 LCD. The PCD8544 is a low power CMOS LCD controller/driver, designed to drive a graphic display of 48 rows and 84 columns. All necessary functions for the display are provided in a single chip, including on-chip generation of LCD supply and bias voltages, resulting in a minimum of external components and low power consumption. The PCD8544 interfaces to microcontrollers through a serial bus interface.	185
3	1	Capacitive Soil Moisture Sensor	DFRobot Capacitive Soil Moisture Sensor V1.0	This capacitive soil moisture sensor is distinguished from most resistive sensors on the market and uses capacitive sensing to detect soil moisture. The problem that the resistance sensor is easily corroded is avoided, and its working life is greatly extended. The sensor has a built-in voltage regulator chip that supports a 3.3-volt working environment, which means it works even on a 3.3V Arduino control board. A miniature PC such as the Raspberry Pi only needs an external ADC (analog to digital signal) conversion module to work. With an external screen and a motherboard, you can talk to your plants! See if it is thirsty and you don't need more water to moisten.	500
4	1	Humidity & Temperature Sensor	KY-015 Temperature & Humidity Sensor Module	variant variant 2; temperature measurement range 0°C to 50°C [32°F to 122°F]; pins 3; package Breakout board; temperature measurement resolution 1°C; chip label KY-015; editable pin labels false; humidity measurement resolution 1% RH; signal transmission range 20m; humidity measurement range 20% to 90% RH; humidity measuring accuracy ±5% RH; pin spacing 300mil; temperature measurement accuracy ±2°C; operating voltage 3.3V to 5.5V; dimensions 18.5mm x 16.5mm [0.728in x 0.650in]	100
5	1	Light Intensity Sensor	BH1750	This Module is a 16-Bit digital Ambient Light Sensor Module with I2C bus interface.	350
6	21	Jumper Wires	Male to Male, Male to Female		50
					Total: 1535
Exported with Fritzing 0.9.3- http://fritzing.org					

Figure 2.12: Bill of Materials

2.6.9 Software Requirements

The below softwares were used to complete the project:

- 2.8.1 Arduino IDE
- 2.8.2 Draw.io
- 2.8.3 Fritzing
- 2.8.4 PyCharm Professional
- 2.8.5 PHPStorm
- 2.8.6 Postman
- 2.8.7 Git

Chapter 3:

RESEARCH METHODOLOGY

3.1 Introduction

The automatic plant observation system has recently attracted tremendous interest thanks to the potential application in rising technology [9]. More importantly, this technique is used to enhance the performance of existing techniques or to develop and design new techniques for the growth of plants. The plant observation system is useful for watering the plants and to observe few parameters for growth of plants. this method is incredibly utilized in few areas like nursery farms and in agriculture. in this system a mechanism is established to seek out the moisture content within the soil with the assistance of soil moisture sensor and relying upon the condition of the device the water is controlled. Another important parameter is by capturing the light, temperature and humidity of the environment by using NodeMCU, and processing the data by using data mining technique to analyze and determine the environment. This helps in providing the appropriate amount of water for plants so reduces some situations like mud cracks, water logging. This helps in irrigating the field even during night time, so does not require the user to switch ON the motor manually.

3.2 Research Subject and Instrumentation

Subject: Data mining on Home Forestation using IoT

Instrumentation:

- NodeMCU
- DHT11
- Light Sensor Module
- Soil Moisture Sensor
- LCD
- Digital Meter
- Jump wires
- Python Server: Django
- PHP Server: Laravel

Software Requirements:

- Arduino IDE
- PHPStorm
- PyCharm Professional
- Pandas Library
- NumPy Library
- Nokia 5110 LCD Library
- DHT11 Library
- BH1750 Library
- JSON Library
- NodeMCU Library
- Adafruit GFX Library

3.3 Data Collection Procedure

3.3.1 Flowchart

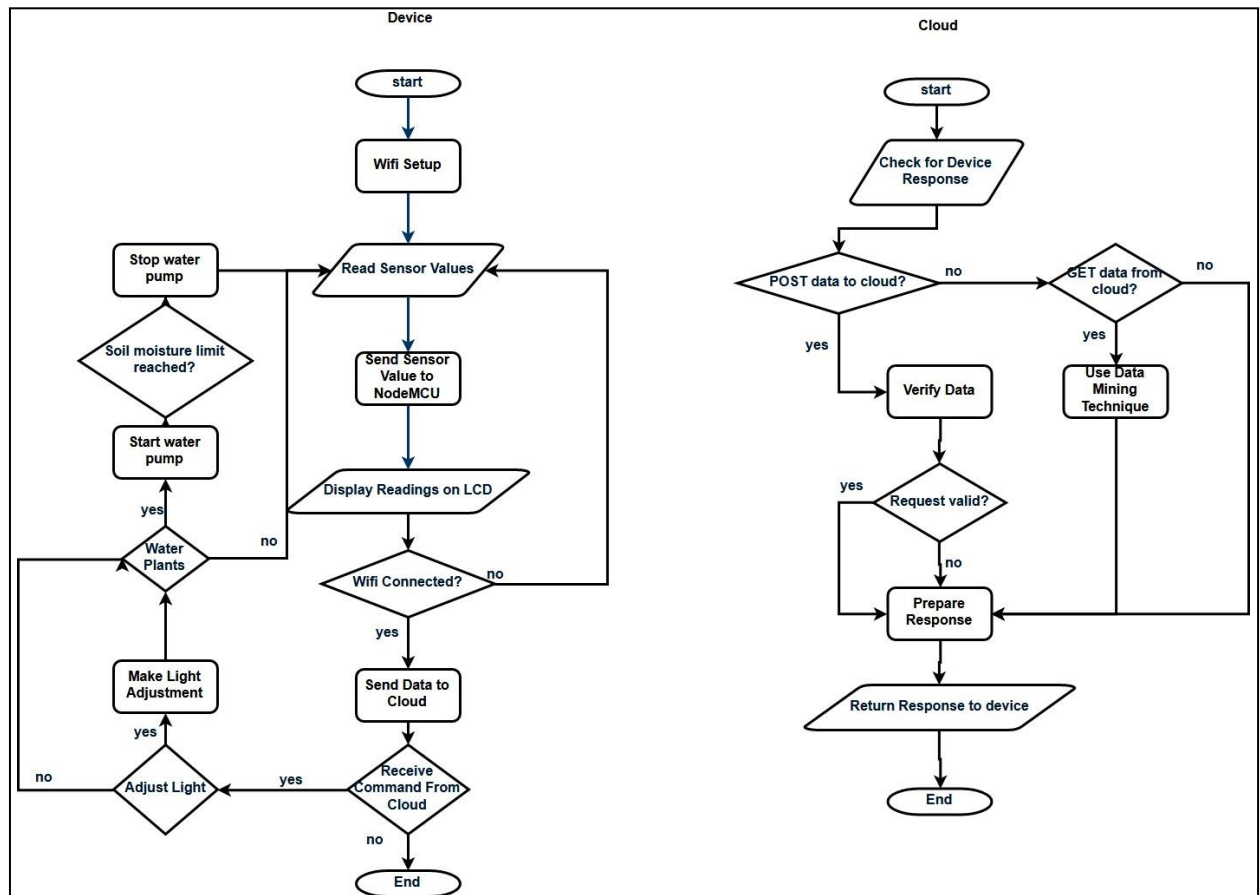


Figure 3.1: Data Collection in Depth

3.3.2 Diagram

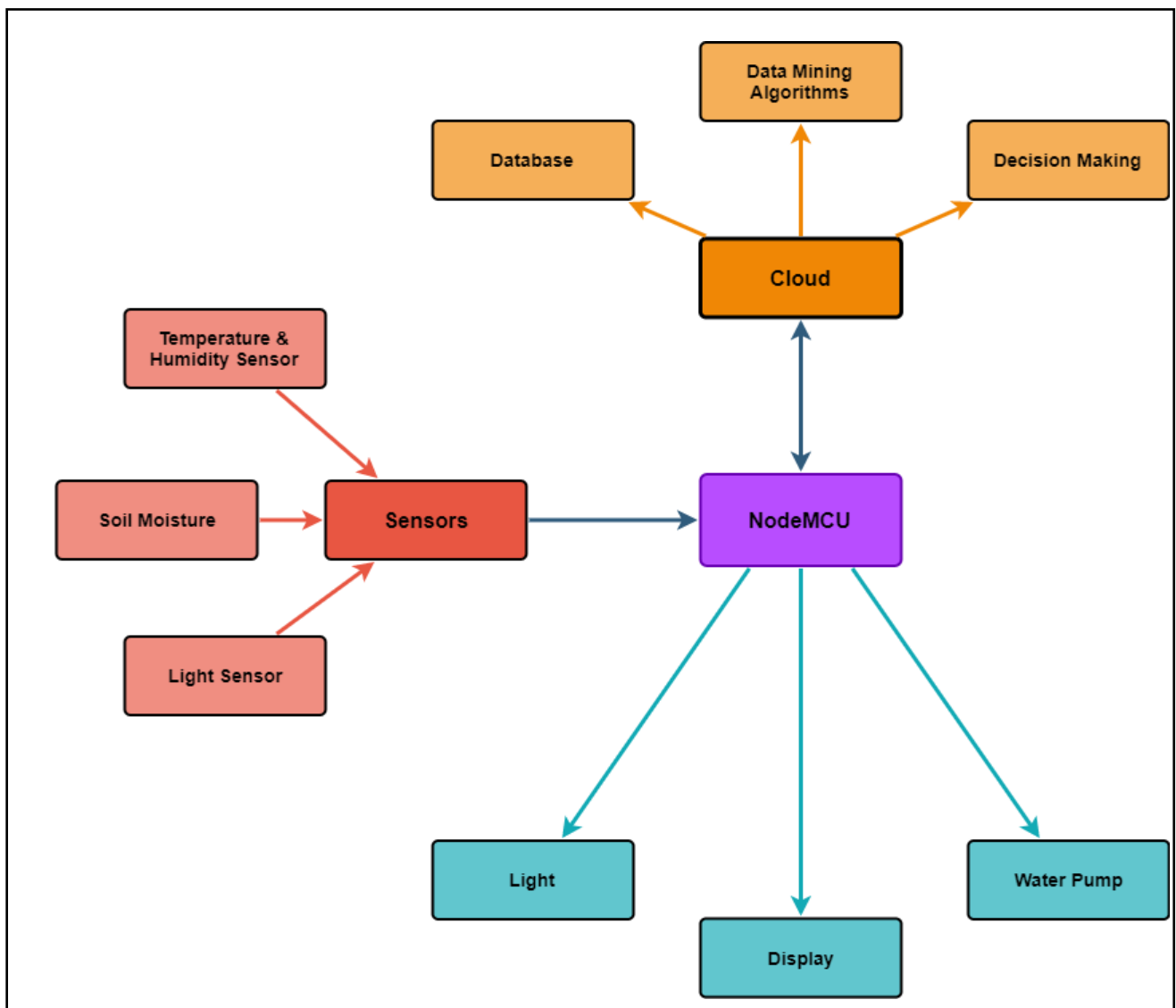


Figure 3.2: Diagram of the Device & Server

3.3.3 Circuit Diagram

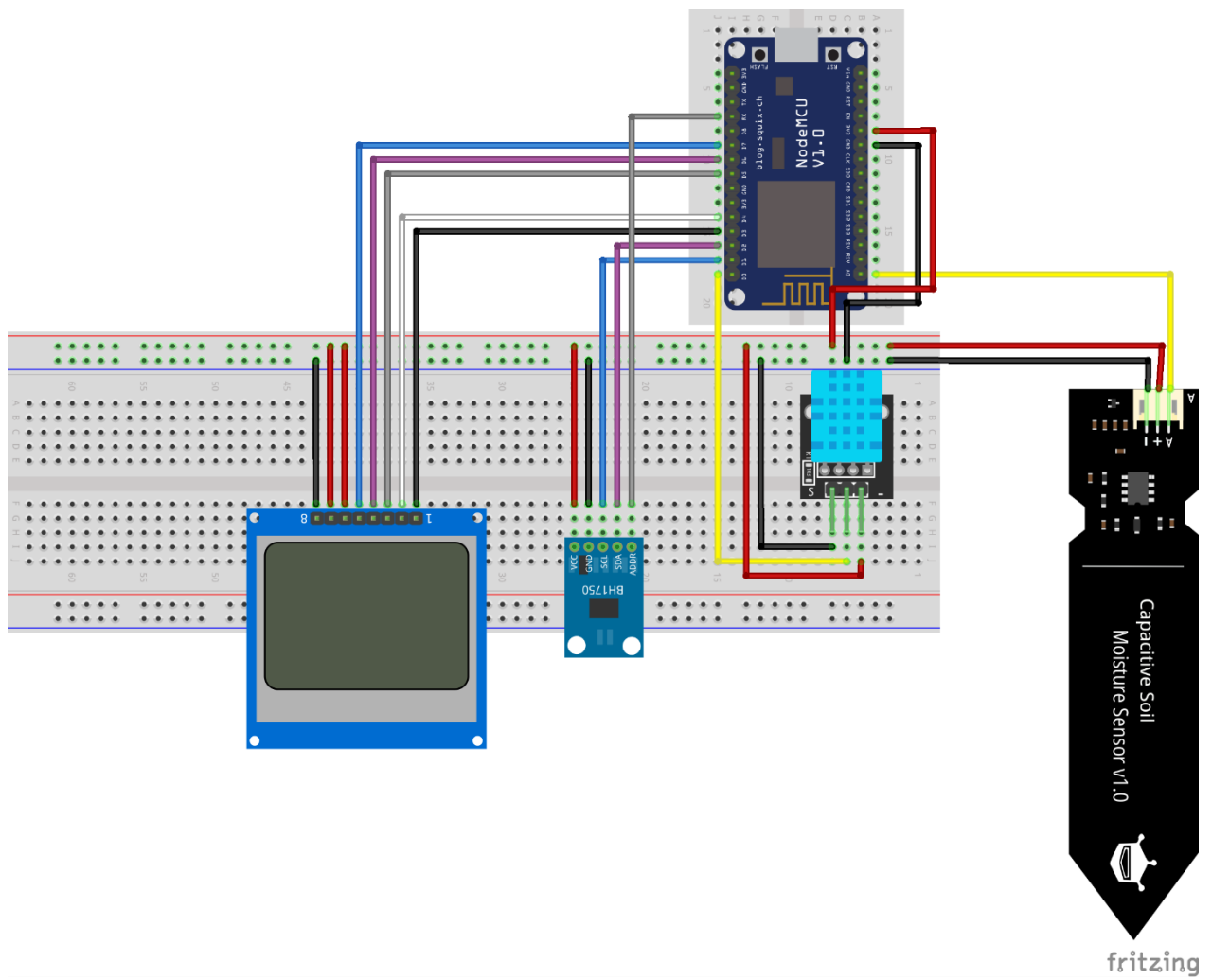


Figure 3.3: Pin Diagram of the Device

3.3.4 Statistical Analysis

For the experiment I took around Twenty-Four Hundreds of data logs. There are Five different parameters which shows how much environment is impactful in terms of plants. These parameters were same in all case. The scatter graph indicates the scatter values of light which makes the result understanding easier.

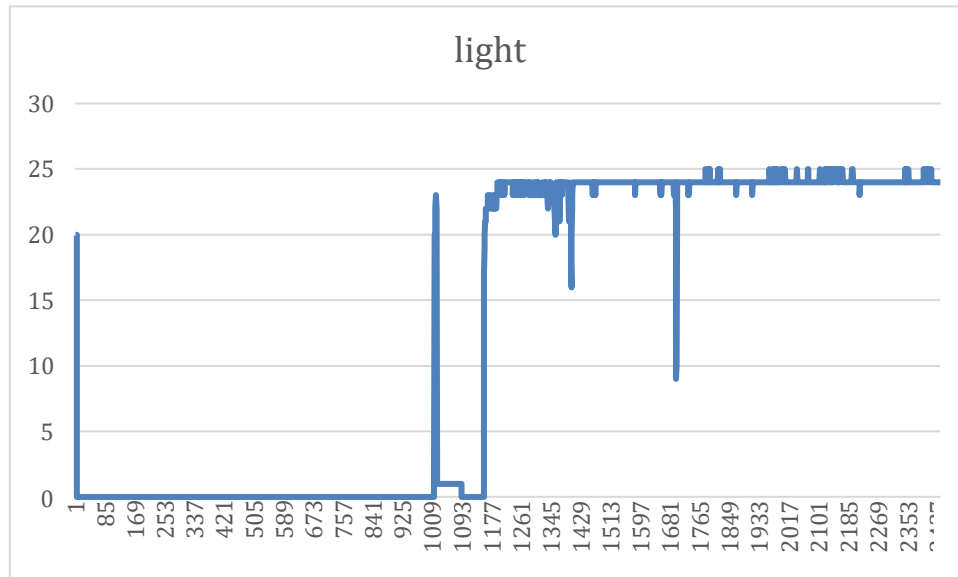


Figure 3.4: Line graph of Light Intensity

The above statistics show that the light intensity was weak at the beginning which was around 0. But later, it raises up to 25. This is a standard light intensity range limit for the plant.

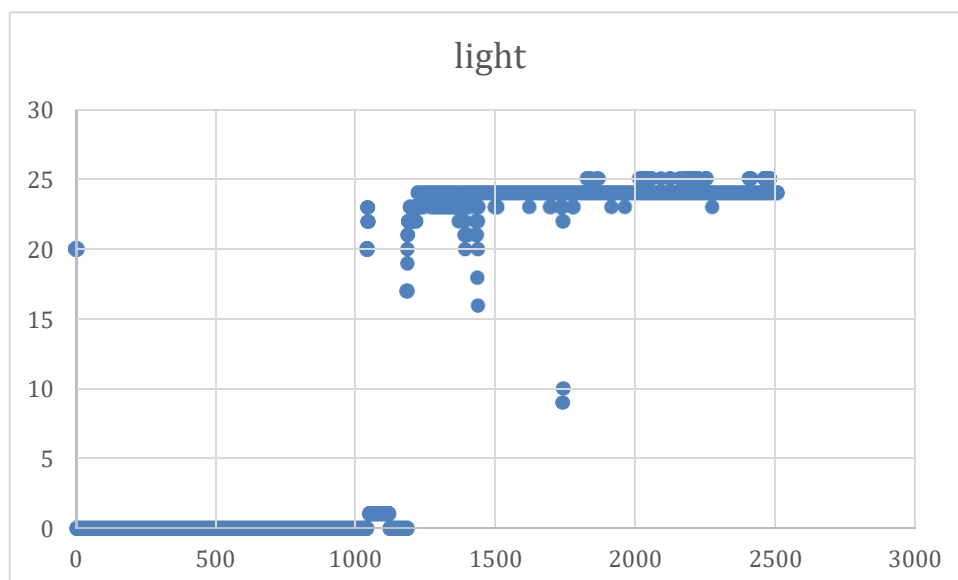


Figure 3.5: Scatter graph of Light Intensity

The same data set shows a precise scatter graph.

The following graphs represent temperature changes

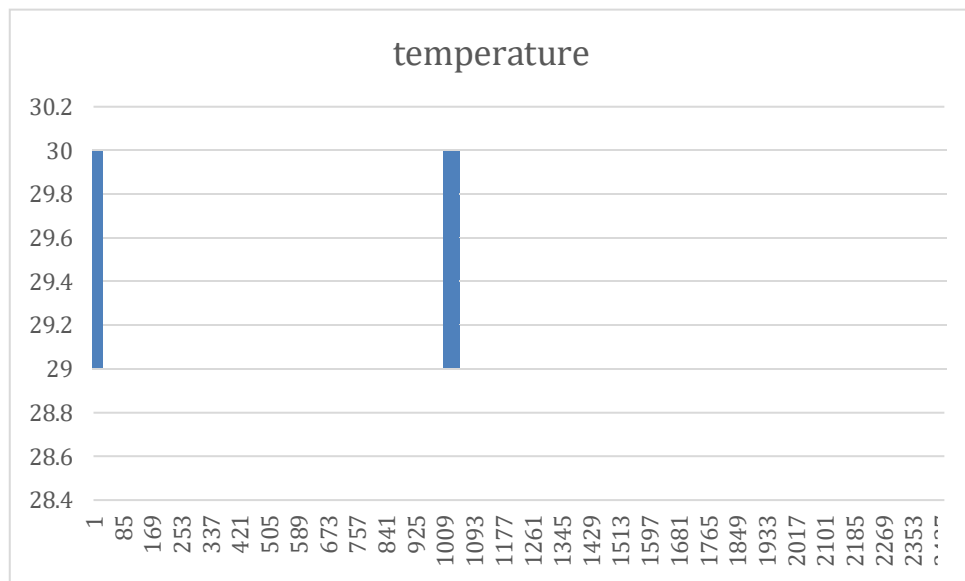


Figure 3.6: Line graph of Temperature Data in Celsius

For the time being, the temperature was fixed between 29 to 30 degree Celsius.

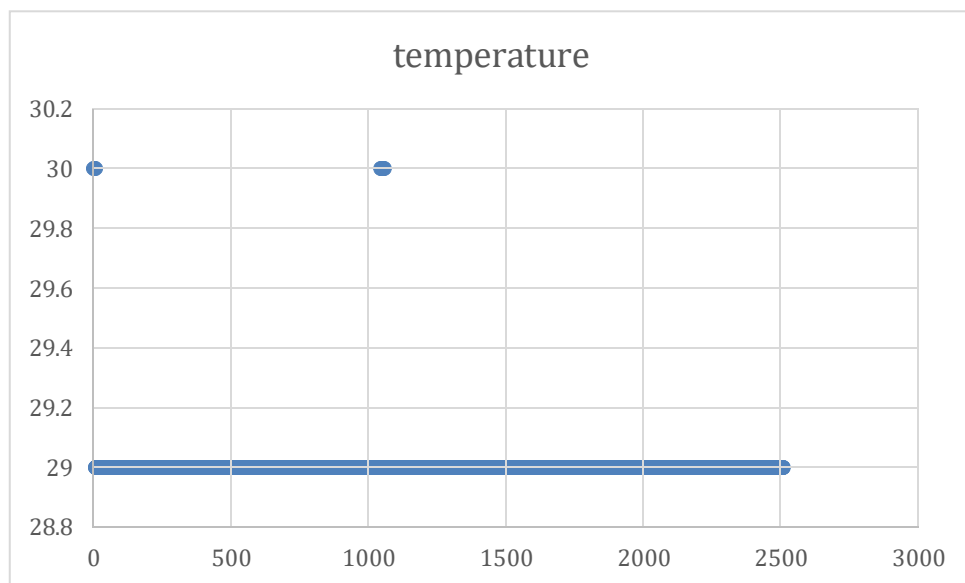


Figure 3.7: Scatter Graph of Temperature Data in Celsius

The scatter graph shows that the temperature was mostly 29 degree Celsius, and in a few cases, it rose up to 30 degree Celsius.

For Humidity variable we get the followings

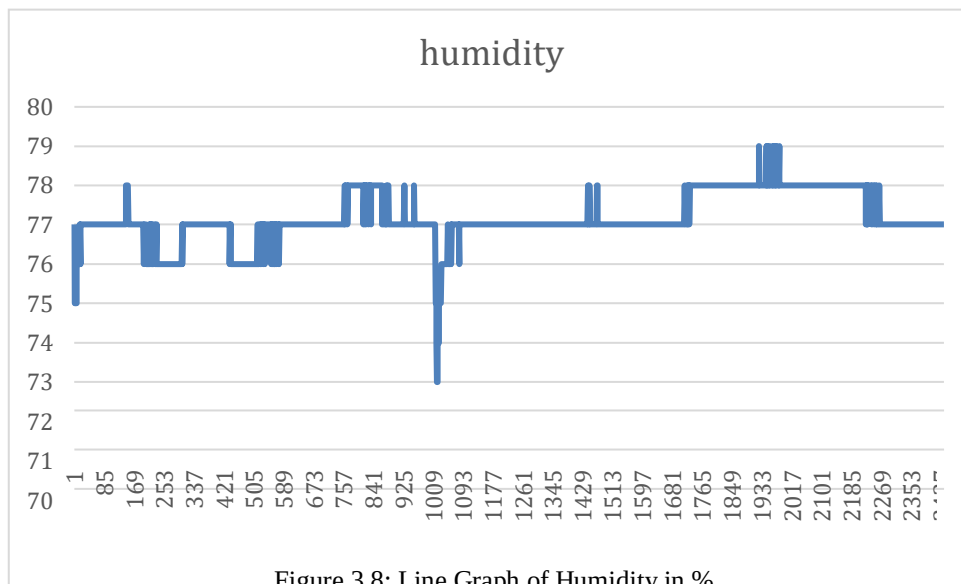


Figure 3.8: Line Graph of Humidity in %

The above figure represents the percentage of humidity of the environment. It was lowest at the 1009th data. But in rest of the cases, the percentage fluctuated in a small amount.

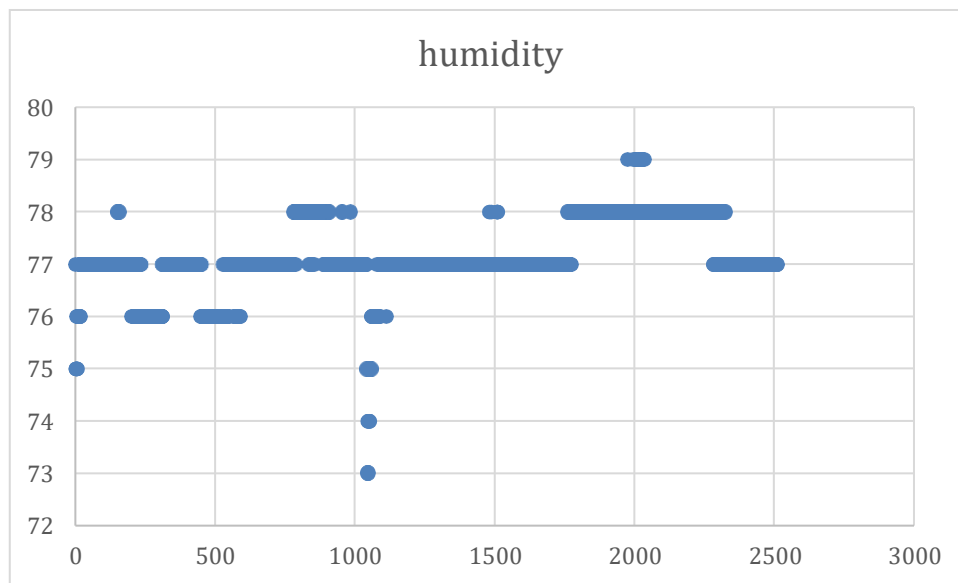
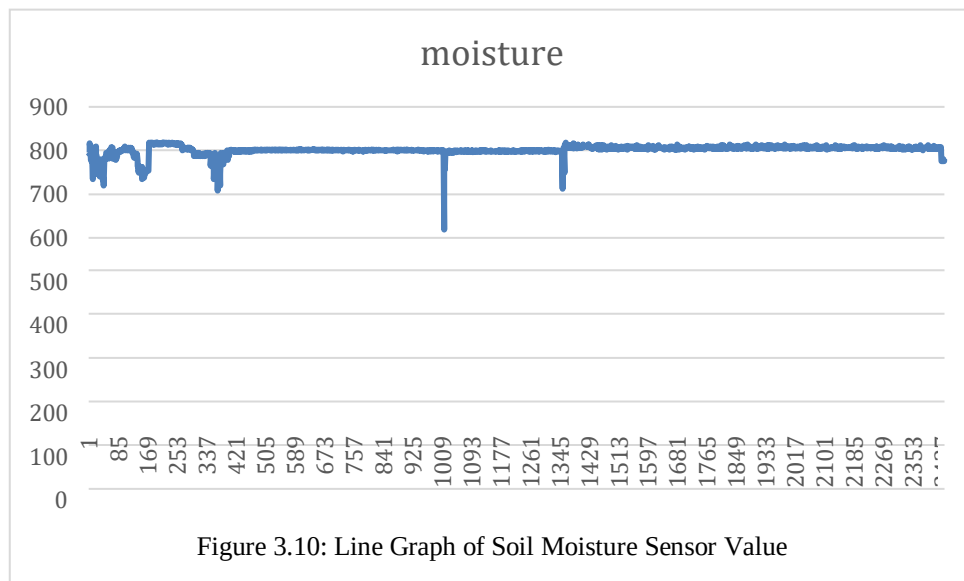


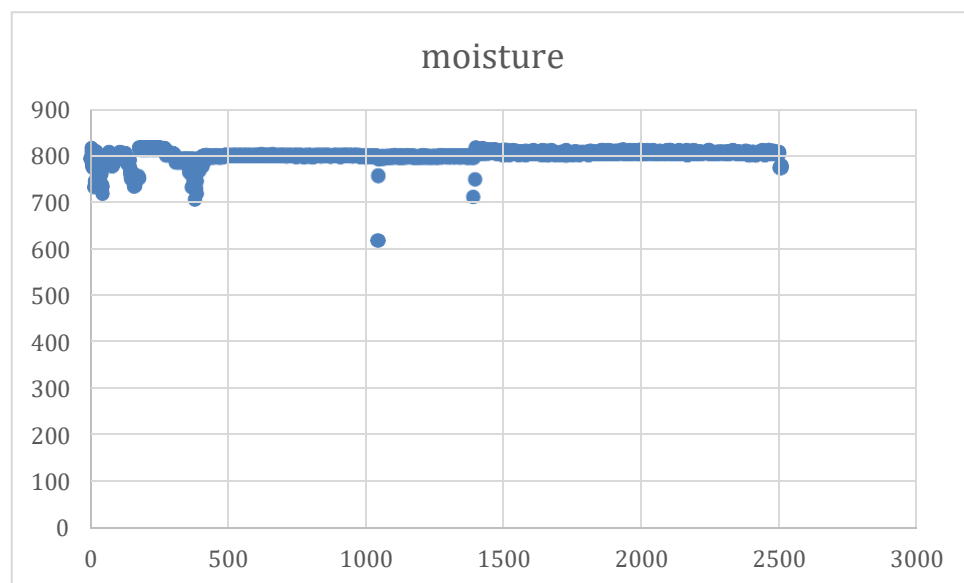
Figure 3.9: Scatter Graph of Humidity Data in %

The scatter graph of the humidity shows that it was mostly 77%.

For soil moisture, sensor returns value of ranges up to eight hundred



The soil moisture sensor value was between 600 and 850. This is a very bad indication that the plant is not getting proper water.



For around Twenty-five hundreds of data, soil is mostly in dry condition. The ratio is 2483:26. Which is very poor.

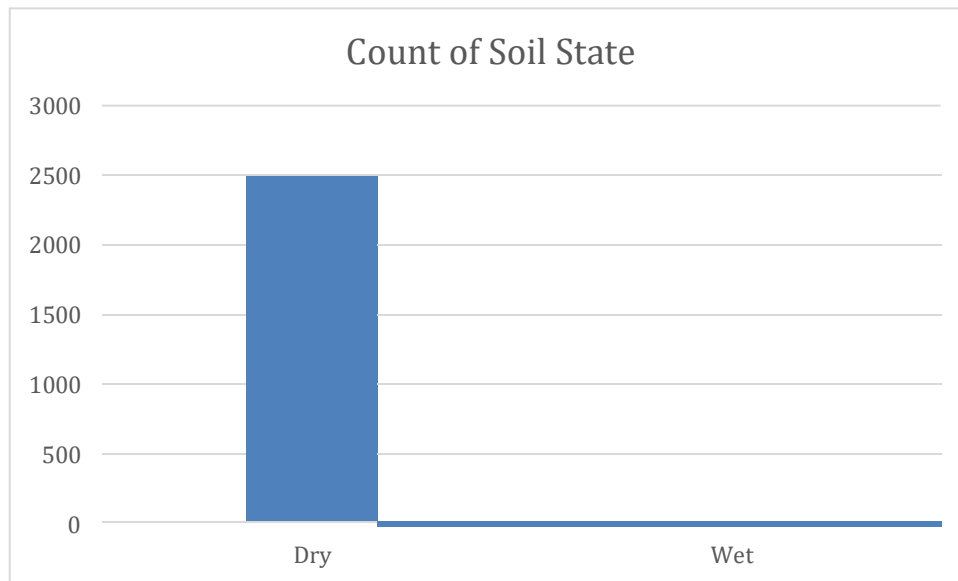


Figure 3.12: Bar chart for Soil State Count

The soil state bar graph gives more precise view of the Soil state. It shows a massive difference between how much time it was dry, and how much time it was wet. It is also noted that, the plant wasn't over watered (Very wet)

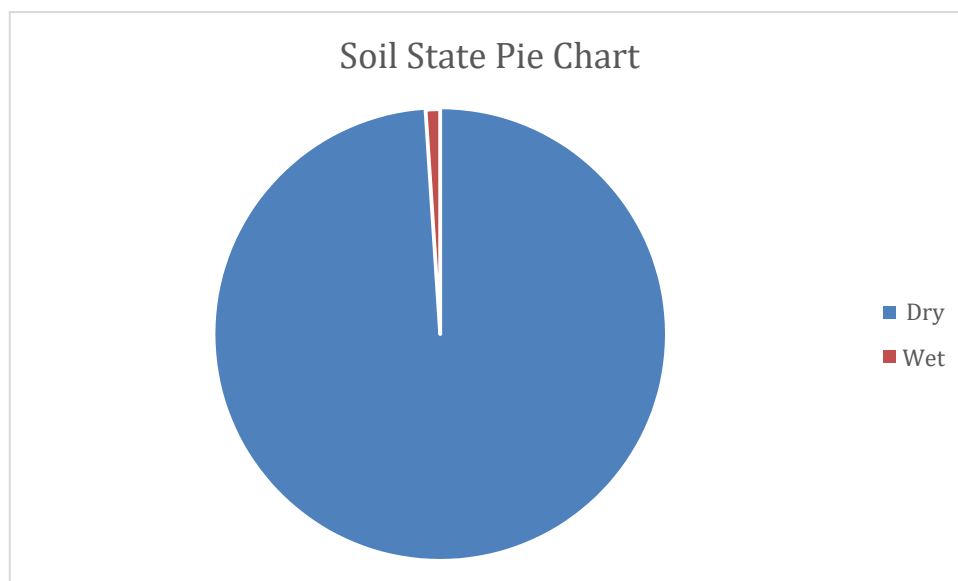


Figure 3.13: Pie Chart for Soil State Count

In the same dataset, the plant stayed alive for twenty-five hundreds of time, and died in environment of 5. Which is stated as 2504:5 ratio.

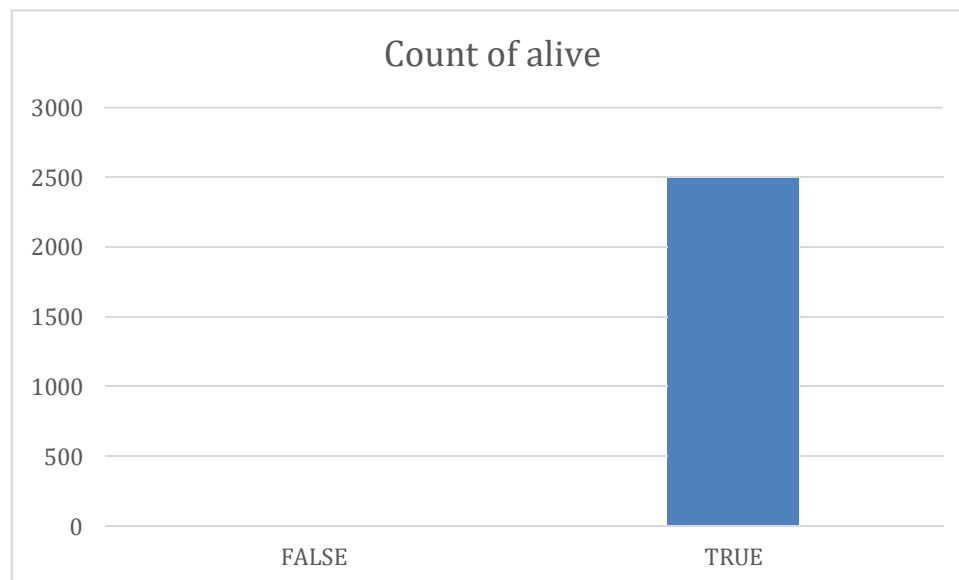


Figure 3.14: Bar Chart for Alive & Dead State

The above bar graph indicates that the plant alive value was read around 2500 times, while the death or weak data was covered for 5 times.

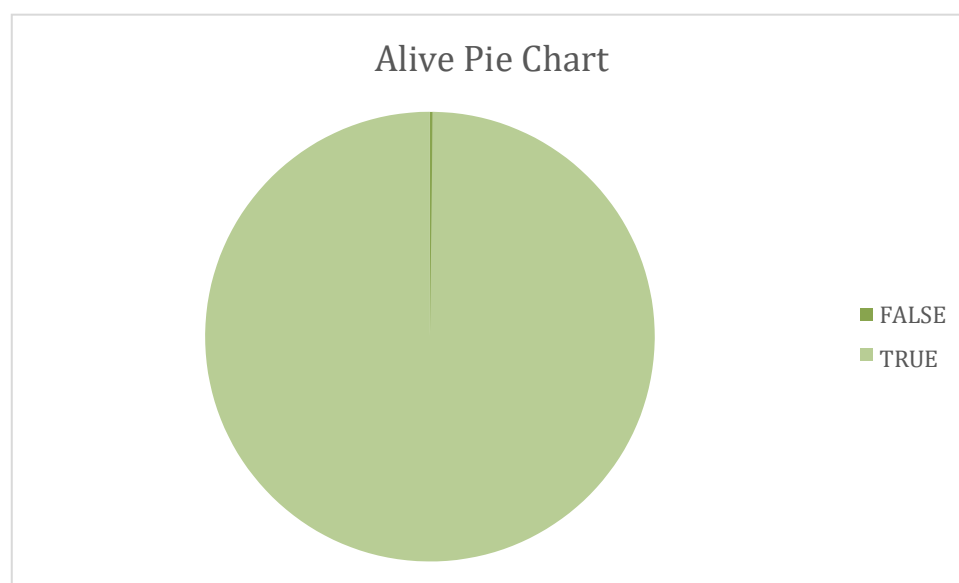


Figure 3.15: Pie Chart for Alive & Dead State

3.4 Implementation Requirements

To implement the experiment, I mainly used IoT Device. For this I followed these steps:

1. **Data Collection:** Use of IoT device to draw the environment data for the server.
2. **Sorting and remove data:** After collecting data I have to sort the data and remove irrelevant data.
3. **Algorithms:** Applying different types of algorithms to evaluate the data.

Chapter 4:

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction

For precise information, we have tried the best output possible. For this, we have chosen better algorithms as it is clearly explained in (Research methodology).

Our focus was to create a dataset where we will be able to implement algorithms and track records of environment and its impact on plants. We have also made it possible to know about barriers and possible outcomes of those barriers by this experiment.

4.2 Experimental Results

It was experimented on a plant called Madagascar Periwinkle.

PLANT TEXONOMY!	
ID	1
NAME	MADAGASCAR PERIWINKLE
SCIENTIFIC_NAME	CATHARANTHUS ROSEUS
KINGDOM	PLANTAE
SUB_KINGDOM	VIRIDIPLANTAE
SUPER_DIVISION	EMBRYOPHYTA
DIVISION	TRACHEOPHYTA
PHYLUM	SPERMATOPHYTA
CLASS	DICOTYLEDONAE
SUB_CLASS	
ORDER	GENTIANALES
FAMILY	APOCYNACEAE
GENUS	CATHARANTHUS
SPECIES	CATHARANTHUS ROSEUS (L.) G. DON

Figure 4.1: Experiment Plant Taxonomy

The experiment was first started with an Arduino UNO, LDR, dht11, soil moisture sensor, 16x2 LCD.

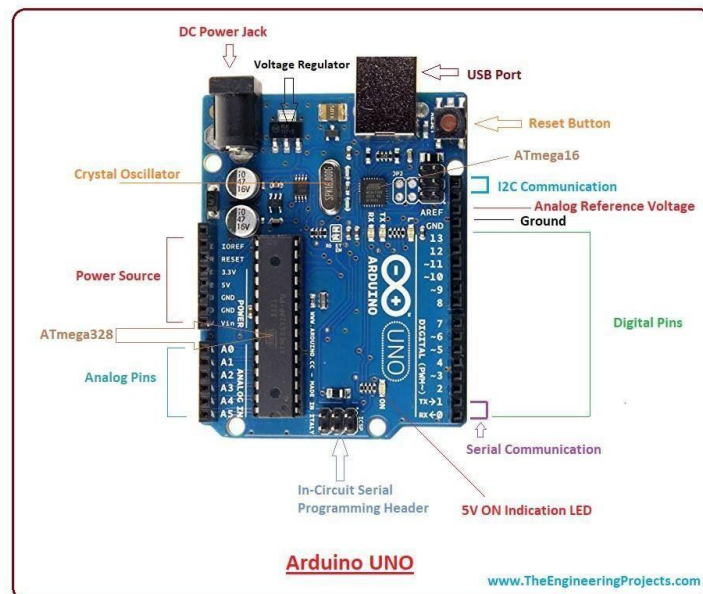


Figure 4.2: Arduino UNO Diagram

The first problem that was faced was using data log. Which means, to log every record of the plant, I had to use a Wi-Fi module. Which took a lot of time to setup, and later was diagnosed having hardware issues. So, I had to go for a smarter solution. But to not having integrated Wi-Fi module it was really tough to continue the experiment. So, I added ESP 8266 module

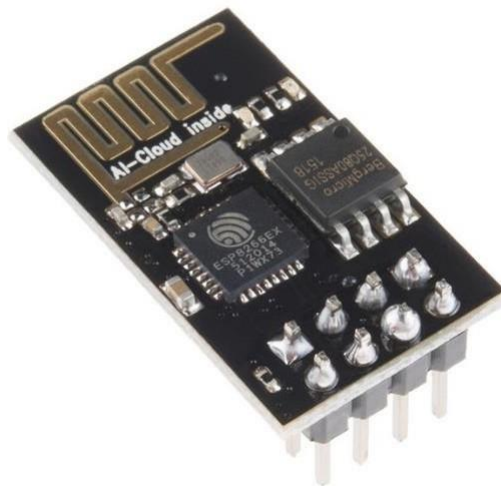


Figure 4.3: ESP8266 Wi-Fi Module

It's a nice & lightweight Wi-Fi module to integrate with Arduino Uno. After experimenting for days, I was not sure if I had hardware issues or technical difficulties. After consulting a few experts, I had to go for a smarter solution. So, I chose NodeMCU.

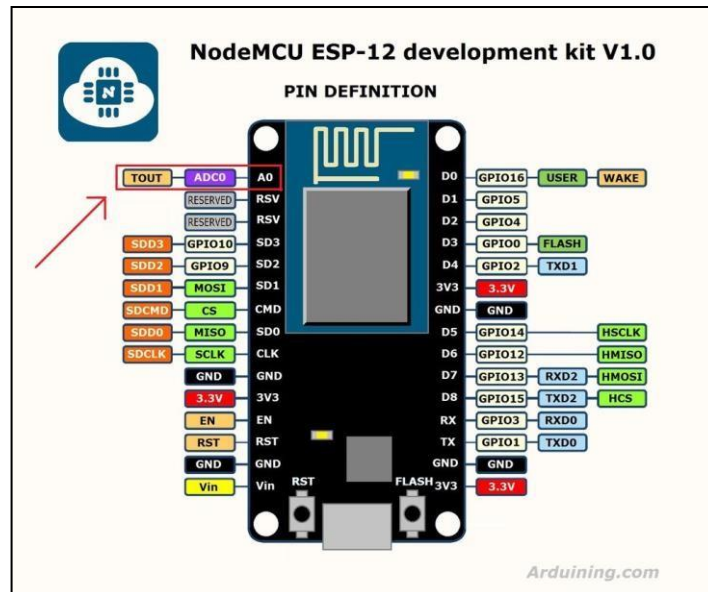


Figure 4.4: NodeMCU Pin diagram

I chose Arduino UNO. NodeMCU, which had a built in Wi-Fi module. It had another drawback. The NodeMCU had only 1 Analog pin. But LDR, Soil moisture sensor and the LCD used 4 Analog pins. So, it was impossible to implement the project in NodeMCU. On the contrary Arduino had 5 analog pins, which was perfect to implement the project.

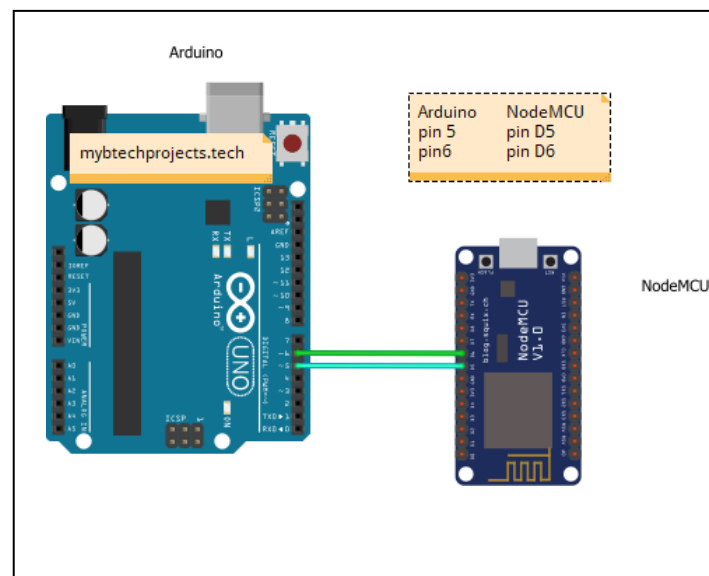


Figure 4.5: Serial Communication between Arduino UNO & NodeMCU

At first, it was okay. The data were fine, values were passing & posting on server. But later, I found out that the serial communication used character by character data transmission. As a result, the communication had to have real time value transmit and receive feature. It was really disappointing sometimes, because there were data missing with Character.

```
Ex : String str = "Bj?rk?oacute?";
```

Figure 4.6: Data Missing due to Serial Communication

A blue printed circuit board (PCB) for a BH1750 digital light sensor. The board is populated with several surface-mount components, including the BH1750 sensor chip, a 10k pull-up resistor, and two 0.1µF capacitors. The text 'BH1750' is printed in white on the PCB. A black 4-pin header is attached to the bottom of the board.

analog. Which reduced the n

File Edit Sketch Tools Help

WiFi-GET-Call | Arduino 1.8.9

WiFi-GET-Call

```
#include <ESP8266HTTClient.h>

//selected serial port
SerialTimeoutException: Write timeout does not exist or your board is not connected

Sketch uses 287632 bytes (27%) of program storage space. Maximum is 1044464 bytes.
Global variables use 27156 bytes (33%) of dynamic memory, leaving 54764 bytes for local variables. Maximum is 81920 bytes.

esptool.py v2.6
2.6
esptool.py v2.6
Serial port COM?
Connecting....
Chip is ESP8266EX
Features: WiFi
MAC: 84:0d:8e:86:3c1a9
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected flash size: 4MB
Compressed 291792 bytes to 210262...

Writing at 0x00000000... (7 %)(Traceback (most recent call last):
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\upload.py", line 25, in <module>
    esptool.main(fakeargs)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 2698, in main
    operation_func(esp, args)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 2158, in write_flash
    esp.flash_defl_block(block, seq, timeout=DEFAULT_TIMEOUT * ratio * 2)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 104, in inner
    return func(*args, **kwargs)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 632, in flash_defl_block
    self.ESP_FLASH_DEFL_DATA, struct.pack('<IIII', len(data), seq, 0, 0) * data, self.checksum(data), timeout=timeout)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 354, in check_command
    val, data = self.command(op, data, chk, timeout=timeout)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 322, in command
    self.write(pkt)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\esptool\esptool.py", line 285, in write
    self._port.write(buf)
  File "C:\Users\IN\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\2.5.1\tools\pyserial\serial\serialwin32.py", line 323, in write
    raise serial.SerialTimeoutException: Write timeout

The selected serial port serial.util.SerialTimeoutException: Write timeout
does not exist or your board is not connected
```

NodeMCU1.0 (ESP-12E Module)



Figure 4.9: Port Issue

The RX pin is normally used to transfer the program from computer to the device. Due to using it for light sensor, there was an error while uploading. A solution was to unplug the pin while uploading the code. And it worked very well. [10]

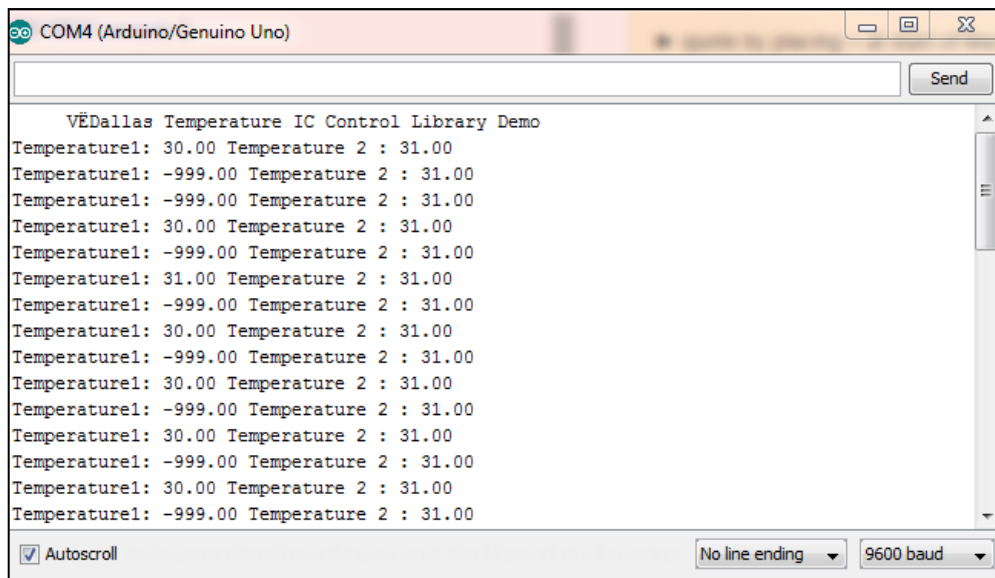


Figure 4.10: DHT11 Sensor Fault Reading

There was another error with the DHT11 sensor. While debugging through serial monitor, it was found that the sensor would read fault data if the sensor reads data within 1203 Milli seconds. It means that, it only returns valid data after 1203 Milli Seconds.

The soil moisture sensor, which was used earlier, seemed to be corroded after sometime. Which caused non valid values.

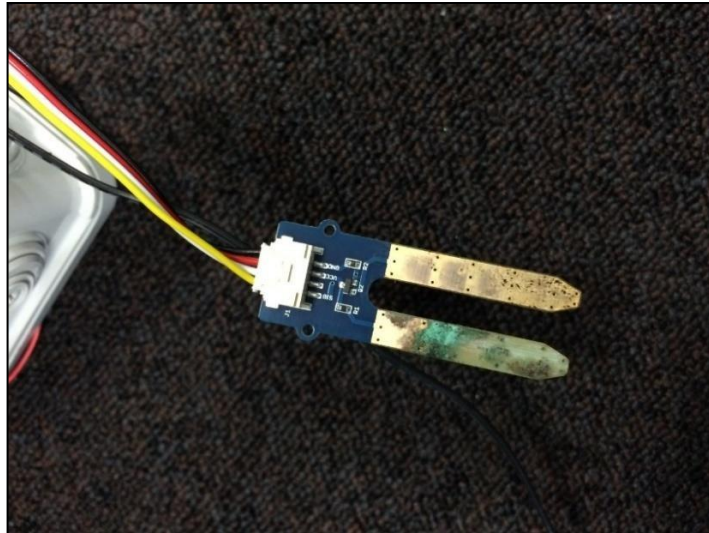


Figure 4.11: Soil Moisture Sensor Corrosion

After researching a lot online, it was found that this module had this issue. It was drawback of the device [11]. There was also another solution. Which is Capacitive Soil moisture sensor by DF Robot.



Figure 4.12: Capacitive Soil Moisture Sensor as a replacement

It is very precise and powerful. And it does not get corroded over time.

I still needed three analog pins. Also, the 16x2 LCD module was too small to print all the values in such a small display unit.

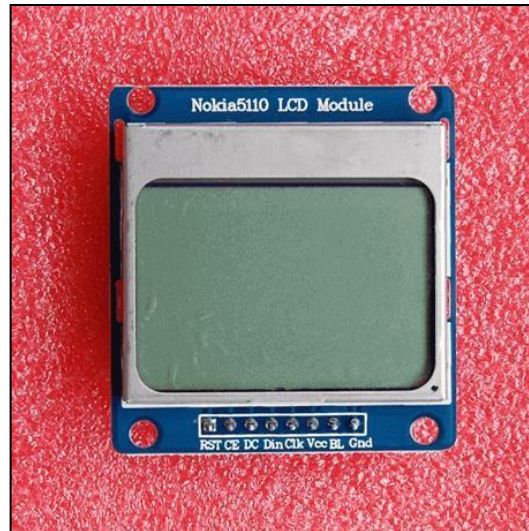


Figure 4.13: Nokia 5110 LCD as a Replacement for 16x2 LCD

And the LCD was replaced using Nokia 5110 LCD. It used 5 digital pins. The pin issues were resolved eventually and the NodeMCU was usable.

Eventually the Nokia LCD Module was found that the backlight was damaged and was not usable.



Figure 4.14: LCD Without Backlight

It was because of not using resistance in the first place.

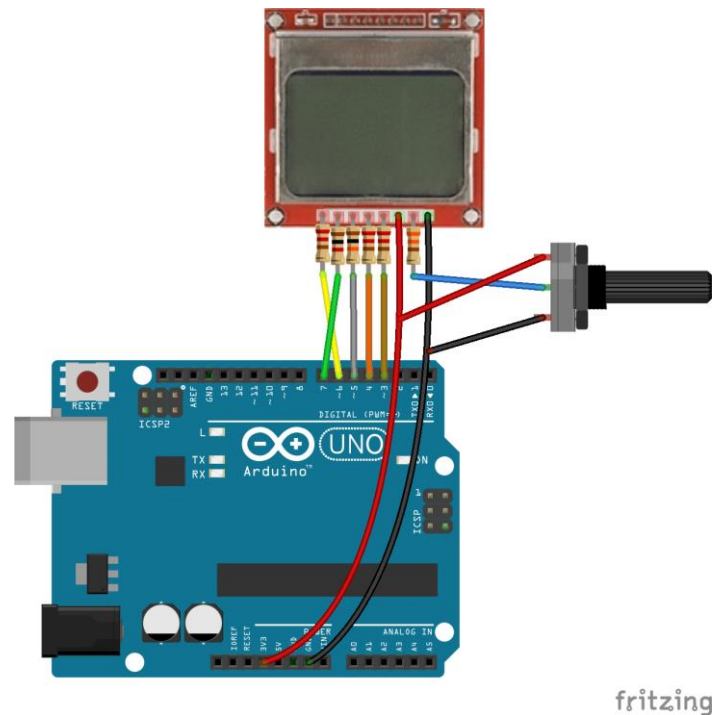


Figure 4.15: LCD without Resistor

So, the rest of the project was done without LCD module. Which was sometimes difficult to debug without computer. The only option was the Serial monitor of Arduino IDE.



Figure 4.16: Serial Data viewer as alternative to LCD

And I took a few days reading with the plant called *Cantharus Roseus*. Unfortunately, the plant died within a few days. And I could not think it would die so early. So, I did not even take a picture of it.



Figure 4.17: Dead Plant

Then I came to think of it, If I implemented the device earlier, the plant could have survived, because then I could have learnt about its necessary element values. Later I realized It was due to lack of water.

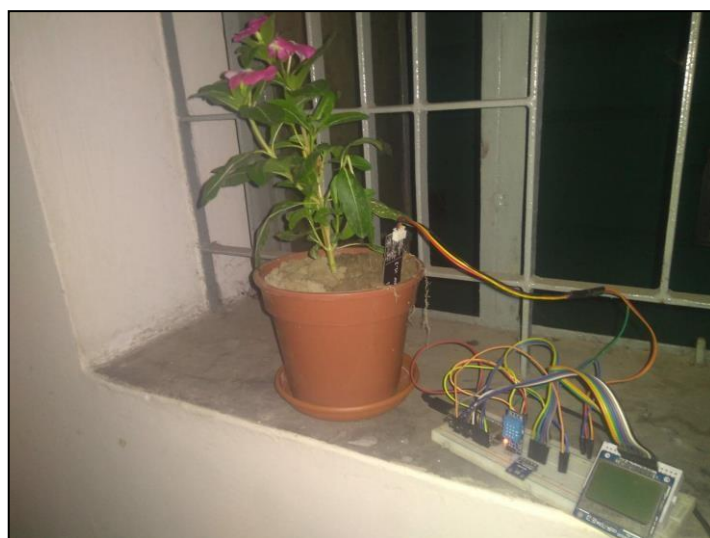


Figure 4.18: Replacement

So, I need to replace the dead plant with a fresh one to continue my research.

The taxonomy of the plant is as follows:

Table 4.1: Plant Taxonomy

Name	Madagascar Periwinkle
Scientific Name	Catharanthus Roseus
Kingdom	Plantae
Sub Kingdom	Viridiplantae
Super Division	Embryophyta
Division	Tracheophyta
Phylum	Spermatophyta
Class	Dicotyledonae
Order	Gentianales
Family	Apocynaceae
Genus	Catharanthus
Species	Catharanthus Roseus (L.) G. Don

After completing the device part, the server part came. The server was hard to implement. It raised hundreds and hundreds of errors. This was a crucial part to handle with the device. The first server was built using Django.

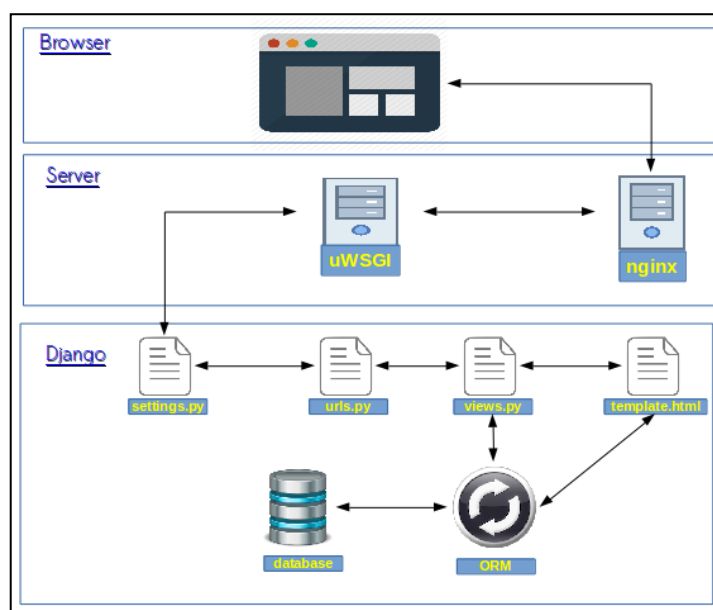


Figure 4.19: Django Architecture

Django is a Python based Web Framework. The Architecture is primarily based on Model View Controller (MVC) pattern. Django MVC Architecture solves lots of problems which were there in the traditional approach for web development [12]. While testing with a Boolean value alive, the server couldn't handle the response. Also, the relationship was a bit difficult to handle sometimes. After debugging the problem, it was decided to roll back to another server called Laravel of PHP.

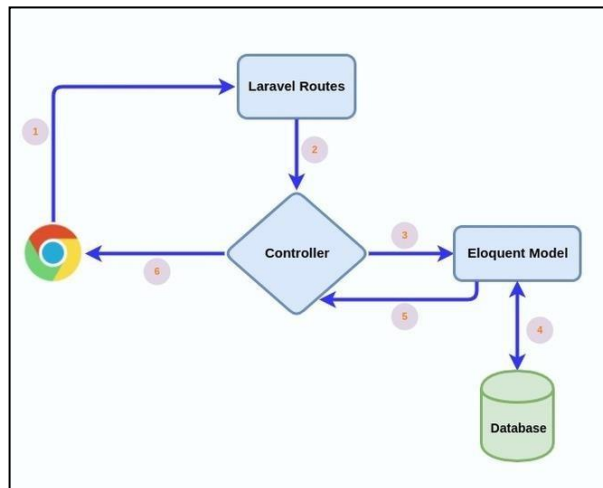


Figure 4.20: Laravel Architecture

Laravel is a php based web application framework having expressive, elegant syntax. They’ve already laid the foundation — freeing developers to create without sweating the small things [13].

One of its strongest point is to use Eloquent relationships for database. Also, the REST Api is very powerful and easy to handle.

There came another issue, called data filtering. While trying to work with lots of datasets, it was hard to implement the dataset directly using php of Laravel framework. On the other hand, python had its own built in library called pandas and NumPy. It can easily handle process massive data and return a precise output. So, I started a cross server communication. At first, I tried with Flask, which had a clean and simple architecture.

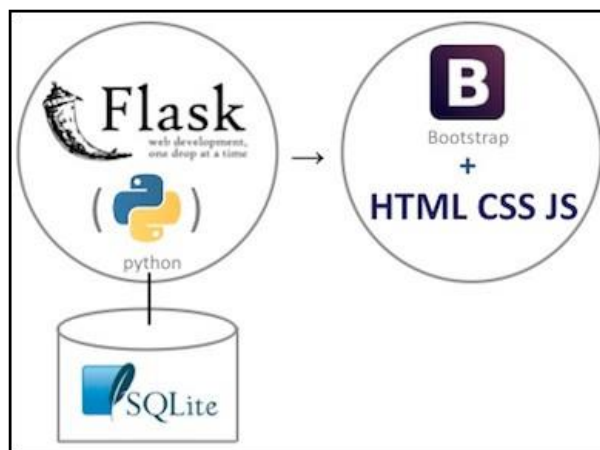


Figure 4.21: Flask Architecture

Flask is a straightforward and minimalist internet framework written in Python. it has no information abstraction layer, form validation, or different elements that an online app would possibly need. However, Flask is enhanced with extensions that may add application options as if they were enforced in Flask itself. It's open source beneath a BSD license.

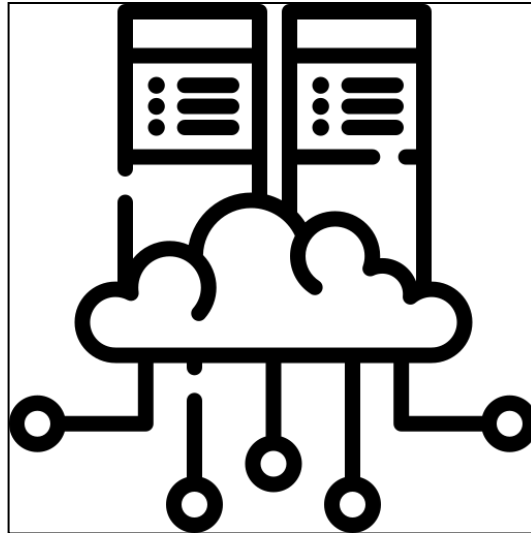


Figure 4.22: Cross Server Communication

While experimenting with cross server communication, there was a small issue. The Laravel server request and response from Django server was taking more time than expected. As a result, the server responded unexpectedly. The whole process of data mining was later tried to be implemented with the Laravel. Although the advanced data mining techniques which was offered by Python was not available. For the time being I had to stick with the basic operation of the range of data. But in future, when the whole process will be done in the Python server, will be more effective and accurate.

4.3 Descriptive Analysis

The data collection procedure is straightforward and simple. The IoT device (Eco Friend) collected raw data from the home environment using 3 different sensors called Light sensor,

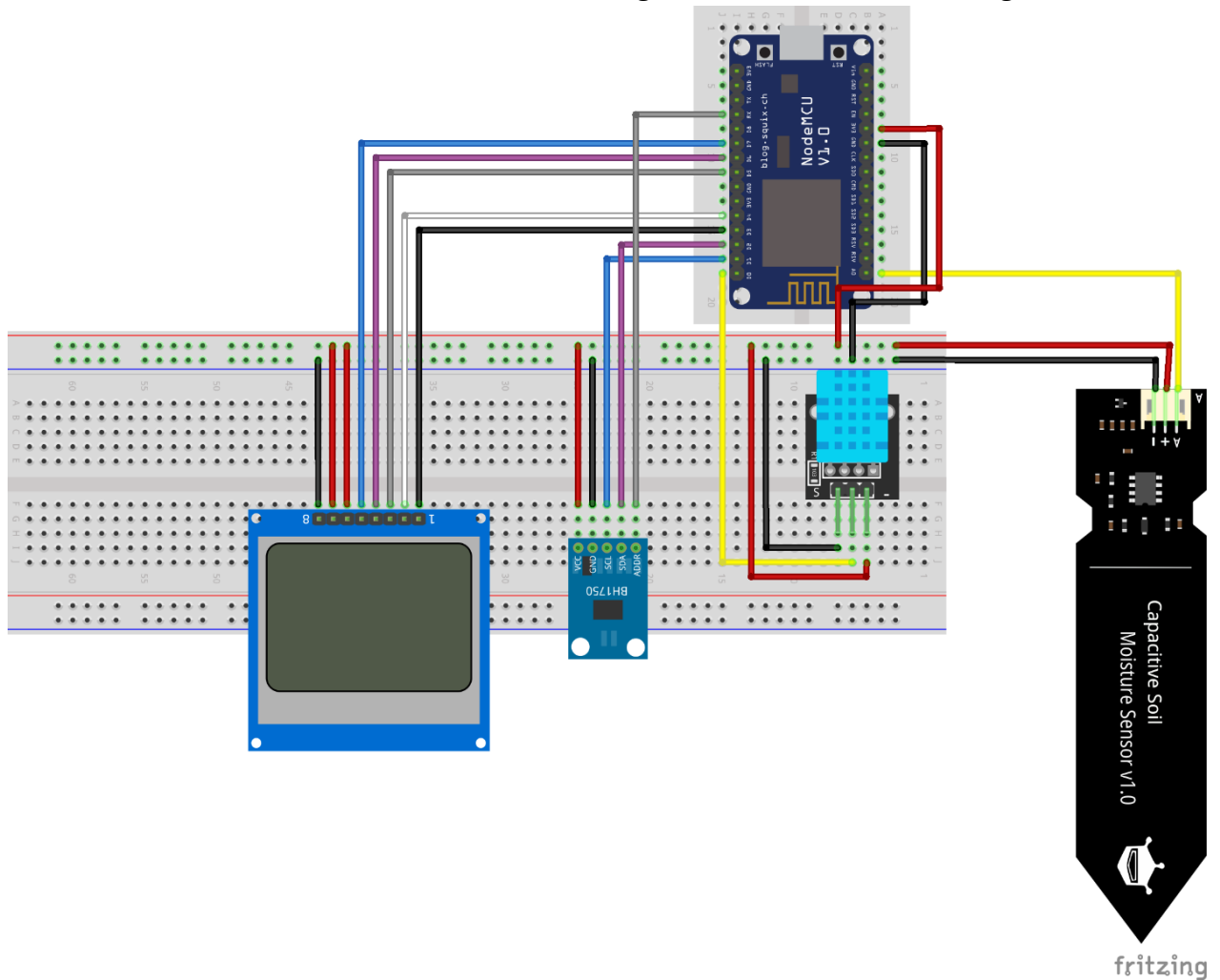


Figure 4.23: Circuit Diagram

Temperature & Humidity sensor, Soil moisture sensor. These sensors collect basic data from the home environment.

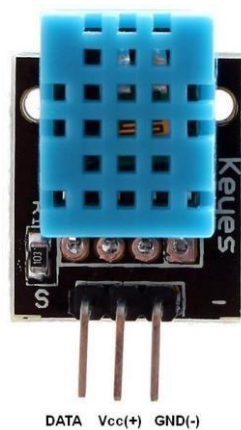


Figure 4.24: DHT11 Pin Diagram

The pins of the DHT11 sensor is connected with the Arduino as follows

Table 4.2: DHT11 Interfacing Table

DHT11	NodeMCU
Data	D0
VCC	+3V
GND	GND

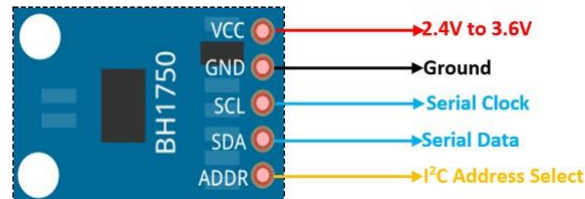


Figure 4.25: BH1750 Pin Diagram

The light sensor uses 5 different pins. The connection of the sensor with the NodeMCU board is as follows.

Table 4.3: BH1750 Interfacing Table

BH1750	NodeMCU
VCC	+3V
GND	GND
SCL	D1
SDA	D2
ADDR	Rx

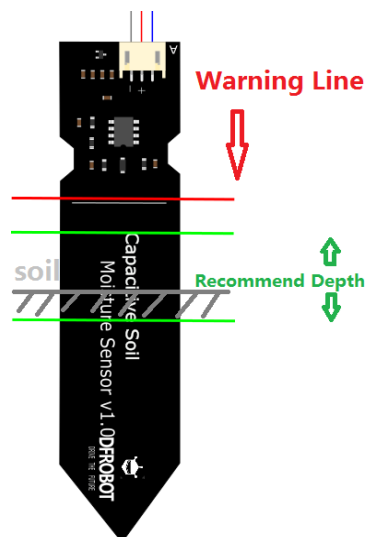


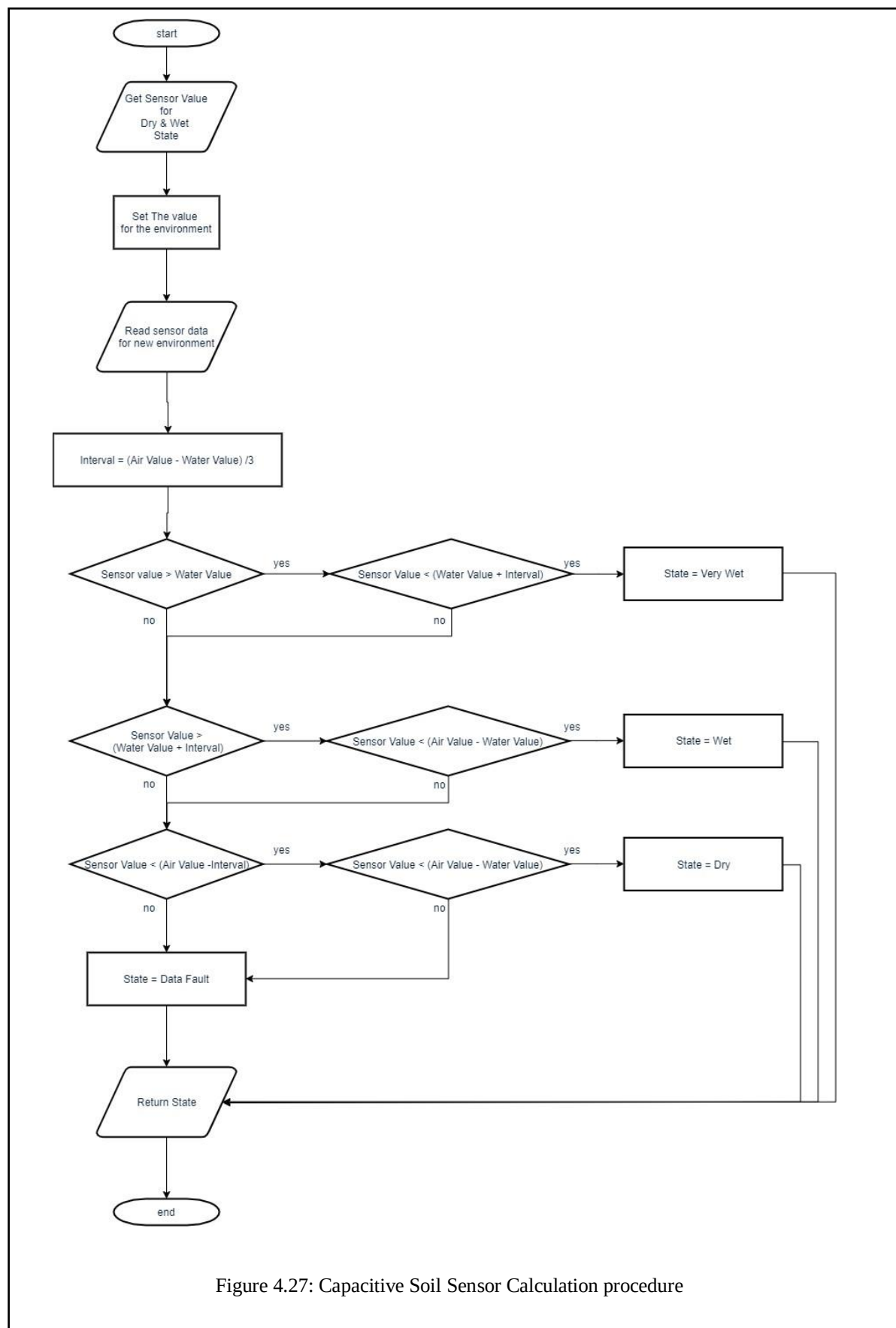
Figure 4.26: Capacitive Soil Moisture Sensor Depth Recommendation

The soil sensor is used to get the moisture value from the environment. The recommended depth is mentioned in the above figure to get more accurate value. The pins are connected as follows:

Table 4.4: Capacitive Soil Moisture Sensor Interfacing Table

Capacitive Soil Sensor	NodeMCU
VCC	+3V
GND	GND
Analog Pin Out	A0

Checking the soil state was a tricky move, the below conditions are implemented:



A data input is taken to read the air value and water value. In our case, Air value was 890, Water value was 470. An interval value is also taken which refers to $interval = \frac{AirValue - WaterValue}{3}$

The conditions were checked as follows:

If (sensor Value > Water Value) and (Sensor Value < (Water Value + Interval)) Then the soil state is Very Wet.

Else If (Sensor Value < Air Value) and (Sensor Value > (Air Value – Interval)) Then the soil state is Wet.

Else If (Sensor Value < Air Value) and (Sensor Value > (Air Value – Interval)) Then the soil state is Dry.

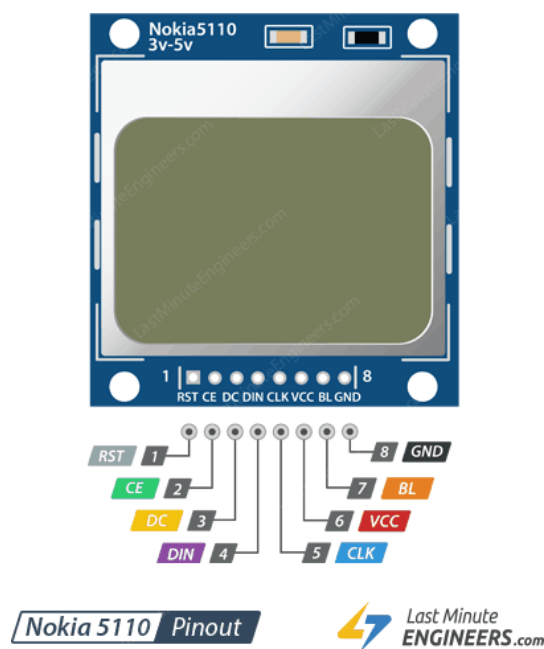


Figure 4.28: Nokia 5110 LCD Pinout

The pin diagram is as follows:

Table 4.5: Nokia 5110 Interfacing Table

Nokia 5110 LCD	NodeMCU
RST	D3
CE	D4
DC	D4
DIN	D6
CLK	D7
VCC	+3.3v
BL	+3.3v
GND	GND

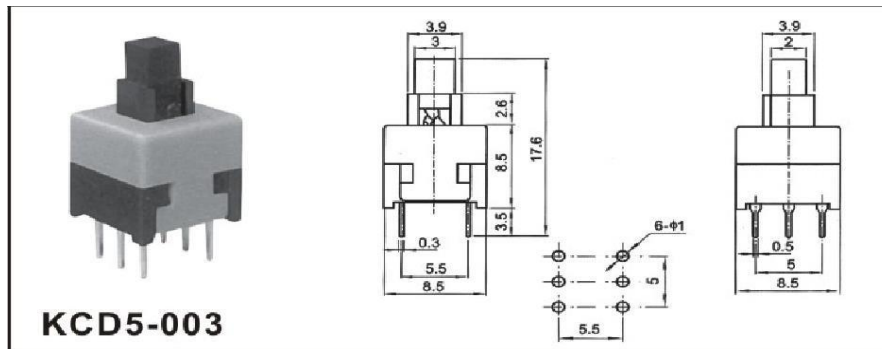


Figure 4.29: Push Button Diagram

A push switch is used to create dataset. A connection indicates the plant is alive, on the other hand disconnect indicates the plant is dead or getting dried out. The pin diagram is

Table 4.6: Push Button Interfacing Table

Button	NodeMCU
Point 1	+3.3V
Point 2	D8

After collecting all the data from the sensors, a get request is formed consisting of the values

- Light
- Temperature
- Humidity
- Soil State
- Soil Moisture
- Alive State
- Plant ID

Initially I assumed the plant id is 1. Which is Madagascar Periwinkle. The server Lists are:

- Laravel
 - Development Server: <https://laravelpi.dev/> or <http://localhost:80/>
 - Production Server: <https://laravelpi.herokuapp.com/>
- Django
 - Development Server: <http://djangpi.dev> or <http://localhost:8000/>
 - Production Server: <http://djangpi.herokuapp.com/>
- Flask
 - Development Server: <https://flaskpi.dev/> or <http://localhost:8800/>
 - Production Server: <https://flaskpi.herokuapp.com/>
- API Routes
 - “/”: Home (Basic GUI) with Datamine Result
 - “plant/”: All Plant View
 - “/create”: Create New Plant
 - “/{id}”: View specific plant information
 - “/edit/{id}”: Edit Specific plant information
 - “/delete/{id}”: Delete plant
 - “/plantlog/{id}”: View data log for specific plant

- “/datalog/”: All data log view
 - “/{id}”: View specific plant information
- “/api/”: Return data in json format
 - “datalog/”: All data logs
 - “/create”: Handle request for creating new Data Log
 - “/{id}”: Return json data for Specific Data Log
 - “/update/{id}”: Update Data Log Information (Not directly used without special case)
 - “/delete/{id}”: Delete a Data Log (Not used)
 - “plant/”:
 - “/create”: Create New Plant
 - “/{id}”: Return json for Specific Plant
 - “/update/{id}”: Handle request and update Plant information
 - “/delete/{id}”: Handle the request for deleting a Plant information
 - “/datamine/{id}”: Return json of processed database with minimum maximum range of values

The datamining process is followed by the below steps:

1. Data is filtered for error
2. Pandas library is used to process the complete list of arrays
3. The data for which plant was alive, is processed
4. Using filtering techniques, the soil state value, which is Dry, Wet, Very Wet, is filtered and converted to number
5. All the values for a fixed column are processed, and regarded as range of values
6. Minimum maximum values are returned
7. And the final output looks like the below figure

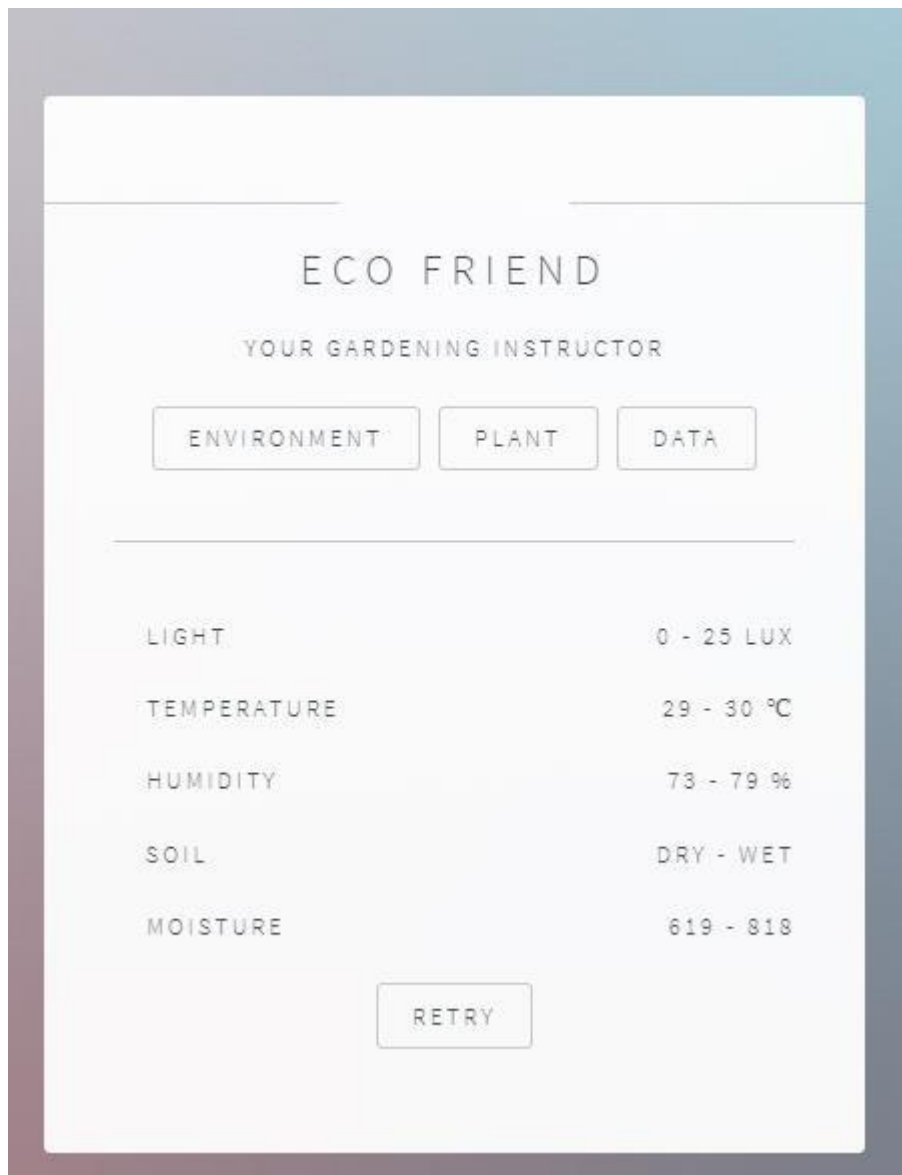


Figure 4.30: Final Datamined Output

4.4 Summary

The device might be small in size, but its impact can be greater than imagination. Think of a greener world, trees everywhere. You don't need to study agricultural science to know basics about trees. Because eco friend has got your back.

The automatic plant observation system has recently attracted tremendous interest thanks to the potential application in rising technology. More importantly, this technique is used to enhance the performance of existing techniques or to develop and design new techniques for the growth of plants. In this world, there was a time when most places of the land were covered by forest. People used to believe that the great forest will never vanish. But the rapid urbanization and cutting of trees affecting the forest and is deteriorating the environment and ecological balance [14].

Chapter 5:

SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

5.1 Summary of the Study

The goal of our research is to build an advanced gardening support system called the Eco Friend. Eco Friend supports gardeners in making planting decisions. With increasing numbers of people in Bangladesh and increasing interest in healthy eating, home gardening is becoming more and more popular. However, it is difficult for beginners to plan gardens because of the wide range of factors to consider, such as soil, weather and climate. Eco Friend supports decisions about what to plant where with data collected by sensors and organized in a database. The concept of Eco Friend is based on precision agriculture, which uses information technology to bring together data from multiple sources to support decisions associated with crop production. Eco Friend uses sensors to collect data in the real world that it then plots in a cloud database. Various models then forecast possible fertilizer needs, pest control needs, and yields. Growers carry out actions based on the forecasts and evaluate the results. SG allows growers to set their own goals. Growers might pursue yield, or particular vegetables, or a mix of flowers, herbs, and vegetables, or a balance of fragrant and beautiful plants. SG uses computer models to forecast the outcomes of management decisions, and the technology of Data Mining (AR) to show the grower the predicted results [15].

5.2 Conclusions

Analyzing and Data mining using IoT with the assistance of a NodeMCU controller helps to ease the foremost tedious job of farming for plant lovers who are in a very time of rush. This method monitors numerous garden parameters and store the data in details of garden through the device. It conjointly helps to unravel several problems occurring within the existing plant watering and farming system. It helps to avoid wasting water and utility bills. Plant observance and good farming mistreatment IoT with the assistance of the NodeMCU controller can bring a lot of convenience and luxury to individuals' lives for taking care of their garden. The user will management and monitor the setting of the garden mistreatment internet application. The controller during this system (Raspberry Pi) provides associate economic and economical platform to implement the plant observance and good farming system mistreatment IOT. the most advantage of the good farming system is that the user will monitor the garden mistreatment the net from so much distances throughout leisure or whenever necessary.

5.3 Recommendations

I recommend,

- There must be work to understand how the device works.
- As the device is still not user friendly, more guidelines and precise information is required to use the device.
- Gardening should be encouraged amongst people.
- Raise awareness about tree plantation.

5.4 Implication for Further Study

- **Commercial field:** Addressing the operations that can be done with the device can help to create more innovative devices based on Home gardening system.
- **Potential Economic Growth:** Helping home gardening can result into more efficient and healthier citizen life. Which can result into potential economic growth.
- **Plant Monitoring:** Using a few more modules and devices, the device can be turned into a plant manager system, where it will recommend users plants best for his environment. It will notify users for any suspicious changes or behavior of trees. It will give weekly report of the plants.
- **Plant Manager:** The device will monitor if the plant is getting proper water and light. If not, it will provide light and water by itself without the interactions of the actual user. It means, just plant tree and set the device. Rest will be done by the device.
- **Using pH sensor:** using pH sensor, the device can recommend the users the proper name and amount of fertilizers needed.
- **Using soil element testing sensor:** Although the soil element testing sensor are still on development, or are used in industrial sectors only. Proper utilization of it can ensure survival of plants. [16]
- **Community Blog:** A community blog for the ones who care about trees. It will encourage people for tree plantation. Also, the discussions will make people know more about plants. In such a way, a good community will be developed where people discuss more about plants.

REFERENCES

- [1] V. K. Solanki, "Advanced Automated Module for Smart and Secure City," *Researchgate*, 2016.
- [2] R. Gill, "Hackster IO," Hackster, 30 March 2016. [Online]. Available: <https://www.hackster.io/ryanjgill2/plant-monitoring-system-88ed2b>.
- [3] "Teach Me Micro," Teach Me Micro, 2019. [Online]. Available: <https://www.teachmemicro.com/intro-nodemcu-arduino/>.
- [4] "Happyec.in," Happyec, 2016. [Online]. Available: <https://happyec.in/product/dht11-temperature-and-humidity-sensor-module/>.
- [5] "DF Robot," DF Robot, 2019. [Online]. Available: https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193.
- [6] "Robo Haat," Robo Haat, 2019. [Online]. Available: <https://robohaar.com/product/bh1750fvi-digital-light-intensity-sensor-module/>.
- [7] "Sparkfun," Sparkfun, 2019. [Online]. Available: <https://www.sparkfun.com/products/10168>.
- [8] K. K. Krishna, "Automatic plant monitoring system," *IEEE*, 2017.
- [9] "Github issues," Github, 3029. [Online]. Available: <https://github.com/esp8266/Arduino/issues/6110>.
- [10] S. Robillard, "Raspberry Stack Exchange," June 2017. [Online]. Available: <https://raspberrypi.stackexchange.com/questions/68133/is-soil-moisture-sensor-corrosion-normal>.
- [11] R. Gour, "Quora," April 2016. [Online]. Available: <https://www.quora.com/What-is-Django-Architecture>.
- [12] "Laravel," Laravel, 2019. [Online]. Available: <https://laravel.com/>.
- [13] S. N. Mukti, "Study of urban forestry through road side tree plantation in Dhaka city," *BUET Journal*, 2006.
- [14] F. Balducci, "Machine Learning Applications on Agricultural Datasets for Smart Farm Enhancement," *University degli studi di Bari*, 2018.
- [15] A. R. J. Janata, "Miniaturization in analytical chemistry," *Analytica Chimica Acta*, vol. 118, no. 1, pp. 45-52.
- [16] Robert and R. Jerjavec, "Personal Develop Fit," 21 March 2019. [Online]. Available: <https://personaldevelopfit.com/quotes-about-sacrifice/>.

Appendices

Appendix A: Research Reflection

A.1 Datamine Process using Python:

```
def datamine():
    # url = "http://djangpi.herokuapp.com/api/dataLogs/"
    url = request.args.get('url')

    datalogs = requests.get(url=url).json()
    # return render_template("index.html", data=datalogs)
    data_frame = pd.DataFrame(datalogs, columns=['id', 'light',
'temperature', 'humidity', 'soil', 'moisture', 'alive'])
    # return render_template("index.html", data=data_frame)
    data_frame_filter = data_frame.query('alive!=0')
    data_frame_filter['soil'] = np.where(data_frame_filter['soil'] ==
'Dry', 0, data_frame_filter['soil'])
    data_frame_filter['soil'] = np.where(data_frame_filter['soil'] ==
'Wet', 1, data_frame_filter['soil'])
    data_frame_filter['soil'] = np.where(data_frame_filter['soil'] == 'Very
Wet', 2, data_frame_filter['soil'])

    # return render_template("index.html", data=data_frame_filter)

    data = {
        'light': [
            data_frame_filter['light'].min(),
            data_frame_filter['light'].max()
        ],
        'temperature': [
            data_frame_filter['temperature'].min(),
            data_frame_filter['temperature'].max()
        ],
        'humidity': [
            data_frame_filter['humidity'].min(),
            data_frame_filter['humidity'].max()
        ],
        'soil': [
            soilTest(data_frame_filter['soil'].min())
            ,
            soilTest(data_frame_filter['soil'].max())
        ],
        'moisture': [
            data_frame_filter['moisture'].min(),
            data_frame_filter['moisture'].max()
        ]
    }
    return Response(json.dumps(data))
```

The above code is used to do the data mining process using Pandas & NumPy Library of Python. The Final Output is as follows-

A.2 JSON Output

```
{  
  "light": [0,25],  
  "temperature": [29,30],  
  "humidity": [73,79],  
  "soil": ["Dry", "Wet"],  
  "moisture": [619,818]  
}
```

This is the final json output of ranged values.

PLUGIARISM REPORT

Turnitin Originality Report Processed on: 04-Nov-2019 11:15 +06 ID: 1206463383 Word Count: 6774 Submitted: 1											
Data Mining on Home Forestation using IoT By Tanveer Hoque	<table> <tr> <td>Similarity Index</td><td>20%</td></tr> <tr> <td>Similarity by Source</td><td></td></tr> <tr> <td>Internet Sources:</td><td>12%</td></tr> <tr> <td>Publications:</td><td>4%</td></tr> <tr> <td>Student Papers:</td><td>11%</td></tr> </table>	Similarity Index	20%	Similarity by Source		Internet Sources:	12%	Publications:	4%	Student Papers:	11%
Similarity Index	20%										
Similarity by Source											
Internet Sources:	12%										
Publications:	4%										
Student Papers:	11%										

3% match (Internet from 28-Jun-2018) https://tsukuba.repo.nii.ac.jp/index.php?action=pages_view_main&active_action=repository_action_common_download&attribute_id=17&block_id=83&file_no=1&item_id=34633&ite
3% match (publications) K. Krishna Kishore, M. H. Sai Kumar, M. B. S. Murthy, "Automatic plant monitoring system", 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017
2% match (Internet from 12-Sep-2019) https://www.ijimai.org/journal/sites/default/files/files/2016/08/ijimai20174_3_5_pdf_17263.pdf
2% match (student papers from 02-Apr-2019) Class: Spring 2019 Assignment: Report Check_ Spring2019 Paper ID: 1104440978
1% match (Internet from 05-Jun-2019) https://www.dpytechnology.com/sensors-modules/Sensors/temperature-and-humidity-sensors/DHT11-Temperature-and-Humidity-Sensor-Module
1% match (Internet from 23-Oct-2019) https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193
1% match (student papers from 07-Apr-2019) Submitted to University of Hong Kong on 2019-04-07
1% match (Internet from 14-Aug-2019) https://duino4projects.com/projects/phone-projects/
1% match (Internet from 24-Jun-2019) https://www.hackster.io/UtkarshVerma/6-pin-push-switch-working-7bb5ef
1% match (student papers from 03-May-2019) Submitted to Ganpat University on 2019-05-03
< 1% match (student papers from 16-Oct-2014) Submitted to University of Wollongong on 2014-10-16
< 1% match (student papers from 13-Mar-2018) Submitted to Sandra Day O'Connor High School on 2018-03-13
< 1% match (student papers from 15-May-2019) Submitted to City of Glasgow College on 2019-05-15
< 1% match (Internet from 09-Sep-2019) https://laravel.com/
< 1% match (student papers from 19-Dec-2018) Submitted to Universiti Teknologi MARA on 2018-12-19
< 1% match (student papers from 05-Apr-2016) Submitted to VIT University on 2016-04-05
< 1% match (Internet from 10-May-2019) https://data-flair.training/blogs/django-architecture/
< 1% match (student papers from 21-Jun-2018) Submitted to Middle East College of Information Technology on 2018-06-21
< 1% match (student papers from 26-Sep-2019) Submitted to NorthTec on 2019-09-26
< 1% match (student papers from 05-Feb-2019) Submitted to Universiti Tenaga Nasional on 2019-02-05
< 1% match (student papers from 31-Oct-2019) Submitted to University of Strathclyde on 2019-10-31
< 1% match (publications) Aris Pujud Kurniawan, Agung Nugroho Jati, Fairuz Azmi, "Weather prediction based on fuzzy logic algorithm for supporting general farming automation system", 2017 5th International Conference on Instrumentation, Control, and Automation (ICA), 2017
< 1% match (student papers from 14-Dec-2015) Submitted to University of Leeds on 2015-12-14

< 1% match (Internet from 19-Oct-2018) http://amsdottorato.unibo.it/8535/1/Lalli_Eleonora_tesi.pdf
< 1% match (student papers from 02-Jun-2019) Submitted to Melbourne Institute of Technology on 2019-06-02
< 1% match (student papers from 26-Apr-2019) Submitted to South Bank University on 2019-04-26
< 1% match (student papers from 28-Jun-2018) Submitted to Universiti Teknologi MARA on 2018-06-28
< 1% match (Internet from 16-Oct-2019) https://www.electronicshub.org/dht11-humidity-sensor-arduino/
< 1% match (Internet from 18-Sep-2019) https://core.ac.uk/download/pdf/148161362.pdf
< 1% match (Internet from 18-Aug-2017) https://github.com/nitins11/Nokia5110LCD
< 1% match (student papers from 02-Apr-2018) Submitted to Daffodil International University on 2018-04-02
< 1% match (student papers from 09-Jan-2017) Submitted to University of Leeds on 2017-01-09
< 1% match (student papers from 30-Oct-2019) Submitted to Daffodil International University on 2019-10-30
< 1% match (student papers from 30-Jul-2014) Submitted to Universiti Malaysia Perlis on 2014-07-30
< 1% match (student papers from 18-Nov-2012) Submitted to University of Hull on 2012-11-18
< 1% match (student papers from 10-Dec-2018) Submitted to Universitas Brawijaya on 2018-12-10
< 1% match (student papers from 23-Oct-2019) Submitted to Leander Independent School District on 2019-10-23
< 1% match (student papers from 07-Dec-2018) Submitted to UIN Sunan Gunung Djati Bandung on 2018-12-07
< 1% match (student papers from 18-Feb-2019) Submitted to Universiti Tenaga Nasional on 2019-02-18