SMART FARMING USING PRECISION AGRICULTURE

Submitted in partial fulfilment of the requirements
of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER ENGINEERING

BY

Bodas Abhishek Sanjay (Roll No. 05)

Saroj Raja Rambarat (Roll No. 51)

Sharma Arun Rajesh (Roll No. 58)

Shah Santosh Gangaram (Roll No. 55)

Supervisor
Prof. Vinayak D. Shinde

DEPARTMENT OF COMPUTER ENGINEERING

SHREE L. R. TIWARI COLLEGE OF ENGINEERING

KANAKIA PARK, MIRA ROAD (E), THANE -401 107, MAHARASHTRA.

Year 2017

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Bodas Abhishek Sanjay

Roll No.:05

Saroj Raja Rambarat

Roll No.:51

Sharma Arun Rajesh

Roll No.:58

Shah Santosh Gangaram

RollNo.:55

Date: The _____ March, 2017

Shree Rahul Education Society's (Regd.)
SHREE L.R. TIWARI College of Engineering
(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)
ISO 9001:2008 Certified.

Kanakia Park, Mira Road(E), Thane-401107, Maharashtra.

CERTIFICATE

This is to certify that the project entitled **"SMART FARMING USING PRECISION AGRICULTURE"** is a bonafide work of

ABHISHEK BODAS (Roll No. 05)

RAJA SAROJ (Roll No. 51)

ARUN SHARMA (Roll No. 58)

SANTOSH SHAH (Roll No. 55)

Submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering"**.

Signature of Supervisor/Guide

Name: Prof. Vinayak D. Shinde

Date: _____

Signature of the H.O.D.

Name: Prof. Vinayak D. Shinde

Date: _____

Signature of the Principal

Name: Dr. S. Ram Reddy

Date: _____

Shree Rahul Education Society's (Regd.)
SHREE L.R. TIWARI College of Engineering
(Approved by AICTE, Government of Maharashtra and Affiliated to University of Mumbai)
ISO 9001:2008 Certified.

Kanakia Park, Mira Road(E), Thane-401107, Maharashtra.

Project Report Approval

This project report entitled "***SMART FARMING USING PRECISION AGRICULTURE"*** by ***Bodas Abhishek Sanjay, Saroj Raja, Sharma Arun and Shah Santosh*** is approved for the degree of Bachelor of Engineering in Computer Engineering.

Examiners

1. Name: _____

   Signature: _____

2. Name: _____

   Signature: _____

Date:

Place:

Acknowledgement

A few sublime human experiences defy expressions of any kind, and a feeling of true gratitude is one of them. We, therefore, find words quite inadequate to express our indebtedness to our Guide **Prof. Vinayak D. Shinde** for her virtuous guidance, encouragement and help throughout this work. Their deep insight into the problem and the ability to provide solutions has been immense value in improving the quality of project at all stages. This experience of working with them shall ever remain a source of inspiration and encouragement for us.

We express our thanks to **Prof. Vinayak D. Shinde**, HOD (CS), SLRTCE, Mira Road , for extending his support that he gave truly help the progression of the project work.

Our sincere thanks to **Dr. S. Ram Reddy**, Principal, SLRTCE, Mira Road for providing me the necessary administrative assistance in the completion of the work.

We are extremely grateful to the celebrated authors whose precious works have been consulted and referred in our project work. We also wish to convey our appreciation to our friends who provided encouragement and timely support in the hour of need.

Special thanks to our Parents whose love and affectionate blessings have been a constant source of inspiration in making this a reality.

All the thanks are, however, only fraction of what is due to Almighty for granting us an opportunity and the divine grace to successfully accomplish this assignment.

ABHISHEK BODAS

RAJA SAROJ

ARUN SHARMA

SANTOSH SHAH

Table of Contents

List of Tables

List of Figures

List of Abbreviations

PA	Precision Agriculture
IOT	Internet Of Things
PHP	Hypertext Pre-processor
GPS	Geo Positioning System
pH	Pouvoir Hydrogène

Abstract

In Project smart farming for precision agriculture is based on technology that we use in farming electronically and through software. These projects implement several technologies undertaken by Raspberry Pi INC, etc. The main purpose of these projects is to provide a farmer a new generation device that can be used in agriculture field over any electronics device using Transmission Control and Internet Protocol through any smart phone or Internet operating device. The technology implemented for this purpose is Internet of Things operating integrated circuits. This project will reduce the gap between customizable electronics devices and programming application or software since because of these project electronic devices can be programmed, interfaced and controlled with any other Object Oriented Languages like Java and Web based language like HTML, PHP, Node. JS, etc. The microprocessor that we use here is Raspberry Pi that will include all different types of sensors in cooperated. This robot will operate in farming place where it can sense data from agriculture like soil quality, Water Level in soil, fertility, Humidity and temperature that requires crop to grow at specific time period it will analyse data that is received from agricultural area and give farmer a statistical data that will help farmers to solve their problems.

Chapter 1

Introduction

1. Introduction

1.1 Introduction

This chapter will introduce the concept of precision agriculture. **Precision agriculture (PA)** is a farming management concept based on observing, measuring and responding to inter and intra-field variability in crops. This chapter will light up the topics like the description of the project and the former formulation of the problem behind it as well as what motivated the makers of the project to take a decision to make this project and its related problem solutions and thus covering up the scope of the project.

1.2 Problem statement

In India the farmers uses traditional methods for farming which is highly dependent on their past experience of crops cultivation but now a days there has been drastic change in ecological and geographical condition due to various causes like global warming, rise in sea level etc. the consequences of these causes produces uncertainty in weather conditions causes irregular rain fall by considering these cases we are developing these device which constantly monitor the soil and unprecedented weather behaviour which will notify the farmer in real time so that profitable decision can be taken by avoiding the risk factors.

1.3 Scope and Motivation

The concept of precision farming is not new for India. Farmers try their best to do the things for getting maximum possible yield with information and technologies available to them but unless & until total information about their fields and advanced technologies are available, they cannot do precision farming in perfect sense.

1.4 Project Objectives

- To make the device easy-to-use for farmers
- To make the device affordable, smart and portable for farmers
- To inspect farming parameters for moisture sensing in agriculture field
- To inspect farming parameters for Humidity sensing in agriculture field
- To inspect farming parameters for salinity sensing in agriculture field
- To inspect farming parameters for rain sensing in agriculture field
- To prepare database for storage of live parameters which is sensed by moisture, humidity, salinity, rain sensors.

Chapter 2

Literature Review

2. Literature Review

Here we will elaborate the aspects like the literature survey of the project and what all projects are existing and been actually used in the market which the makers of this project took the inspiration from and thus decided to go ahead with the project covering with the problem statement.

2.1 Existing System

The High precision positioning systems (like GPS) are the key technology to achieve accuracy when driving in the field, providing navigation and positioning capability anywhere on earth, anytime under any all conditions. The systems record the position of the field using geographic coordinates (latitude and longitude) and locate and navigate agricultural vehicles within a field with 2cm accuracy.

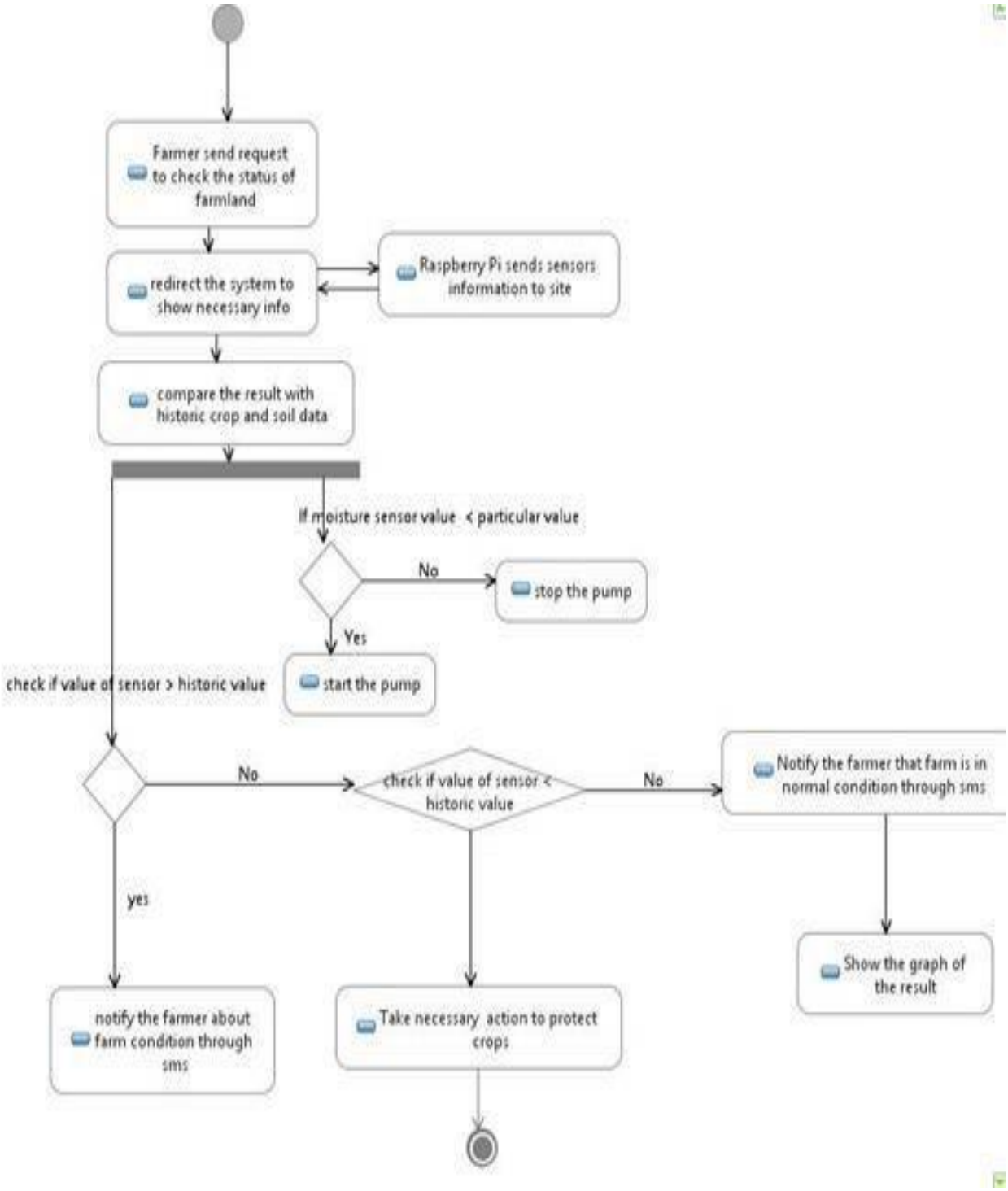
Automated steering systems: enable to take over specific driving tasks like auto-steering, overhead turning, following field edges and overlapping of rows. These technologies reduce human error and are the key to effective site management:

Assisted steering systems show drivers the way to follow in the field with the help of satellite navigation systems such as GPS. This allows more accurate driving but the farmer still needs to steer the wheel.

Automated steering systems, take full control of the steering wheel allowing the driver to take the hands off the wheel during trips down the row and the ability to keep an eye on the planter, sprayer or other equipment.

Intelligent guidance systems provide different steering patterns (guidance patterns) depending on the shape of the field and can be used in combination with above systems.

2.2 Proposed System



2.3 Advantages and Limitations

Advantages:

1. Our project will be a portable device which is useful, affordable and easy to use for farmers.
2. These project will give feasible solution to farmer that they can make use of during necessity

3. Obtain data in real time: the application of sensing devices in your fields will allow a continuous monitoring of the chosen parameters and will offer real time data ensuring an updated status of the field and plant parameters at all time.

4. Automate your field management: by incorporating a Decision Support System (DSS) in your Precision Agriculture environment the best conditions for the specific soil and plant species will be automatically optimised based on the data obtained by the sensors. The DSS will suggest the best moment for watering (or whether there is need or not), the need to irrigate to wash the salt content due to an excess in the radicular area, the need to fertilise, etc.

5. Save time and costs: by introducing a PA system in the daily operation of an agricultural exploitation time is saved due to the on-line measurement methods. Data from the sensors is automatically transmitted to a central server and this can be consulted using a Smartphone or Laptop. Or even, email or SMS alerts can be programmed to notify the field owner when there is a need to irrigate, fertilise or address any issue in their properties. Moreover, costs in terms of water, pesticides and others are optimised and can easily be reduced.

6. Improve your image: By using PA technology, not only the yield and profits will be increased but also the perception of the general public and Public Administration (through Smart Agriculture and environmental care) towards your activity will be enhanced.

Limitations:

1. Farmer has to carry device to various parts of field.
2. Internet connectivity problem in farms.
3. Under heavy conditions, it may not work efficiently.
4. Device cannot sustain any physical damage.
5. Data transferred and managed through the Internet of Things can be accidently exposed or leaked out. This might hamper with the personal privacy of the individual concerned and pose a threat to public security. Smart agriculture appliances usually devour a large amount of data and information about a user. So, harm and damage can be done to the concerned user if it is altered or mishandled.

Chapter 3

Requirement Analysis and Planning

3. Requirement Analysis and Planning

3.1 Functional Requirement

A functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.

VALIDATIONS TO BE PERFORMED:

Farmer will be provided id and password for login process.

- Source of input
  - Log In
- Input
  - Farmer id
  - Password
- Process
  - Farmer id and password are matched
- Output
  - Home page will open.

Hardware Requirements:

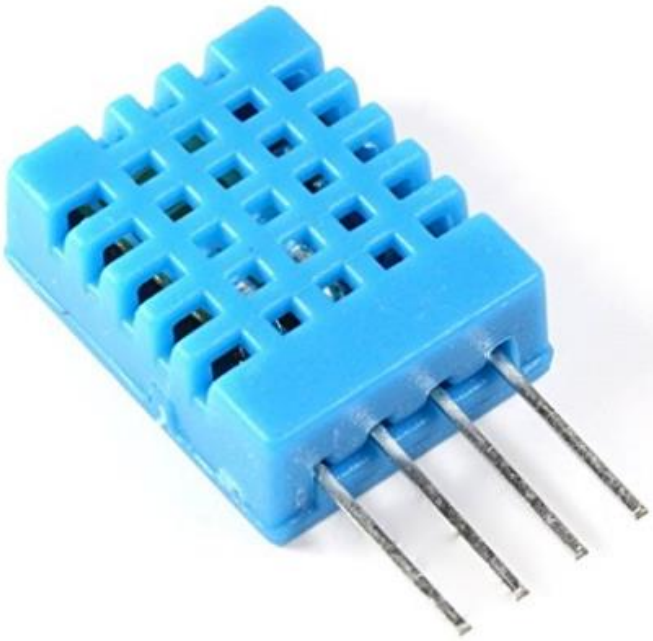
Board:

Raspberry pi 3 model B

Sensors:

RAIN DROP SENSOR Module KG004

SOIL MOISTURE SENSOR MODULE



DHT11 temperature and humidity sensor module

BMP180 High Precision Digital Barometric Pressure Sensor Board Module

jumper wires female to female

Xiaomi Mi 10400 mAh Power Bank Portable Travel External battery Pack Charger

Videocon 127cm (50) Full HD LED TV (VKV50FH16XAH, 4 x HDMI, 2 x USB)

Male to Male Full Hdmi Cable XSphere

QUANTUM QHM8810 MULTIMEDIA USB COMBO MOUSE+KEYBOARD

Sensor Description:

RAIN DROP SENSOR Module KG004

Rain Drop Sensor description:

The Raindrop Detection Sensor module is an easy-to-use and low cost drop recognition sensor. The sensor works through a series of exposed parallel traces on board which produces electrical variations when drops or water volume changes. By using microcontrollers or ADC ICs (Arduino and PIC) it's fairly easy to convert the analog output from the sensor to digital values. 1. Water Sensor water level sensor is an easy-to-use, cost-effective high level/drop recognition sensor, which is obtained by having a series of parallel wires exposed traces measured droplets/water volume in order to determine the water level 2. Easy to complete water to analog signal conversion and output analog values can be directly read Arduino development board to achieve the level alarm effect 3. Rainwater can be used to detect water level detection This can be directly read by an Arduino or a comparator circuit if you wish to use it as a rain detection alarm. It can be used to monitor a variety of weather conditions.

Soil Moisture Sensor Module

- Operating Voltage: 3.3V ~ 5V
- Sensing Probe Dimensions: 60x30mm
- Panel PCB Dimensions: 30 x 60mm
- On-board LM393 comparator
- On-board power indicator LED
- On-board digital switching indicator LED

MOISTURE SENSOR DESCRIPTION:
This Soil Moisture Sensor Module can be used to detect the moisture of soil or judge if there is water around the sensor, let the plants in your garden reach out for human help. Insert this module into the soil and then adjust the on-board potentiometer to adjust the sensitivity. The sensor would outputs logic HIGH or LOW when the moisture is higher or lower than the threshold set by the potentiometer. With help of this sensor, it will be realizable to make the plant remind you: Hey, I am thirsty now, please give me some water.

Product Description:

- DHT11 sensor adopts
- The module can detect surrounding environment of the humidity and temperature
- High reliability and excellent long-term stability
- Output form digital output
- Has fixed bolt hole and easy installation

- Humidity Measurement Range: 20%¡«90%RH (0-50 deg C Temperature compensation)
- Temperature Measurement Range: 0¡«+50 deg C
- Humidity Measurement Accuracy: ¡À5.0%RH
- Temperature Measurement Accuracy: ¡À2.0 deg C
- Response Time: <5 seconds
- Size (L x W x H): Approx.0.6 x 0.5 x 0.2 inch / 15 x 12 x 5mm
- Color: Blue

BMP180 High Precision Digital Barometric Pressure Sensor Board Module description :
The BMP180 is a high-precision pressure sensors, small size, low power consumption, and can be used on mobile devices

Its excellent performance and lowest absolute precision can reach 0.03hPa, and extremely low power consumption, only 3 μA

BMP180 hosts using powerful 8-pin ceramic Leadless chip (LCC) ultra thin package, you can via the I2C bus directly connected to the microprocessor

- Pressure range: 300~1100hPa (elevation 9000 m ~-500 m)
- Power supply voltage: 1.8V~3.6V (VDDA), 1.62V~3.6V (VDDD)
- LCC8 package: lead-free ceramic carrier package (LCC)
- Size: 3.6mmx3.8x0.93mm
- Low power consumption: 5 μA, in standard mode
- High accuracy: low-power mode, the resolution of 0.06hPa (0.5 m)
- High linearity mode, resolution of 0.03hPa (0.25 m)
- With temperature output
- I2C interface
- Temperature compensation
- Lead-free, in line with the RoHS standard
- MSL 1-response time: 7.5ms
- Standby current: 0.1 μA
- No external clock circuit

Software Requirements:

- Web Servers
- web applications
- Database
- Basic programing knowledge of java, sql, python, php.

3.2 Non-Functional Requirement

3.2.1 Performance requirements

3.2.1.1 Reliability:

The reliability of our system is predicted to be very high and critical for proper functioning of the overall system. Our reliability is a factor which will be focused more than any other factors affecting it. The user may suffer very severely if the reliability is low as the main purpose of the system is to provide the proper scheduling of the applications.

3.2.1.2 Application literate

The user should be knowledgeable and aware about the usage of the mobile applications.

3.2.1.3 Availability:

The system will be available to farmer having an Android Smart phone.

3.2.2 Safety Requirements

There might occur a situation where database may get crashed at any certain time due to virus or any other operating system failure. The crash of the database may result in loss of all the data which will be very costly for our application. Therefore, the backup of the database will be created. The system is utmost secure and thus provides an in-depth security.

3.2.3 Security requirements:

The new profile formed is validated against the given email address. The system or the system requirements should not be manipulated by the users. In case of any such manipulation by the registered users, strict action should be taken for the security of the system.

3.3 System requirements

3.3.1 Development Environment Specification

Software requirements

1. Web Servers
2. Web applications
3. Database.
4. Basic programing knowledge of java, sql, python, php.

Hardware requirements

1. Android Smartphones with Jellybean 4.0 OS and higher.
2. Raspberry PI 3 with rain, temperature, moisture, weather and pH sensors.

3.4 Feasibility Study

3.4.1 Operational Feasibility

1. System will provide right and accurate information to the farmer.

3.4.2 Technical Feasibility

1. The device consists of basic technicalities which will be basic for farmers.

3.4.3 Economical Feasibility

The device will be cost-effective and will have less effect on their pockets.

1. The project is based on android platform which is been used by majority of the mobile users and which is also available at cheap rates, hence the application is economically light on the user's pocket.
2. The device will be cost-effective and will have less effect on their pockets.

3.4.4 Legal Feasibility

1. It does not abide any agricultural and environmental laws hence it is legally feasible by all means.
2. Terms and condition will be drafted and implemented on the application before making the application LIVE.

3.5 Project Planning

3.5.1 Project Timeline:

ACTIVITY	IMPLEMENTATION TIME				RESPONSIBILITY
1. Research on Precision Agriculture	August	September	October	November	
	15/8/16				Team Members
2 Surveying farmers condition in remote region of Maharashtra	18/8/16				Team Members
3. Analysis of current technologies for agricultural solution	23/8/16				Team Members
4 Requirement gathering	29/8/16				Team Members
5 Feasibility Analysis		10/9/16			Team Members
6 Designing the framework of precision agriculture		18/9/16			Team Members
7 Implementing IOT Technology for sensor data retrieval			3/10/16		Team Members
8 To prepare a database for storing historic data prescribed by the agricultural analyst and experts for mapping of every cultivation				10/10/16	Team Members

GROUP MEMBERS NAME	POSITION OF EVERY MEMBER	TASK PERFORMED BY MEMBER
Raja Saroj	Team Leader	Concept, Designing, Integration
Arun Sharma	Team Member	Concept, Designing, Integration
Abhishek Bodas	Team Member	Analysis, Documentation and Maintenance
Santosh Shah	Team Member	Coding, Testing, Documentation

Table 2

Chapter 4

Analysis Modelling

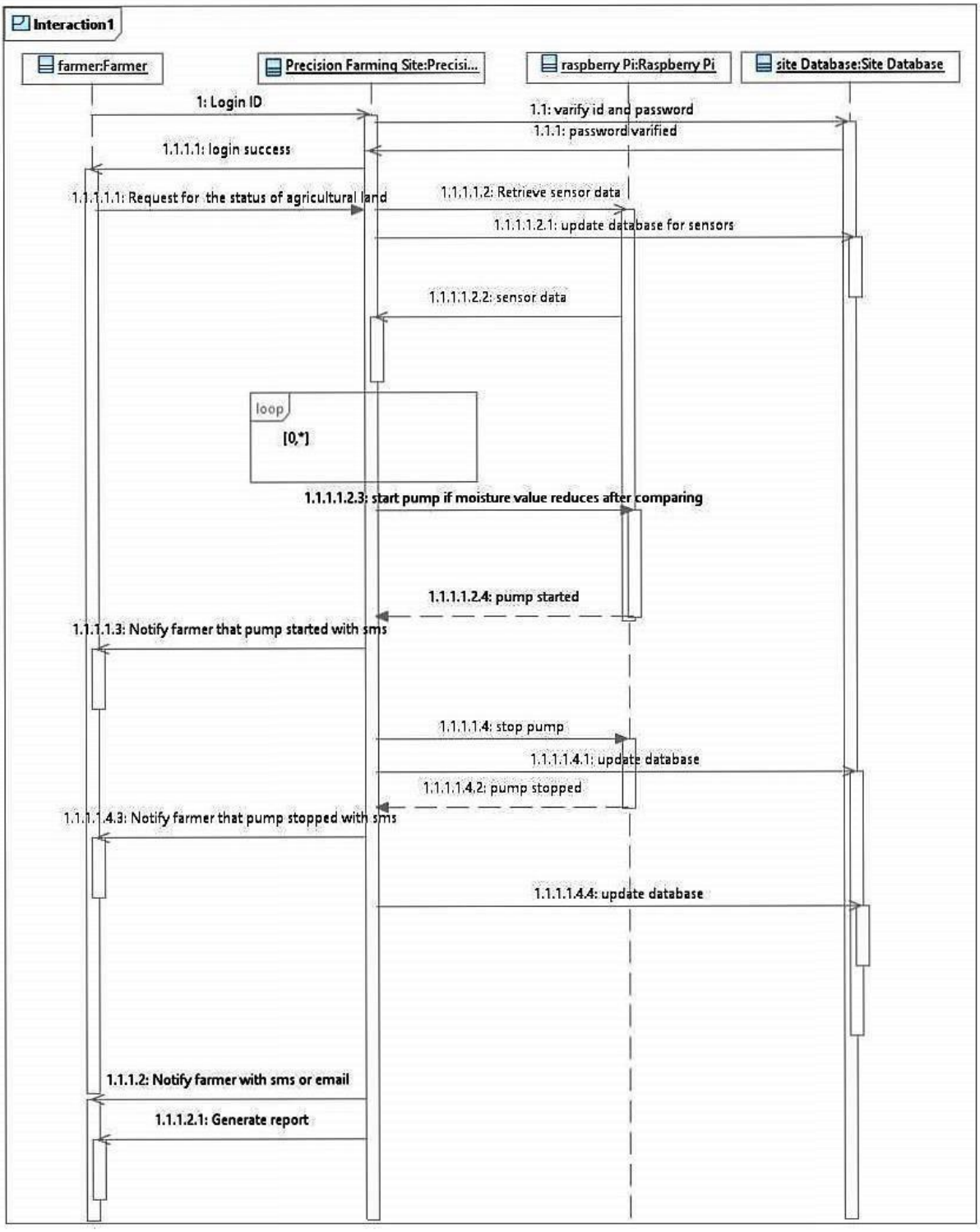
4.Analysis Modeling

In this chapter, all the aspects of the proposed system will be covered in the diagrammatic manner and provides the detailed manner of the system.

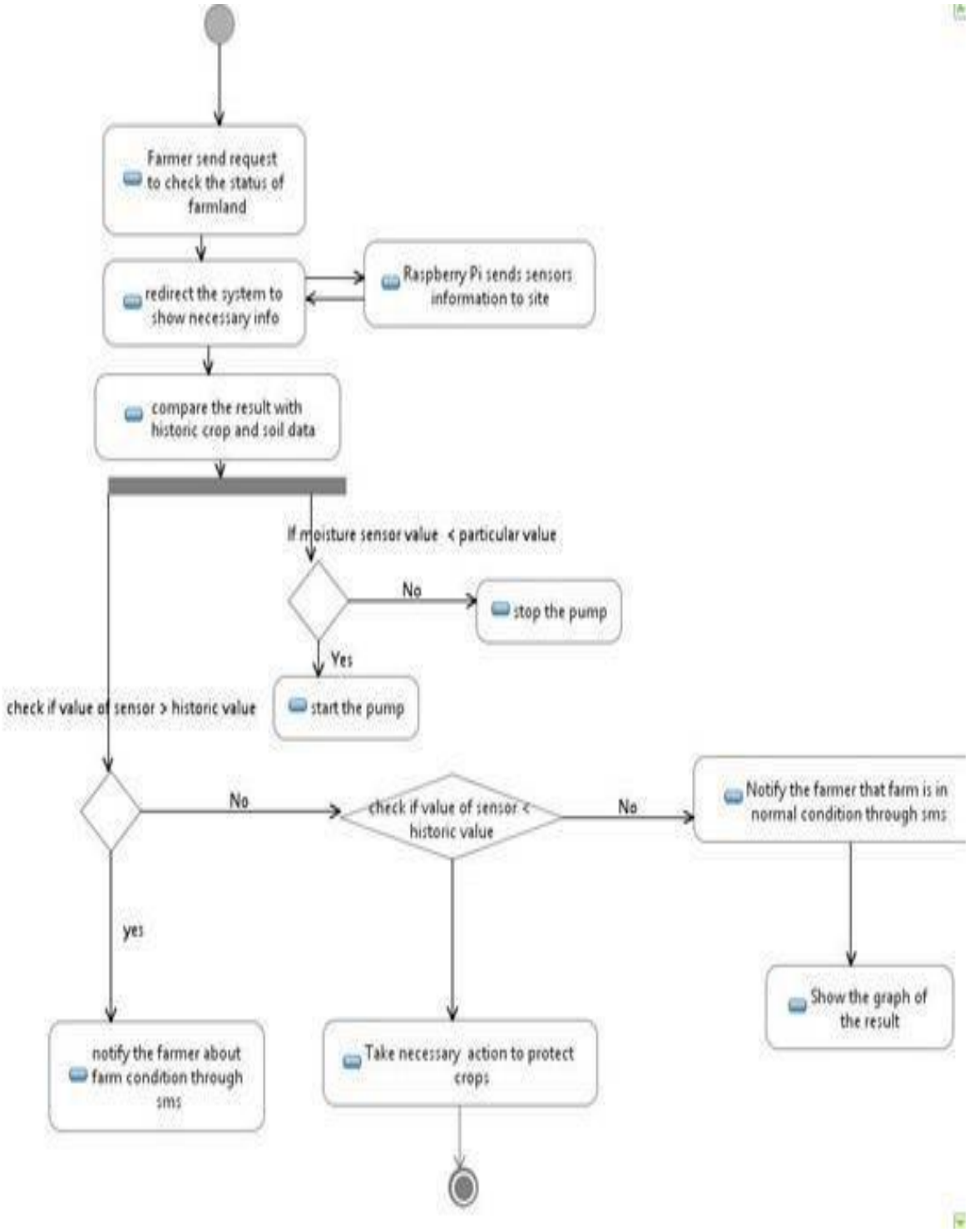
4.1 Behavioral Modeling

(explain Use case, Interaction diagrams, Activity & State-chart Diagram)

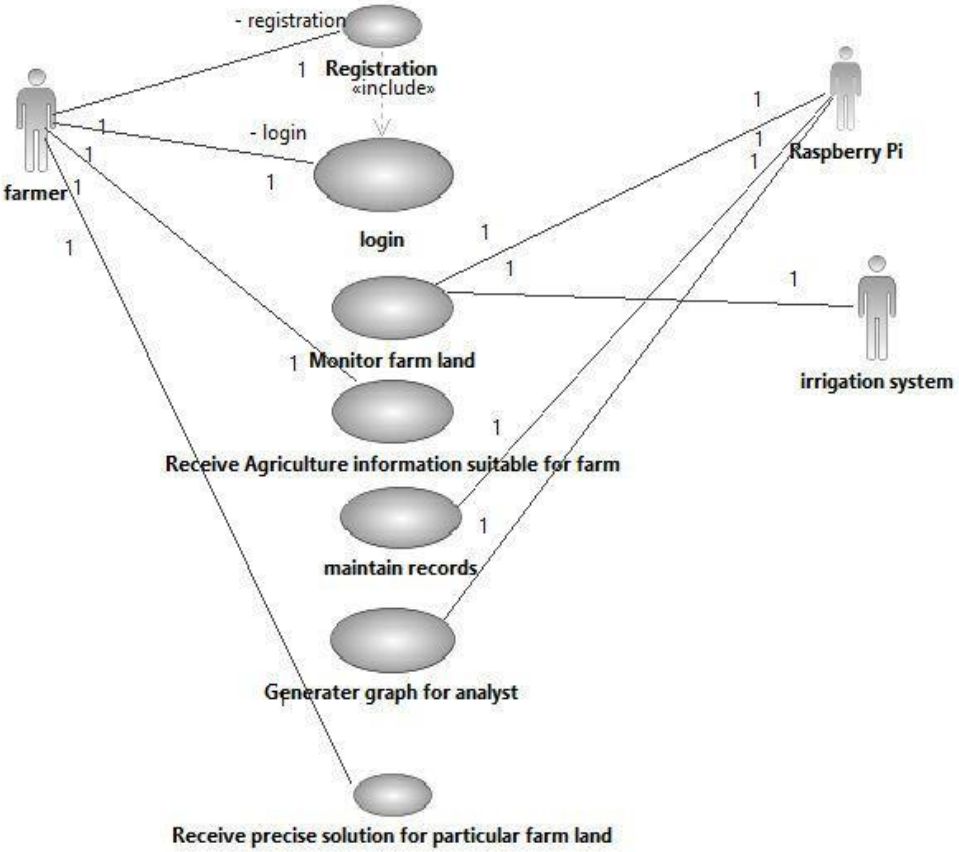
4.1.1 Sequence Diagram



4.1.2 Activity Diagram

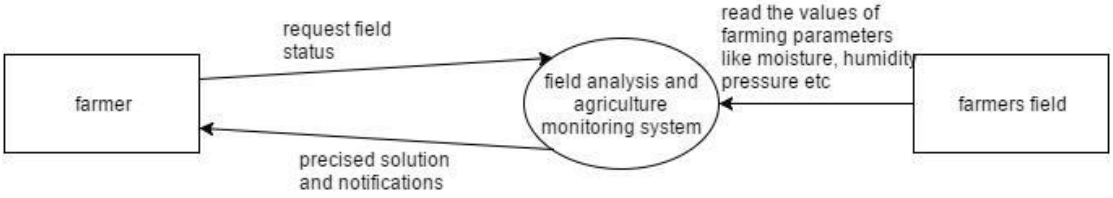


4.1.3 Interaction Diagram

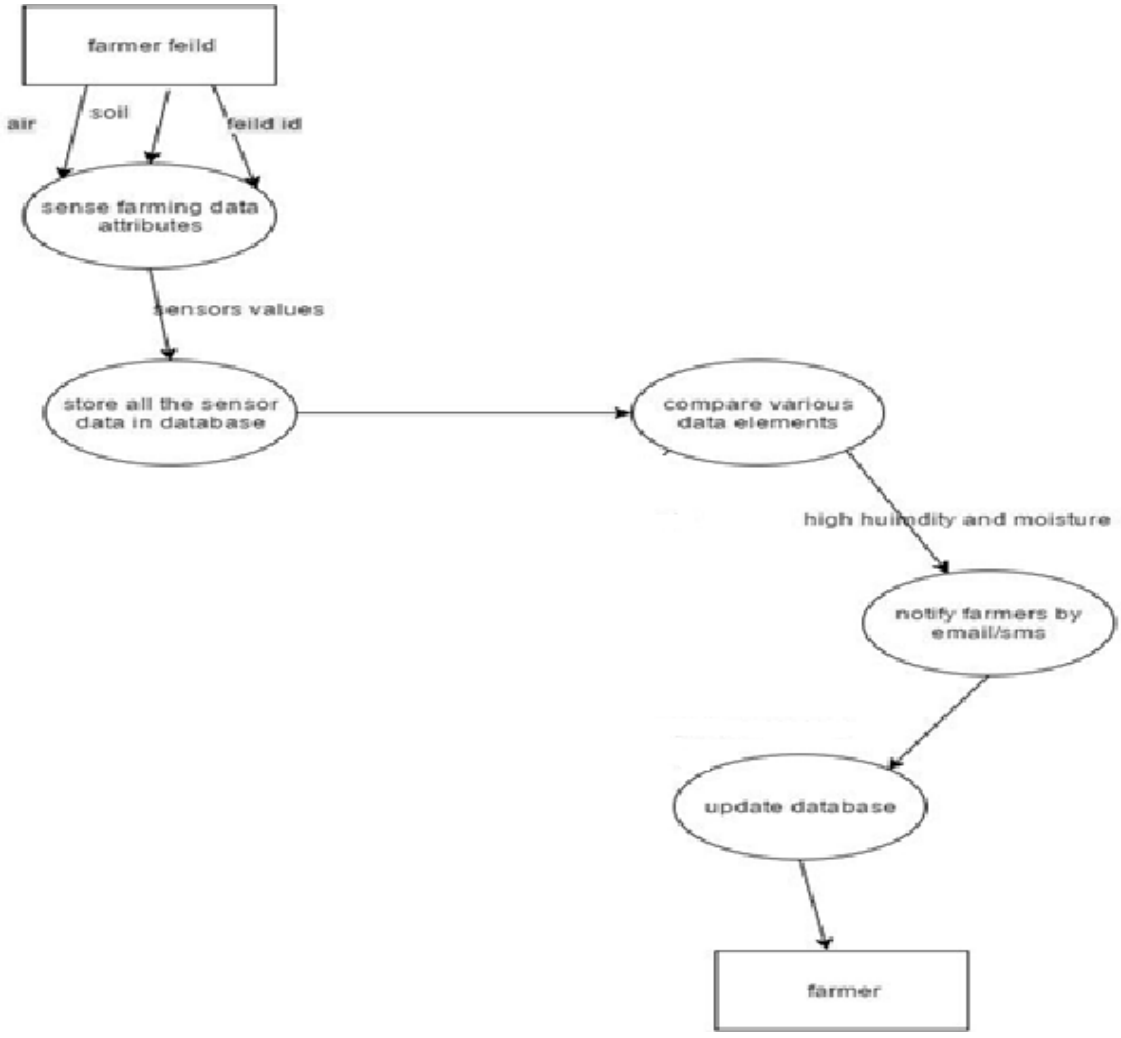


USE CASE

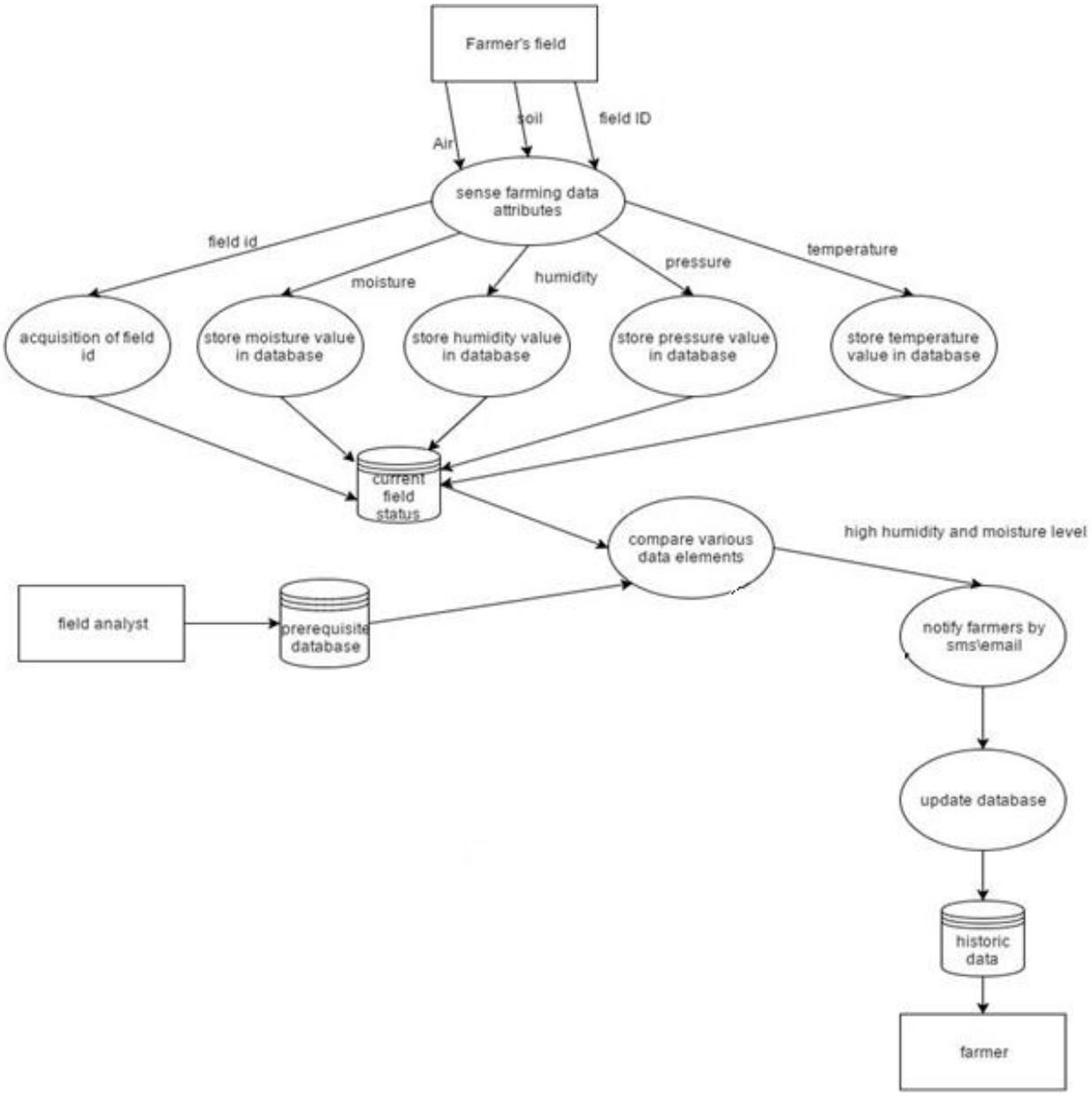
4.2 Data Flow Diagram (3 levels)



DFD LEVEL 0



DFD LEVEL 1


Farmer's field
Air
soil
field ID
sense farming data attributes
field id
moisture
humidity
pressure
temperature
acquisition of field id
store moisture value in database
store humidity value in database
store pressure value in database
store temperature value in database
current field status
compare various data elements
high humidity and moisture level
field analyst
prerequisite database
notify farmers by sms\email
update database
historic data
farmer

DFD LEVEL 2

4.3 ER Diagram

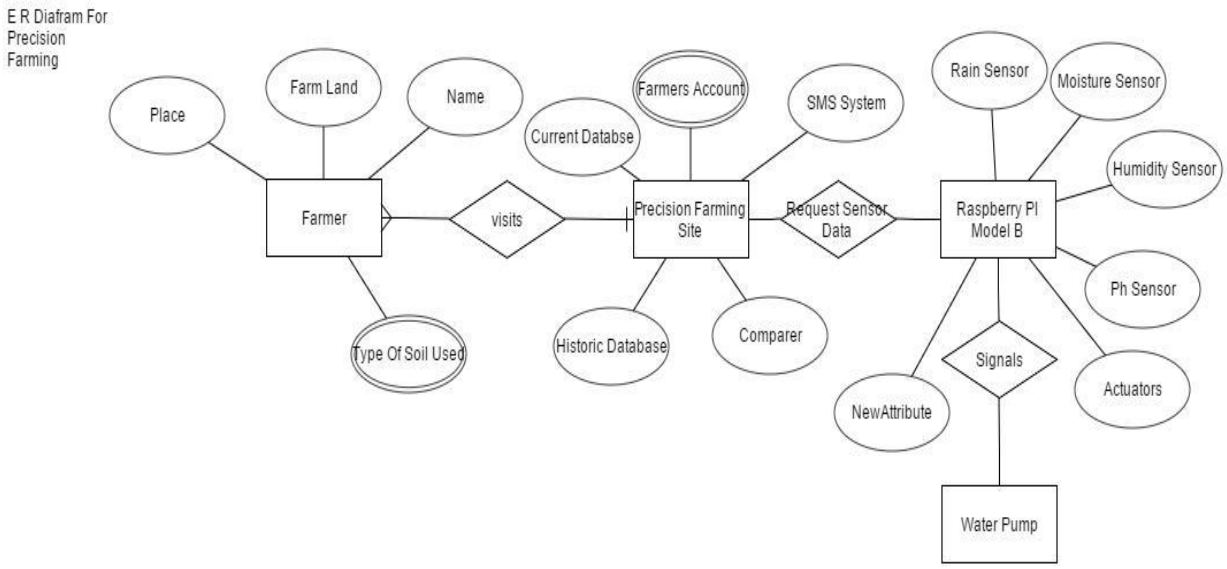


Table Name:- Humidity_Of_Farm_land

Temperature	Humidity
25	45%
27	40%

Table Name:- Moisture_Of_Soil

Moisture Level	Dryness
10mm	31%
15mm	21%

NORMALIZATION

Temperature	**Salinity**	**Moisture**	**Humidity**
27	4	10mm	45%
25	6	15mm	40%

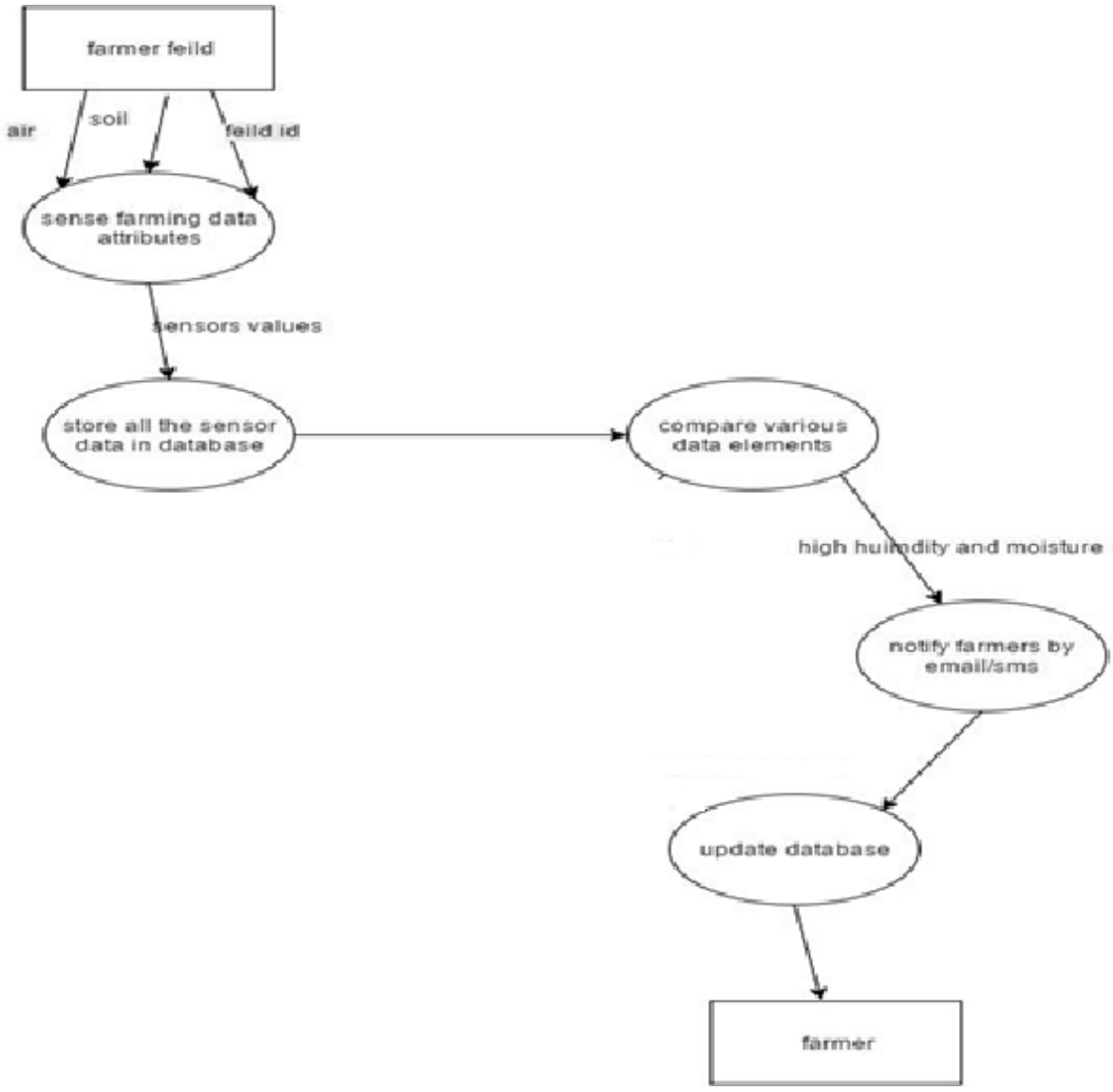
Chapter 5

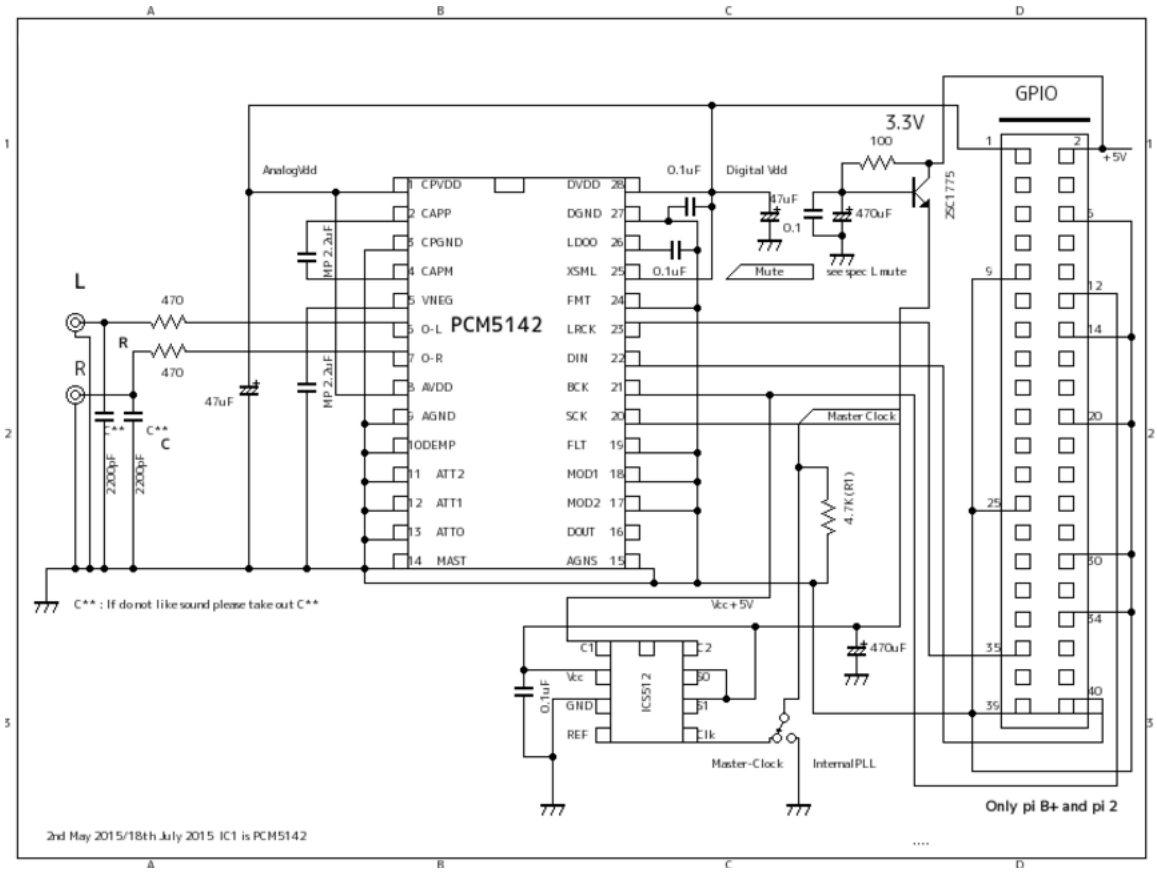
System Design

5.System Design

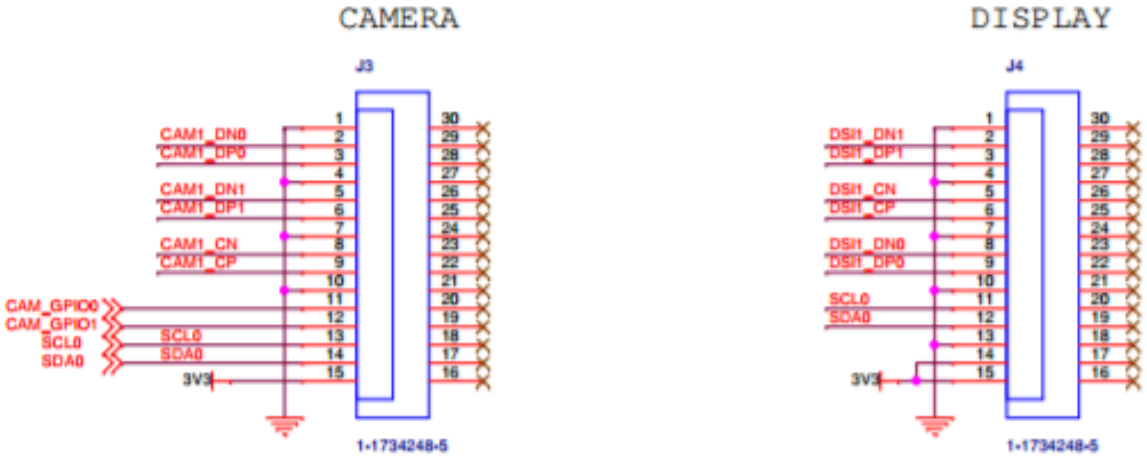
5.1 Application Architecture

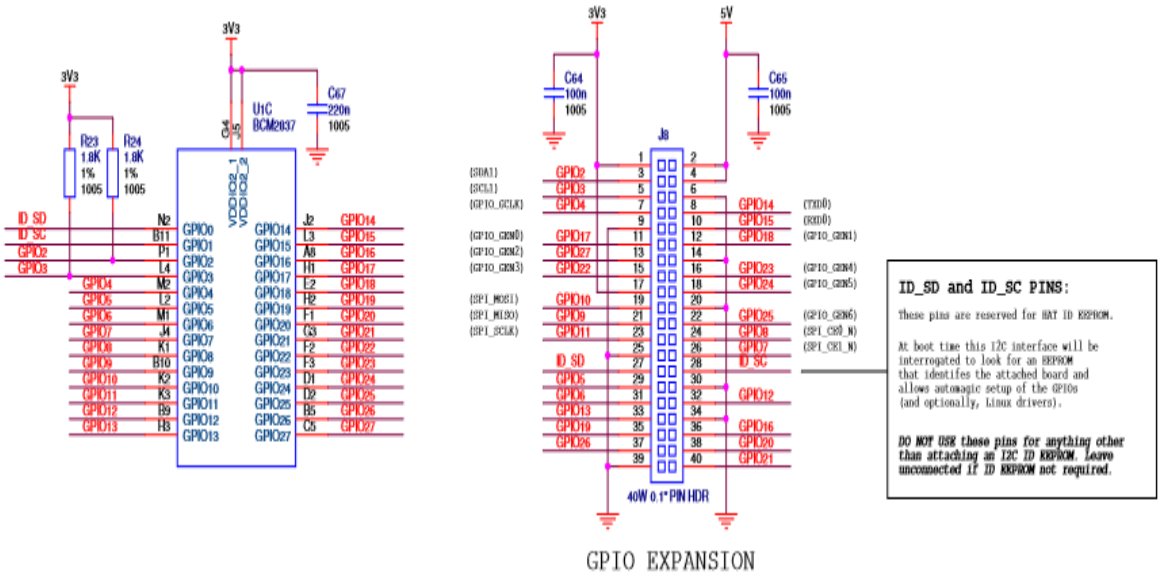
This system consist of a Linux computer which is raspberry pi version 3 model B it is interfaced with several set of sensors like humidity sensor, rain sensor, barometric sensor, moisture sensor etc. these sensors sense the values from the environment and save it in the database and will display this data to the farmer in the graphical format through a website.



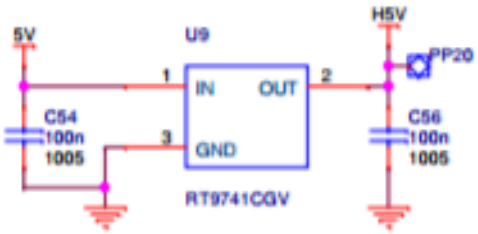


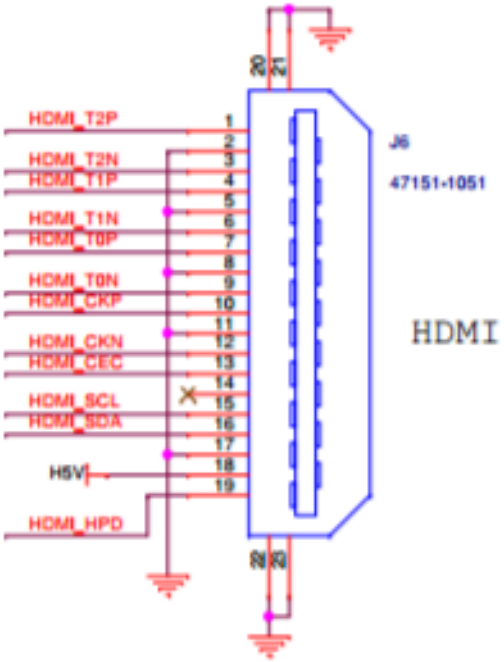
Circuit diagram Raspberry pi 3 model B

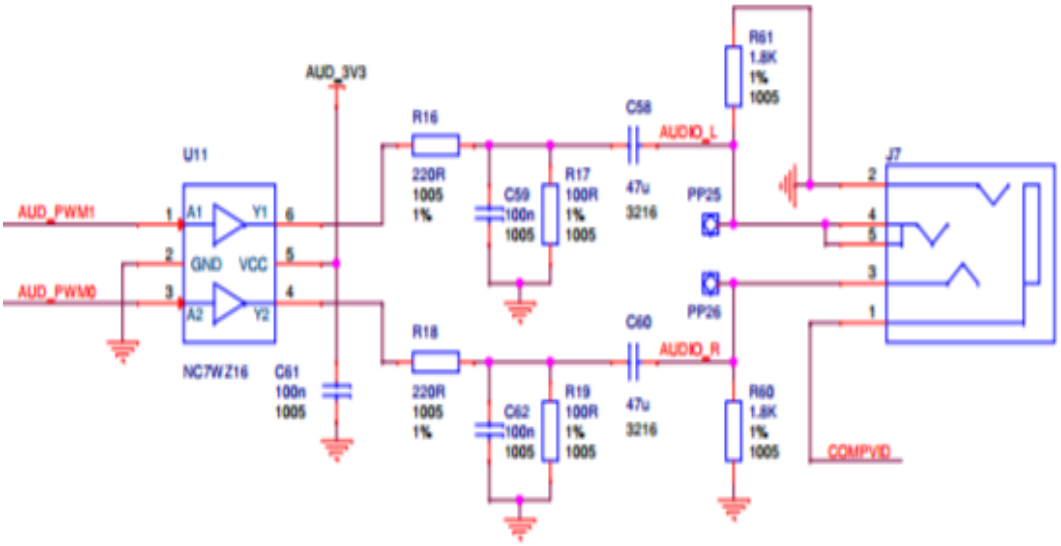




GPIO EXPANSION

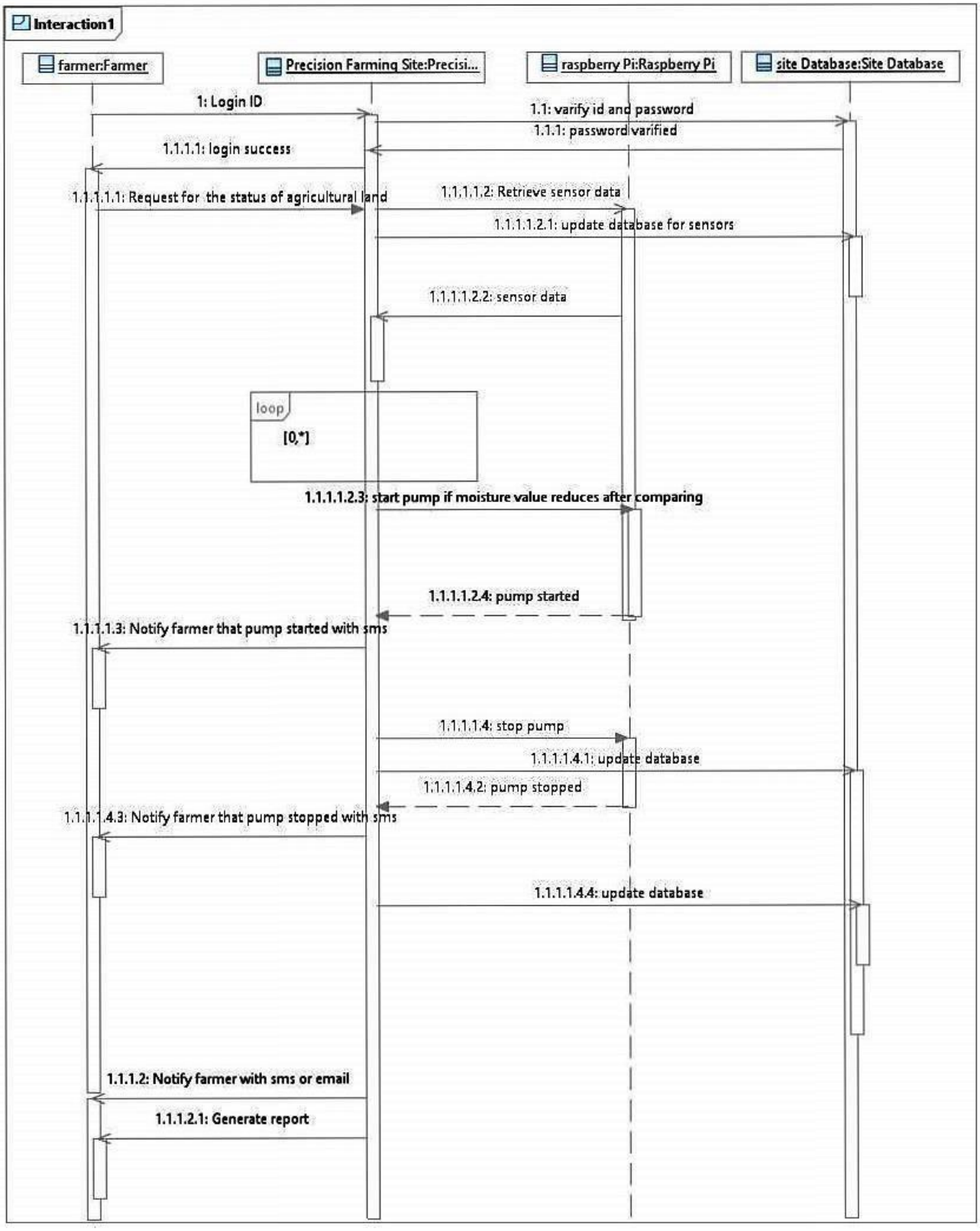




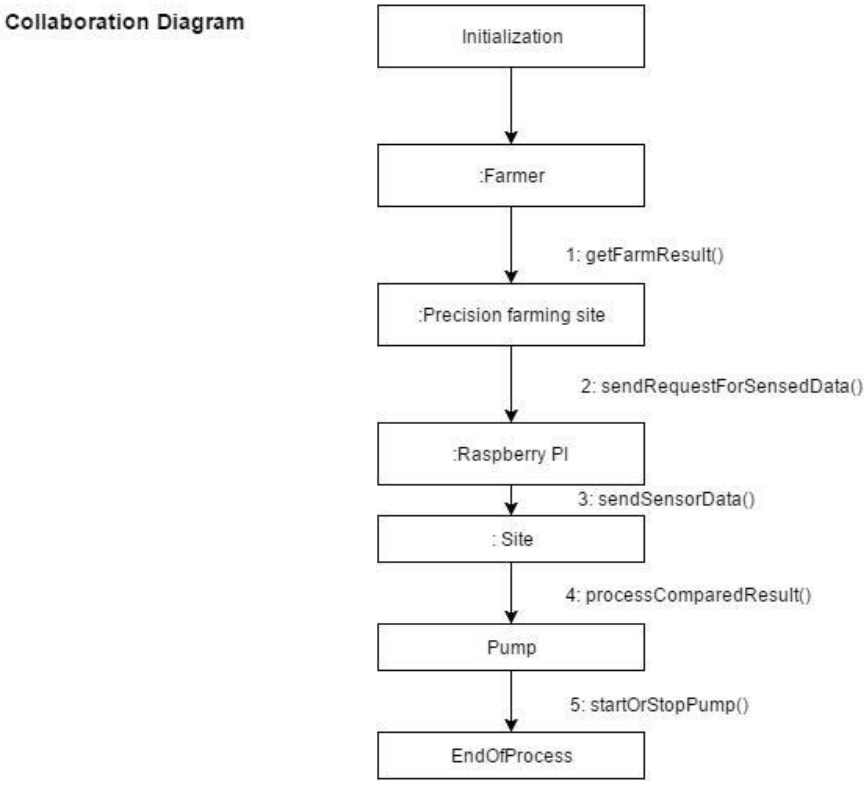

AUD_3V3
U11
AUD_PWM1
AUD_PWM0
A1
A2
GND
VCC
Y1
Y2
NC7WZ16
C61
100n
1005
R16
220R
1005
1%
R18
220R
1005
1%
C59
100n
1005
C62
100n
1005
R17
100R
1%
1005
R19
100R
1%
1005
C58
47u
3216
C60
47u
3216
AUDIO_L
AUDIO_R
PP25
PP26
R61
1.8K
1%
1005
R60
1.8K
1%
1005
J7
COMPVID

A/V

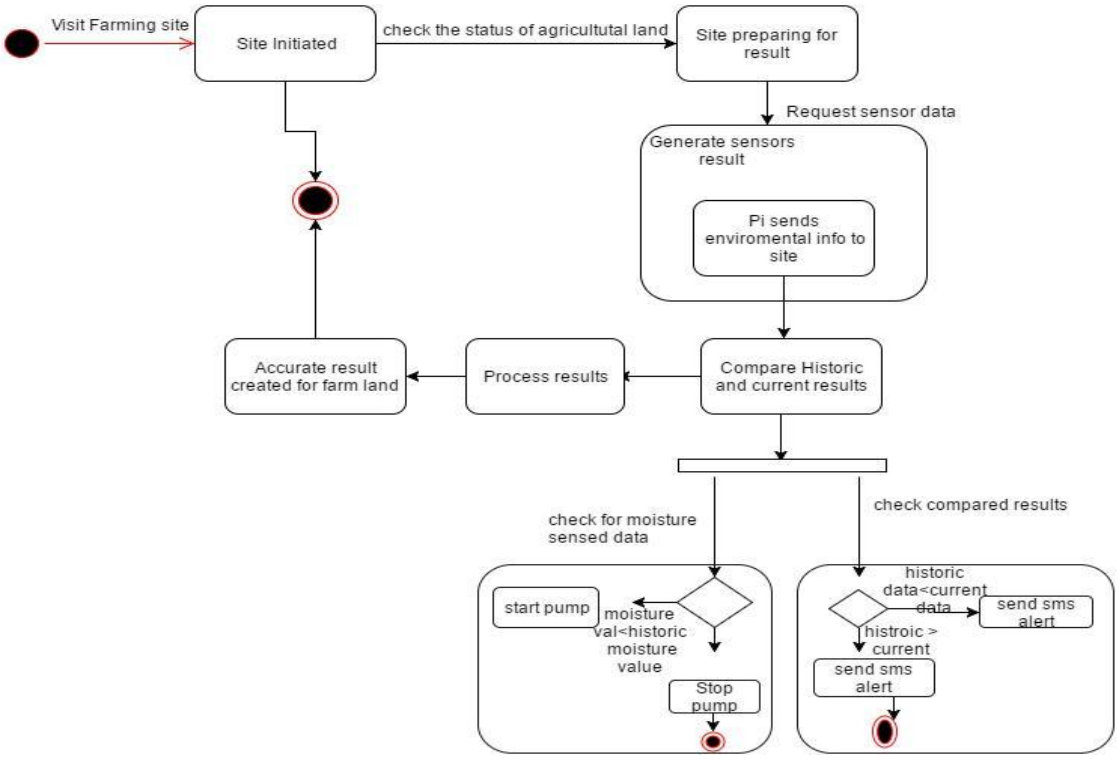
5.2 Structural Diagrams



SEQUENCE DIAGRAM

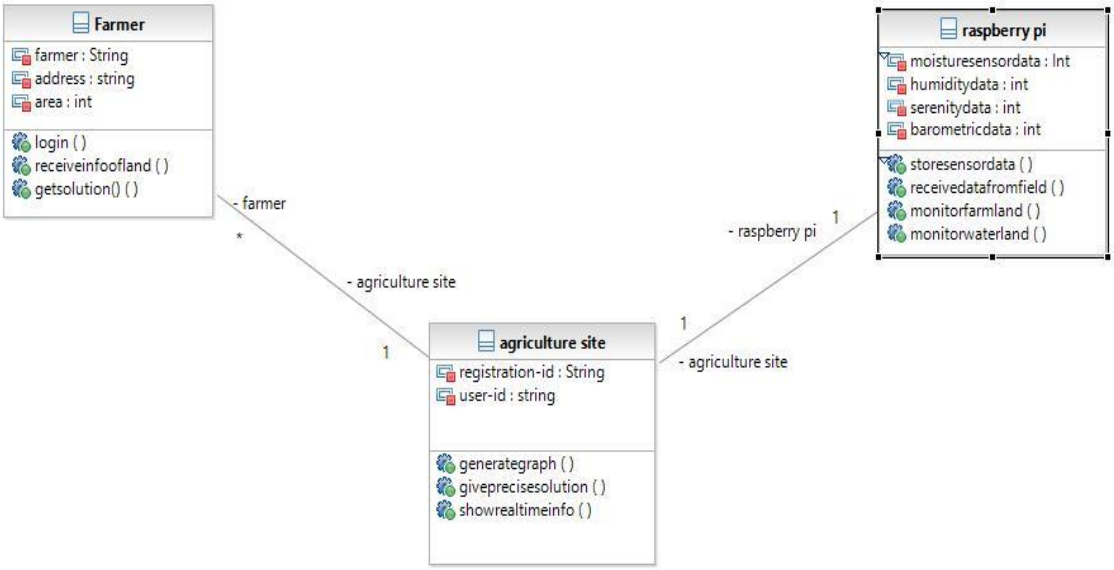


COLLABORATION DIAGRAM

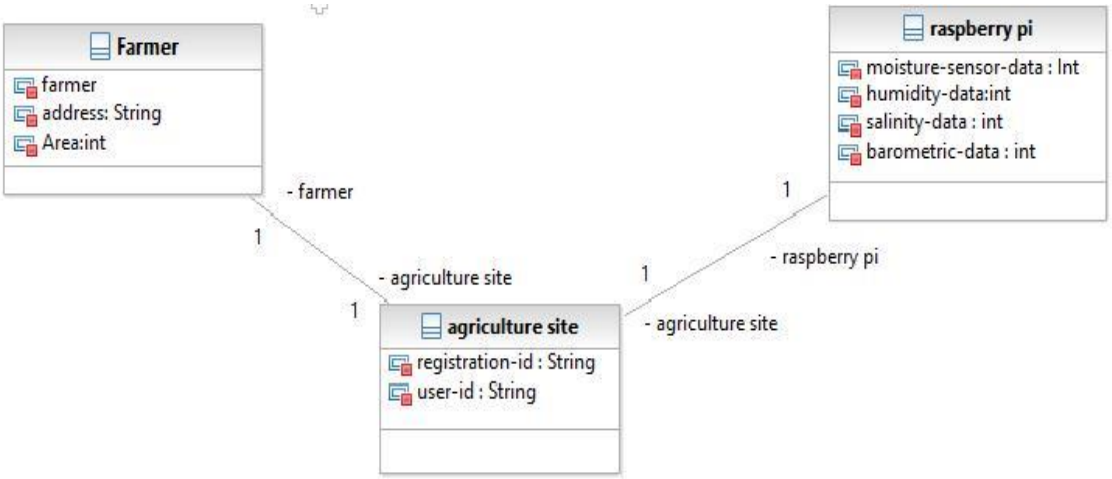


STATE DIAGRAM

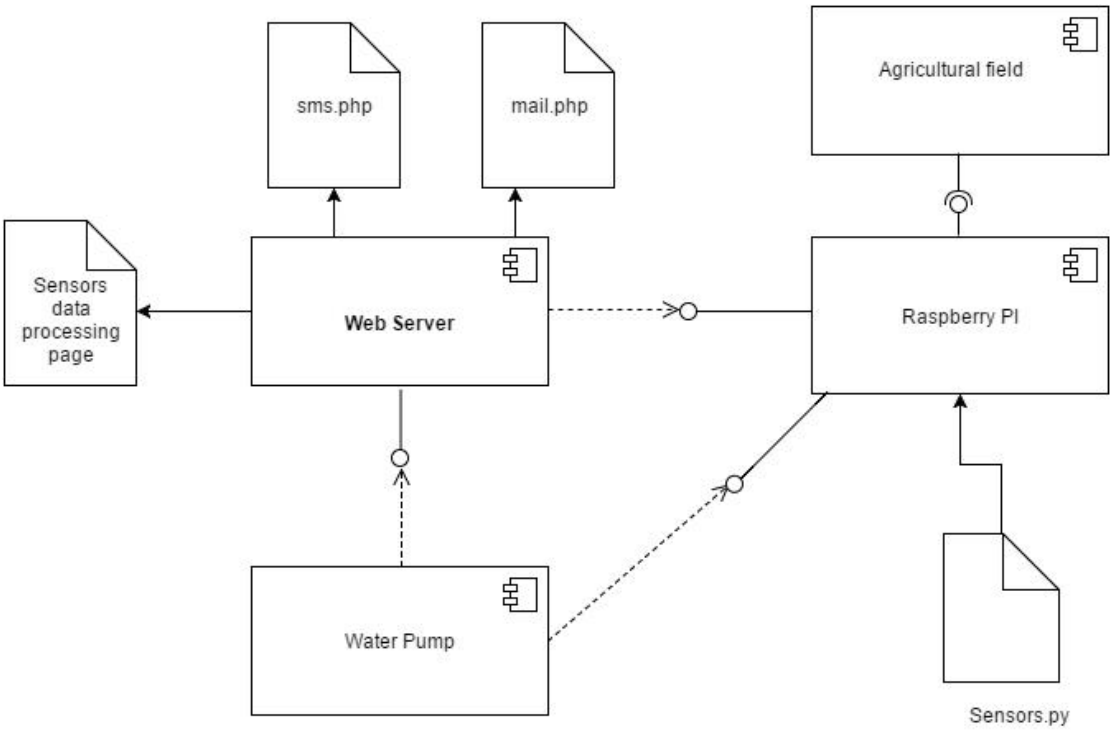
5.2.1 Class Diagram



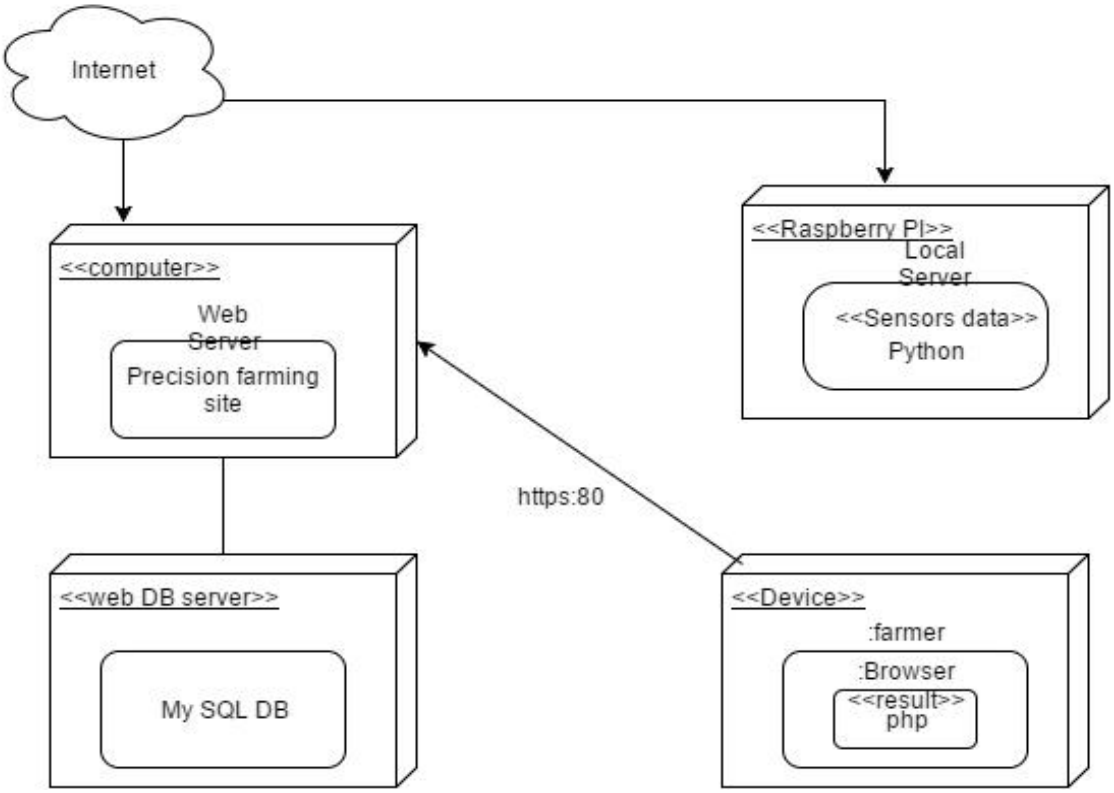
5.2.2 Object Diagram



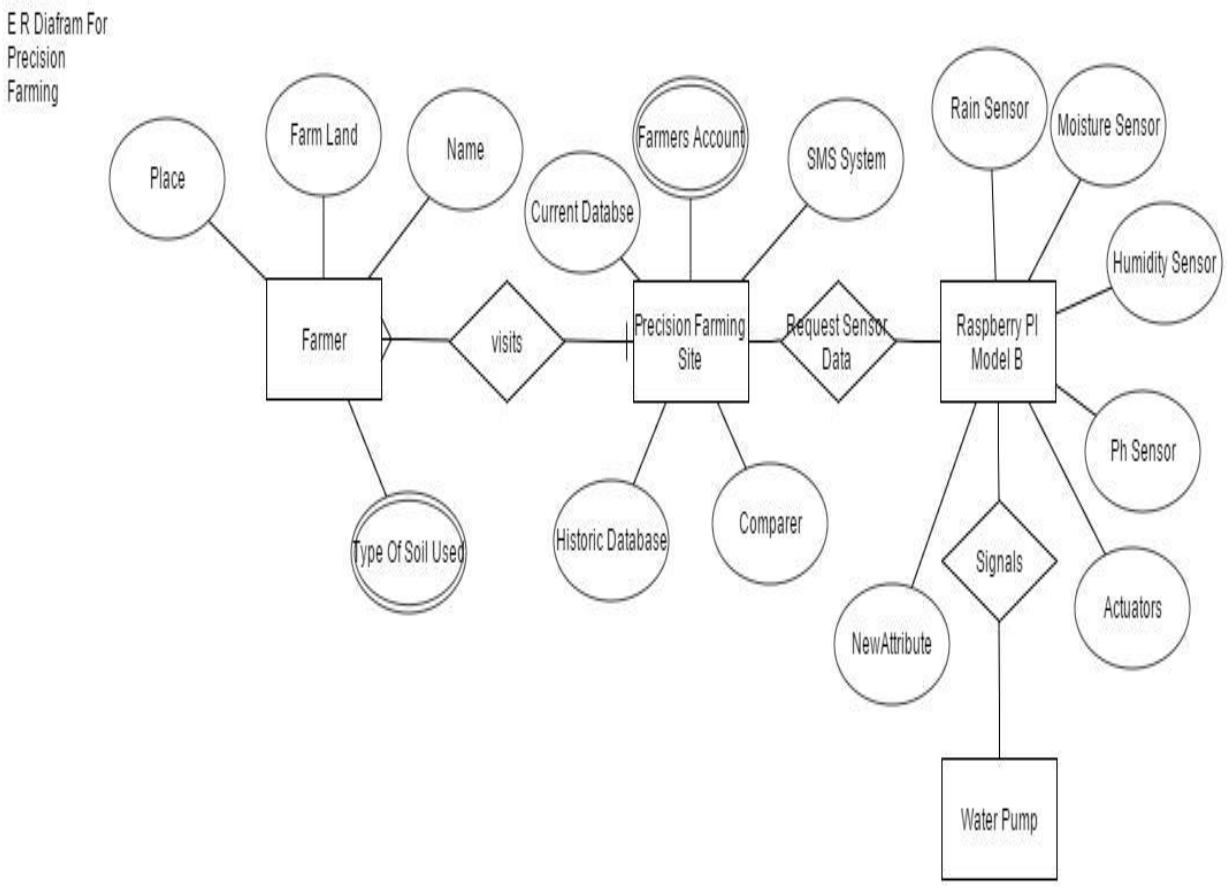
5.2.3 Component / Package Diagram



5.2.4 Deployment diagrams



5.3 Database Design



5.4 Graphical User Interface

PRECISION FARMING

Search... Search

WELCOME YOU ALL IN SMART PRECISION APPROACH IN FAR

Home | About us | Live Data | Check Field Status | Contact Us | Report | USER LOGOUT

PRECISION FARMING INTRODUCTION

Smart farming for precision agriculture is based on IOT and Monitoring agriculture which is accessible through the main frame Internet Infrastructure using web application. The server response system will be integrated with the strong database support in order to process the response or feedback build and bring to market new innovative IoT applications at 10 times the speed of other approaches with our rapid application development environment and drag and drop mashup builder. It has Leverage big data and analytics to provide new insights and recommendations to aid in better decision-making. It Enable farmers to easily visualize data and take action on insights and recommendations order to process the response or feedback build and bring .

Overview Of Precision farming Agriculture

. The main purpose of these site is to provide a farmer a new generation device that can be use in agriculture field over any electronics device using Transmission Control and Internet Protocol through any smart phone or Internet operating device. The technology implemented for this purpose is Internet Of Things. The microprocessor that we use here is Raspberry Pi that will include all different types of sensors in cooperated. This device will operate in farming place where it can sense data from agriculture like soil quality, Water Level in soil, fertility, Humidity and temperature that requires crop to grow at specific time period it will analyze data that is received from agricultural area and give farmer a statistical data that will help farmers to solve their crop cultivation.

Smart and precision farming motto

Despite being a newer concept in the field, there has been a tremendous popularity in the agricultural circuits about the benefits of smart farming and the applicability of IoT. It has been looked upon as a hope to encourage innovation in agriculture with connected farms speculated to be the future of farming considering the Indian perspective . IoT could be utilized to make the best out of our agricultural potential.New innovative IoT applications are addressing these issues and increasing the quality, quantity, sustainability and cost effectiveness of agricultural production. Todays large and local farms can, for example, leverage IoT to remotely monitor sensors that can detect soil moisture, crop growth and livestock feed levels, remotely manage and control their smart connected harvesters and irrigation equipment, and utilize prediction logics and analytics to quickly analyze operational data.

IOT based farming

Smart farming for precision agriculture is based on IOT and Monitoring agriculture which is accessible through the main frame Internet Infrastructure using web application.The server response system will be integrated with the strong database support in order to process the response or feedback obtained from device. This project will reduce the gap between customizable electronics devices and programming application or softwares since because of these project electronic devices can be programmed, interfaced and Monitored with any other Object Oriented Languages like Java and Web based language like HTML, PHP etc. New innovative IoT applications are addressing these issues and increasing the quality, quantity, sustainability and cost effectiveness of agricultural production. Todays large and local farms can, for example, leverage IoT to remotely monitor sensors that can detect soil moisture, crop growth and livestock feed levels, remotely manage and control their smart connected harvesters and irrigation equipment, and utilize prediction logics and analytics to quickly analyze operational data combined with 3rd party information, such as weather services, to provide new insights and improve decision making.

Chapter 6

Implementation

6.Implementation

6.1 Software Implementation

Installation of OS on raspberry pi:

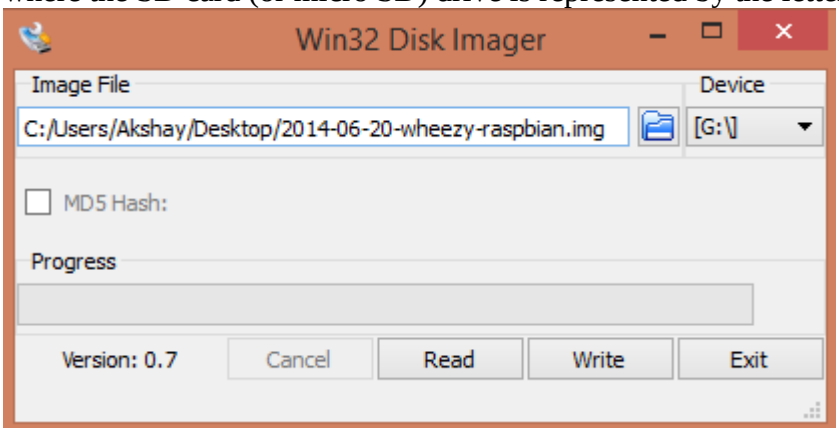
Raspberry Pi is a credit card sized micro processor available in different models with different processing speed starting from 700 MHz. Whether you have a model B or model B+, or the very old version, the installation process remains the same. People who have checked out the official Raspberry Pi website, might have seen them recommending the "NOOBS" or "NOOBS LITE" Operating System (aka "OS") for beginners. But using the Pi is very easy and from being a beginner, one will turn pro in no time. So, it's better to go with the more powerful and more efficient OS, the Raspbian. The main reason why Raspbian is extremely popular is that it has thousands of pre built libraries to perform many tasks and optimize the OS. This forms a huge advantage while building applications.

1. Downloading Raspbian and Image writer

Download the latest version of Raspbian from here. You can download it directly or via the torrents.
You will be needing an image writer to write the downloaded OS into the SD card (micro SD card in case of Raspberry Pi B+ model). So download the "win32 disk imager".

2. Writing the image

Insert the SD card into the laptop/pc and run the image writer. Once open, browse and select the downloaded Raspbian image file. Select the correct device, that is the drive representing the SD card. If the drive (or device) selected is different from the SD card then the other selected drive will become corrupted. SO be careful.
After that, click on the "Write" button in the bottom. As an example, see the image below, where the SD card (or micro SD) drive is represented by the letter "G:\"



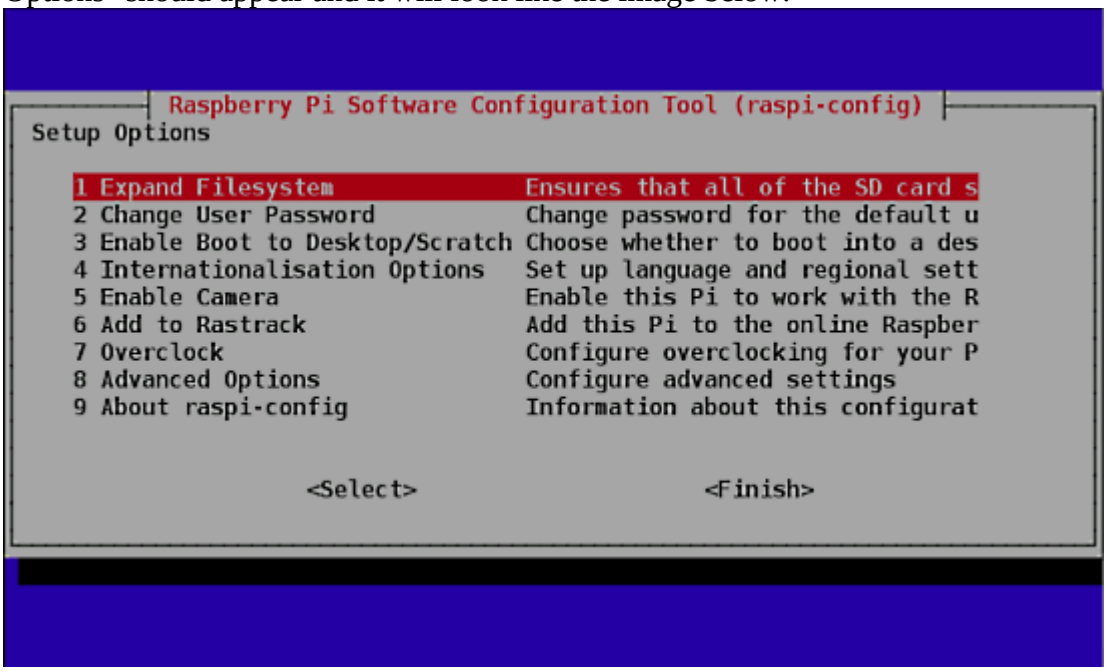
Once the write is complete, eject the SD card and insert it into the Raspberry Pi and turn it on. It should start booting up.

3. Setting up the Pi

Please remember that after booting the Pi, there might be situations when the user credentials like the "username" and password will be asked. Raspberry Pi comes with a default user name and password and so always use it whenever it is being asked. The credentials are:

```
login: pi

password: raspberry
```

When the Pi has been booted for the first time, a configuration screen called the "Setup Options" should appear and it will look like the image below.



If you have missed the "Setup Options" screen, its not a problem, you can always get it by typing the following command in the terminal.

```
sudo raspi-config
```

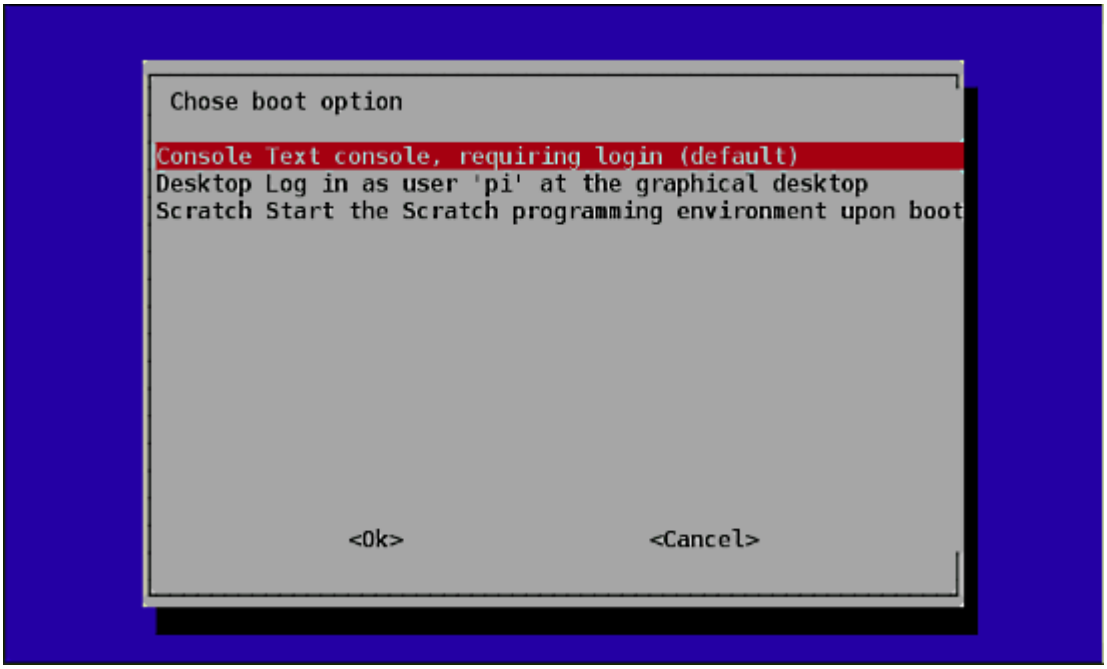
Once you execute this command the "Setup Options" screen will come up as shown in the image above.
Now that the Setup Options window is up, we will have to set a few things. After completing each of the steps below, if it asks to reboot the Pi, please do so. After the reboot, if you don't get the "Setup Options" screen, then follow the command given above to get the screen/window.

The first thing to do:

select the first option in the list of the setup options window, that is select the "Expand Filesystem" option and hit the enter key. We do this to make use of all the space present on the SD card as a full partition. All this does is, expand the OS to fit the whole space on the SD card which can then be used as the storage memory for the Pi.

The second thing to do:

select the third option in the list of the setup options window, that is select the "Enable Boot To Desktop/Scratch" option and hit the enter key. It will take you to another window called the "choose boot option" window that looks like the image below.



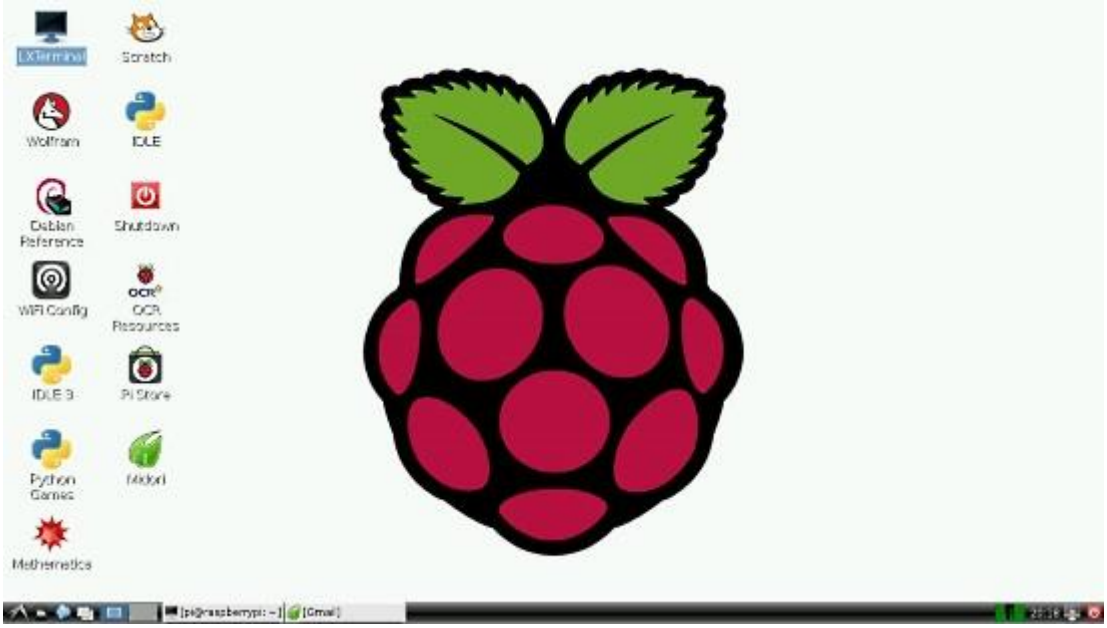
In the "choose boot option window" , select the second option, that is, "Desktop Log in as user 'pi' at the graphical desktop" and hit the enter button. Once done you will be taken back to the "Setup Options" page, if not select the "OK" button at the bottom of this window and you will be taken back to the previous window. We do this because we wan to boot into the desktop environment which we are familiar with. If we don't do this step then the Raspberry Pi boots into a terminal each time with no GUI options.

Once, both the steps are done, select the "finish" button at the bottom of the page and it should reboot automatically. If it doesn't, then use the following command in the terminal to reboot.

sudo reboot

4. **Updating the firmware**

After the reboot from the previous step, if everything went right, then you will end up on the desktop which looks like the image below.



Once you are on the desktop, open a terminal and enter the following command to update the firmware of the Pi.

sudo rpi-update

Updating the firmware is necessary because certain models of the Pi might not have all the required dependencies to run smoothly or it may have some bug. The latest firmware might have the fix to those bugs, thus its very important to update it in the beginning itself.

Install Apache, PHP and MySQL on Raspberry Pi:

The fundamental services required to turn your Raspberry Pi into a web server consist of Apache (the web server itself), PHP (scripting language) and MySQL (database server). When installed on a Linux based system, the collective term for these is LAMP.

Step 1 – Make sure you're up-to-date

Ensure you're running the latest system software. To check for updates and install them enter the following command:

```
sudo apt-get update
```

Step 2 – Change your hostname

Just to miminise the chances of any hickups we're going to change the hostname from the default 'raspberrypi'. From the terminal, enter:

```
sudo nano /etc/hostname
```

Nano editor will launch. Replace 'raspberrypi' with the domain from which your Raspberry Pi will run from. It's not essential to change the hostname, but it is something I have done. Once you've changed the hostname press CTRL + X to exit nano, and save when prompted. Now restart your Raspberry Pi to take to the new hostname:

```
sudo reboot
```

Once restarted, find out the FQDN to ensure your RPi has the correct hostname. A FQDN is a Fully Qualified Domain Name. To do this enter the following command once your RPi has restarted and you've logged back in with SSH:

```
hostname --fqd
```

If the hostname you entered is returned, congratulations! Go stick the kettle on and have a brew before moving onto the next part.

Step 3 – Install Apache

Here’s where the fun begins. We’re going to start by installing Apache and some other packages. To do this its begin with entering:

```
sudo bash
```

This means we’re not having to type sudo each time we run a command. When you’ve done this, enter the following:

```
apt-get install apache2 apache2-doc apache2-utils
```

This shouldn’t take long. Once we’ve done that we’re going to install a few support packages including PHP. Once complete, enter the following command:

```
apt-get install libapache2-mod-php5 php5 php-pear php5-xcache
```

This too shouldn’t take too long. Follow up with installing the support package for database connectivity:

```
apt-get install php5-mysql
```

Now we’re going to install MySQL server. Do this by entering the following command:

```
apt-get install mysql-server mysql-client
```

As part of the installation, you’ll be asked to set a root password. Enter a password and then confirm it when prompted in the blue screen.

Reboot your device, you're ready to start hosting from your Raspberry Pi.

Install PhpMyAdmin on your Raspberry Pi:

PhpMyAdmin is a handy web interface for managing local MySQL databases, and can make database queries, management and backups easy.

Step 1 – Begin the PhpMyAdmin installation

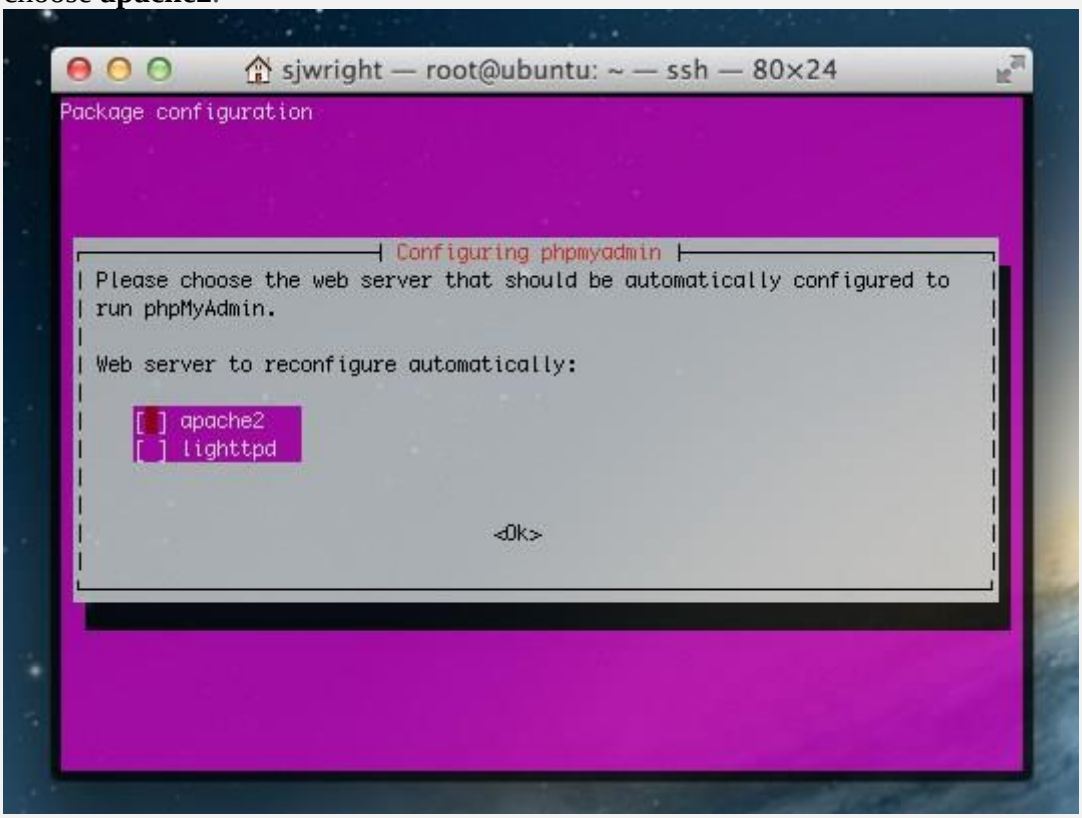
From terminal, we begin by changing to the root user in terminal by entering:

```
sudo bash
```

Now we need to install the PhpMyAdmin package using:

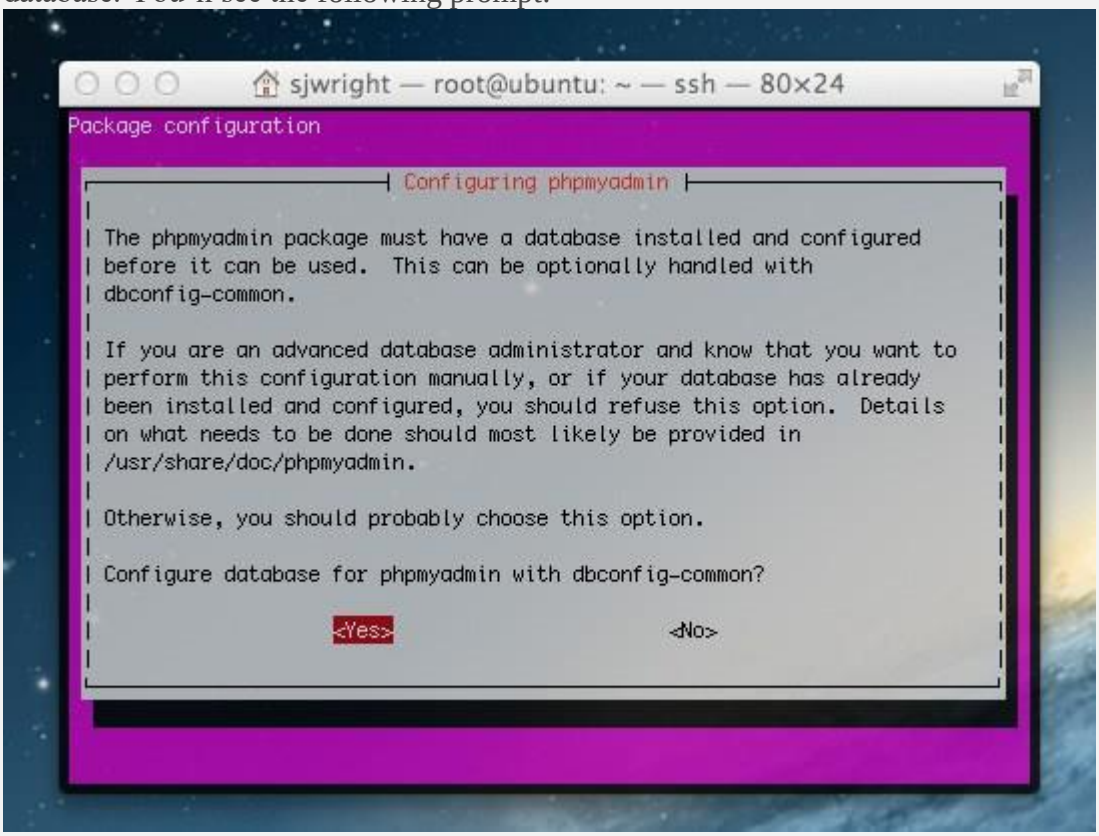
```
apt-get install phpmyadmin
```

The package will begin installing. You will be asked which web server is installed, choose **apache2**.



PhpMyAdmin installed on my Raspberry Pi so had to install it on a ubuntu VM for the purpose of this tutorial – apologies for the wrong colours, but I can assure you the procedure is the same for Debian/Raspbian and Ubuntu!

Step 2 – configure for dbconfig-common: Next we'll need to configure PhpMyAdmin's database. You'll see the following prompt:



When prompted, choose **Yes**. Next you'll be asked for an administrative password, this is the root password that was set during the MySQL installation
You'll be asked to set a password for PhpMySQL. I've used the same password as the MySQL root password, but its up to you what you set here. Make a note of it somewhere. That's PhpMyAdmin installed. Next we need to change the apache configuration to allow us to use http://your.raspberrypi.domain/phpmyadmin to access it.

Step 3 – Configure Apache to work with PhpMyAdmin
We need to alter the Apache configuration in order to access PhpMyAdmin. To do this, enter the following command to alter the configuration:

```
nano /etc/apache2/apache2.conf
```

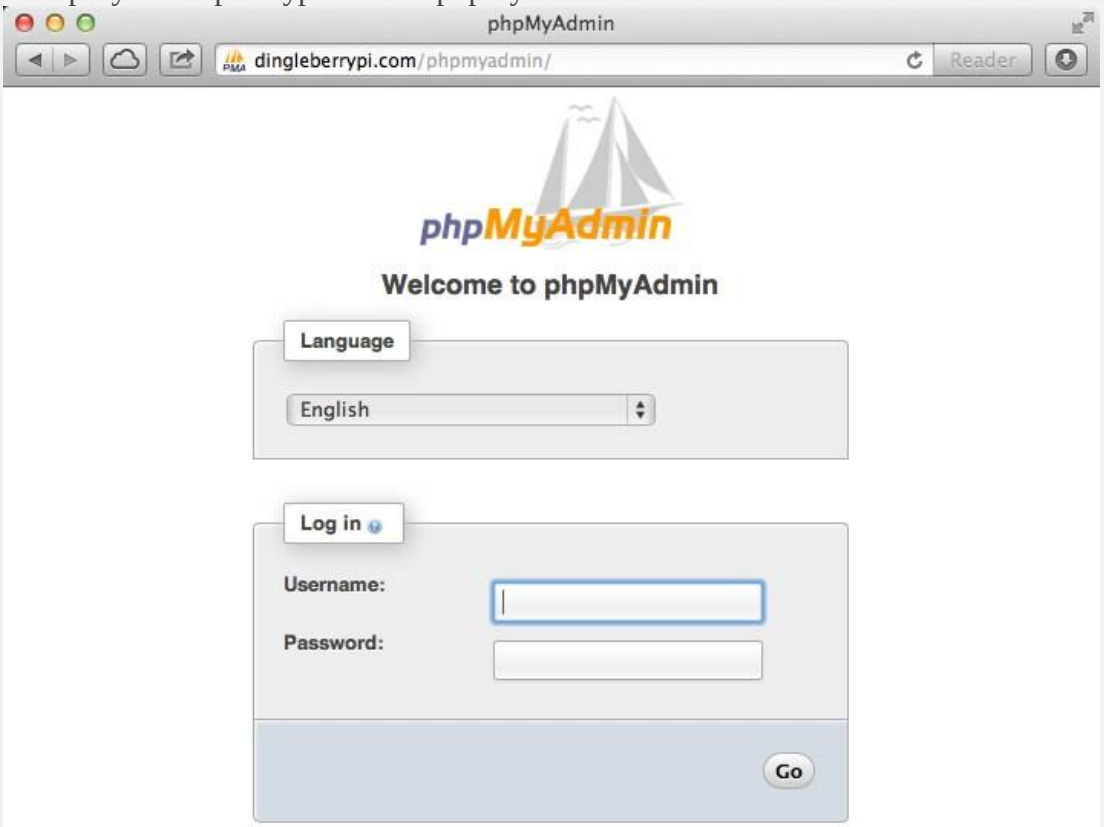
The configuration file will load in Nano. Navigate to the bottom of the file (keep pressing CTRL + V to jump page by page until you're at the bottom of the file) and add the following new line to the file:

```
Include /etc/phpmyadmin/apache.conf
```

Save the file (CTRL + X and enter Y when prompted to save) and restart Apache2. To restart Apache, enter the following command:

```
/etc/init.d/apache2 restart
```

That's it! You're all installed and ready to go. Give accessing it a try by going to your Raspberry Pi's IP address or domain name and add '/phpmyadmin' to the end in your web browser,
ie http://your.raspberrypi.domain/phpmyadmin.

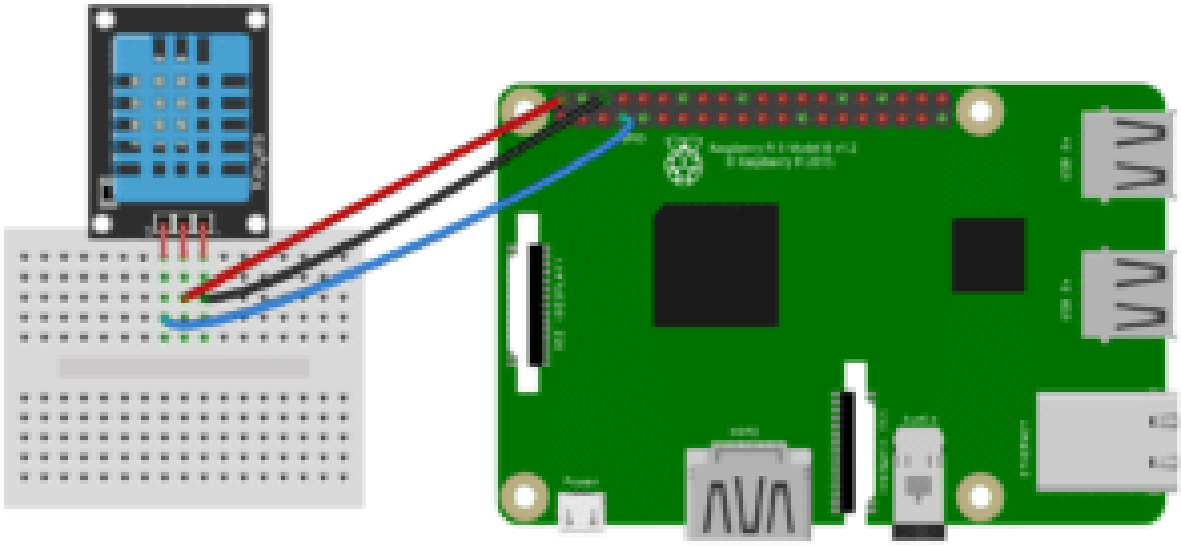


6.2 Retrieve Data From Sensors

Interfacing of DTH11 with raspberry pi:

THREE PIN DHT11 WITH SSH OUTPUT:

If you have a *three* pin DHT11 and want to output the humidity and temperature to an SSH terminal, wire it like this:



PROGRAMMING THE DHT11 WITH PYTHON:

We'll be using the Adafruit DHT11 Python library. You can download the library using Git, so if you don't have Git installed on your Pi already, enter this at the command prompt:

sudo apt-get install git-core

- Note: If you get an error installing Git, run sudo apt-get update and try it again.

To install the Adafruit DHT11 library:

1. Enter this at the command prompt to download the library:

git clone https://github.com/adafruit/Adafruit_Python_DHT.git

2. Change directories with:

cd Adafruit_Python_DHT

3. Now enter this:

sudo apt-get install build-essential python-dev

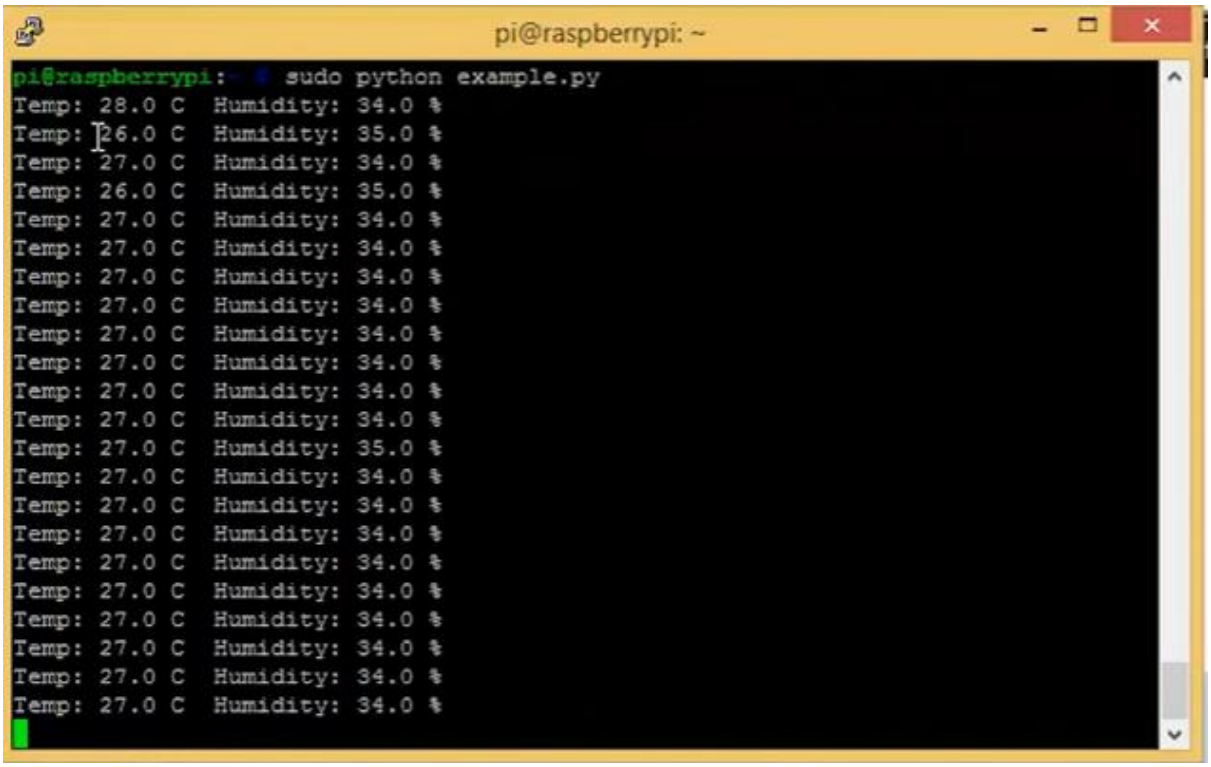
4. Then install the library with:

sudo python setup.py install

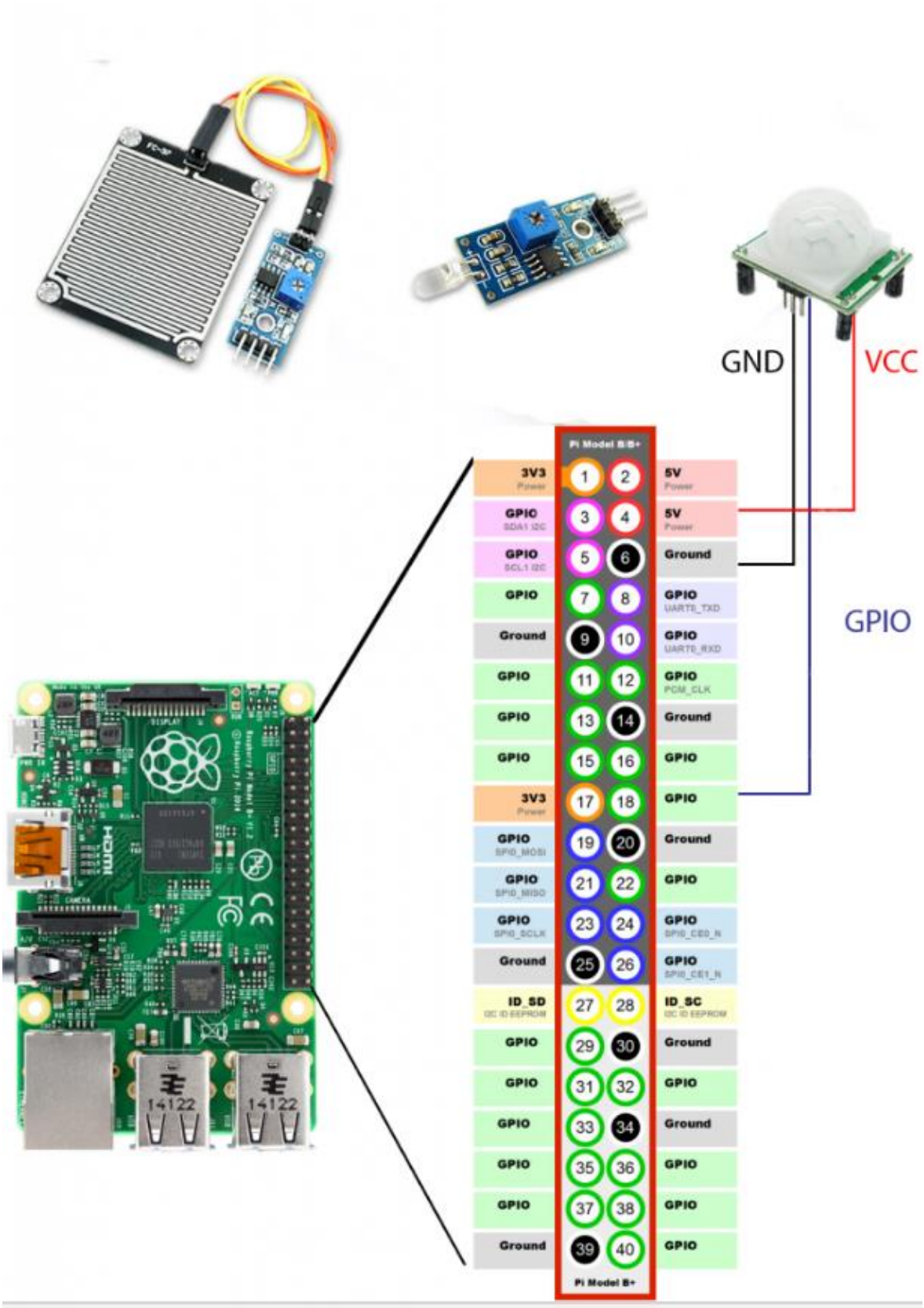
This Python program will output the temperature and humidity readings to an SSH terminal:

```
#!/usr/bin/python
import sys
import Adafruit_DHT

while True:

    humidity, temperature = Adafruit_DHT.read_retry(11, 4)

    print 'Temp: {0:0.1f} C  Humidity: {1:0.1f} %'.format(temperature, humidity)
```

OUTPUT TO AN SSH TERMINAL:



Interfacing of Rain sensor with raspberry pi:



Make Connection of Rain Sensor with Raspberry Pi as mentioned below:

1.1: Connect Vcc Pin of Sensor with 5V Pin of Raspberry Pi.

1.2: Connect GND Pin of Sensor with GND Pin of Raspberry Pi.

1.3: Connect DO Pin of Sensor to 18 Pin of Raspberry Pi. (As Programmed in Code)

Programming Rain Sensor with python:

```
#!/usr/bin/python
import sys
sys.path.append('/home/pi/Adafruit_Python_DHT/Adafruit_python_BMP/examples')

import bmp
import MySQLdb
import Adafruit_DHT
import time
import datetime
import RPi.GPIO as GPIO # This is the GPIO library we need to use the GPIO pins on the Raspberry Pi
import smtplib # This is the SMTP library we need to send the email notification

GPIO.setmode(GPIO.BCM)

channel1 = 17
channel2 = 23

GPIO.setup(channel1,GPIO.IN)
GPIO.setup(channel2,GPIO.IN)



# This line tells our script to keep an eye on our gpio pin and let us know when the pin goes HIGH or LOW
#GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)
# This line asigns a function to the GPIO pin so that when the above line tells us there is a change on the pin, run this function
#GPIO.add_event_callback(channel, callback)

while True:


   if GPIO.input(channel2):
           print "Water not detected"
           rainStatus = "Its not raining!!"
           sendEmail(message_dead2)
   else:
           print "Water  detected"
```

```
            rainStatus = "It is raining!!"
            sendEmail(message_alive2)

    if GPIO.input(channel1):
                    print "LED off"
            waterStatus = "No Water"
                    sendEmail(message_dead1)
                    actionTaken = "Farmer Notified"
    else:
                    print "LED on"
            waterStatus = "Water"
            sendEmail(message_alive1)
            actionTaken = "Farmer Notified
```

Setting up email notification Service:

The **smtplib** module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult **RFC 821** (Simple Mail Transfer Protocol) and **RFC 1869** (SMTP Service Extensions).

```
import smtplib # This is the SMTP library we need to send the email notification


# Define some variables to be used later on in our script

# You might not need the username and password variable, depends if you are using a provider or if you have your raspberry pi setup to send emails
# If you have setup your raspberry pi to send emails, then you will probably want to use 'localhost' for your smtp_host

smtp_username = "rbsarojtest@gmail.com" # This is the username used to login to your SMTP provider
smtp_password = "pass$123" # This is the password used to login to your SMTP provider
smtp_host = "smtp.gmail.com" # This is the host of the SMTP provider
smtp_port = 587 # This is the port that your SMTP provider uses
# TLS = 587(default) for gmail port and SSL = 465 for gmail port
smtp_sender = "rbsarojtest@gmail.com" # This is the FROM email address
smtp_receivers = ['razsaroj@gmail.com','sarun0795@gmail.com'] # This is the TO email address

# The next two variables use triple quotes, these allow us to preserve the line breaks in the string.

# This is the message tha
#t will be sent when NO moisture is detected
```

```
waterStatus = ""
rainStatus = ""
actionTaken = ""

message_dead1= """From: Sender Name <rbsarojtest@gmail.com>
To: Receiver Name <razsaroj@gmail.com>
Subject: Moisture Sensor Notification

Warning, no moisture detected! Plant death imminent!!! :'(
"""

# This is the message that will be sent when moisture IS detected again

message_alive1 = """From: Sender Name <rbsarojtest@gmail.com>
To: Receiver Name <razsaroj@gmail.com>
Subject: Moisture Sensor Notification

Panic over! Plant has water again :)
"""


#rain sensor notification
message_dead2 = """From: Sender Name <rbsarojtest@gmail.com>
To: Receiver Name <razsaroj@gmail.com>
Subject: Rain Sensor Notification

Cool , its not raining :'(
"""

# This is the message that will be sent when moisture IS detected again

message_alive2 = """From: Sender Name <rbsarojtest@gmail.com>
To: Receiver Name <razsaroj@gmail.com>
Subject: Rain Sensor Notification

Warning: its raining protect your crops :)
"""




# This is our sendEmail function
def sendEmail(smtp_message):
        try:
                smtpObj = smtplib.SMTP(smtp_host, smtp_port)
            smtpObj.starttls()
            smtpObj.ehlo()
```

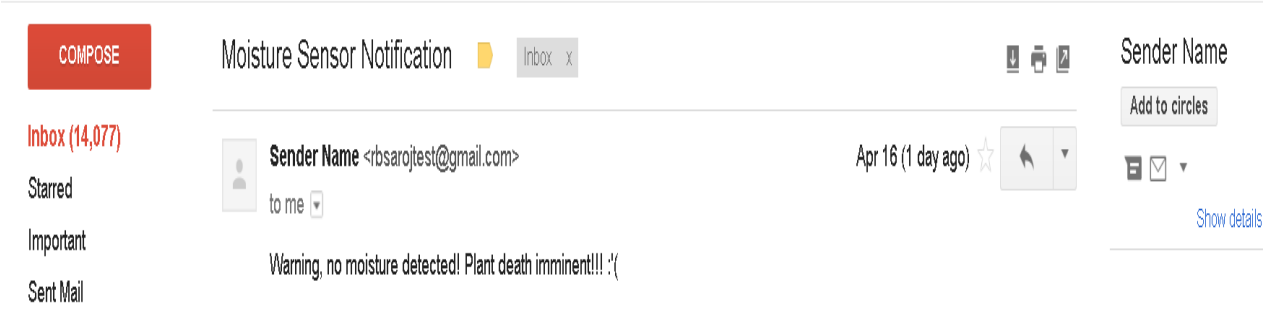
```
            smtpObj.login(smtp_username, smtp_password) # If you don't need to login to
your smtp provider, simply remove this line
            smtpObj.sendmail(smtp_sender, smtp_receivers, smtp_message)
            print "Successfully sent email"

        except smtplib.SMTPException:
            print "Error: unable to send email"
            actionTaken = "Farmer not notified"
```

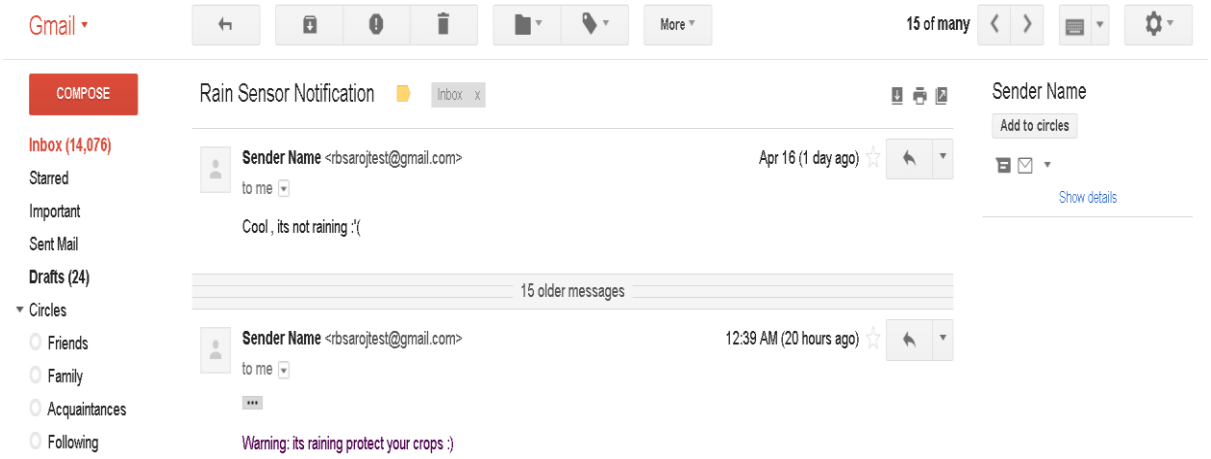
class smtplib.**SMTP**([*host*[, *port*[, *local_hostname*[, *timeout*]]]])

An **SMTP** instance encapsulates an SMTP connection. It has methods that support a full repertoire of SMTP and ESMTP operations. If the optional host and port parameters are given, the SMTP **connect()** method is called with those parameters during initialization. If specified, *local_hostname* is used as the FQDN of the local host in the HELO/EHLO command. Otherwise, the local hostname is found using **socket.getfqdn()**. If the **connect()** call returns anything other than a success code, an **SMTPConnectError** is raised. The optional *timeout* parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default timeout setting will be used). If the timeout expires, **socket.timeout** is raised.

For normal use, you should only require the initialization/connect, **sendmail()**, and **quit()** methods

Output of Email Notification Service





Saving the values in database:

```
# This is an infinte loop to keep our script running
    if True:
     humidity , temperature= Adafruit_DHT.read_retry(11, 4)
     #read preesure value from bmp location in different folder
     bmp.read_bmp()
     date=time.strftime("%Y/%m%/%d")
     clock=time.strftime("%H:%M")
     db = MySQLdb.connect(host="localhost", user="root",passwd="raspberry", db="test")
     sql = "INSERT INTO
test2(Date,Time,Humidity,Temperature,MoistureStatus,Rainstatus,ActionTaken)
VALUES('%s','%s','%s','%s','%s','%s','%s')" % \
            (date,clock,humidity,temperature,waterStatus,rainStatus,actionTaken)
     cur = db.cursor()
     try:
       cur.execute("select * from test2")
     except:
       print "No value to display"
     try:
     # cur.execute("INSERT INTO test2(Date,Time,Humidity,
Temperature,MoistureStatus,Rainstatus,ActionTaken) VALUES
(%s,%s,%s,%s,%s)",(date,clock,humidity,temperature,waterStatus))
       cur.execute(sql)
       print "values successfully inserted"

       db.commit()
     except:
       print "failed to insert"
     try:
       #merge two tables test2 and test3
       cur.execute("INSERT INTO test4
(Date,Time,Humidity,Temperature,MoistureStatus,Rainstatus,ActionTaken,Pressure,Altitude
,Sealevel) Select * from test2 p INNER JOIN test3 q ON p.ID = q.ID")
```

```
      print "new table successfully created"
     # r = db.store_result()
     # for row in r.fetch_row(0):
     #     print row
    except:
       print "Values Fetched"
    db.rollback()

    db.close()
    print 'Temp: {0:0.1f} C  Humidity: {1:0.1f} %'.format(temperature,humidity)

    # This line simply tells our script to wait 0.1 of a second, this is so the script doesnt hog all
of the CPU
    time.sleep(0)
```

MySQLdb is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2.0 and is built on top of the MySQL C API.

How do I Install MySQLdb?

Before proceeding, you make sure you have MySQLdb installed on your machine. Just type the following in your Python script and execute it:

```
#!/usr/bin/python


import MySQLdb
```

If it produces the following result, then it means MySQLdb module is not installed:

```
Traceback (most recent call last):
  File "test.py", line 3, in <module>
    import MySQLdb
ImportError: No module named MySQLdb
```

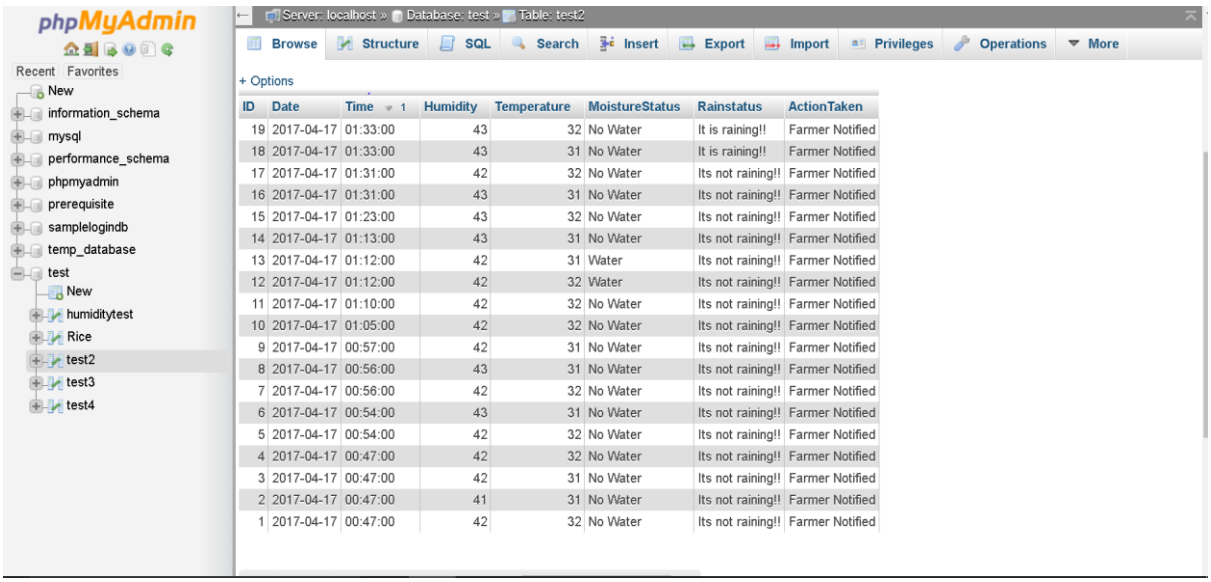
To install MySQLdb module, use the following command:

```
For Ubuntu, use the following command -
$ sudo apt-get install python-pip python-dev libmysqlclient-dev
For Fedora, use the following command -
$ sudo dnf install python python-devel mysql-devel redhat-rpm-config gcc
```

For Python command prompt, `use` the following command -

pip install MySQL-python

Output of Database:



phpMyAdmin

Recent Favorites

New
information_schema
mysql
performance_schema
phpmyadmin
prerequisite
samplelogindb
temp_database
test
New
humiditytest
Rice
test2
test3
test4

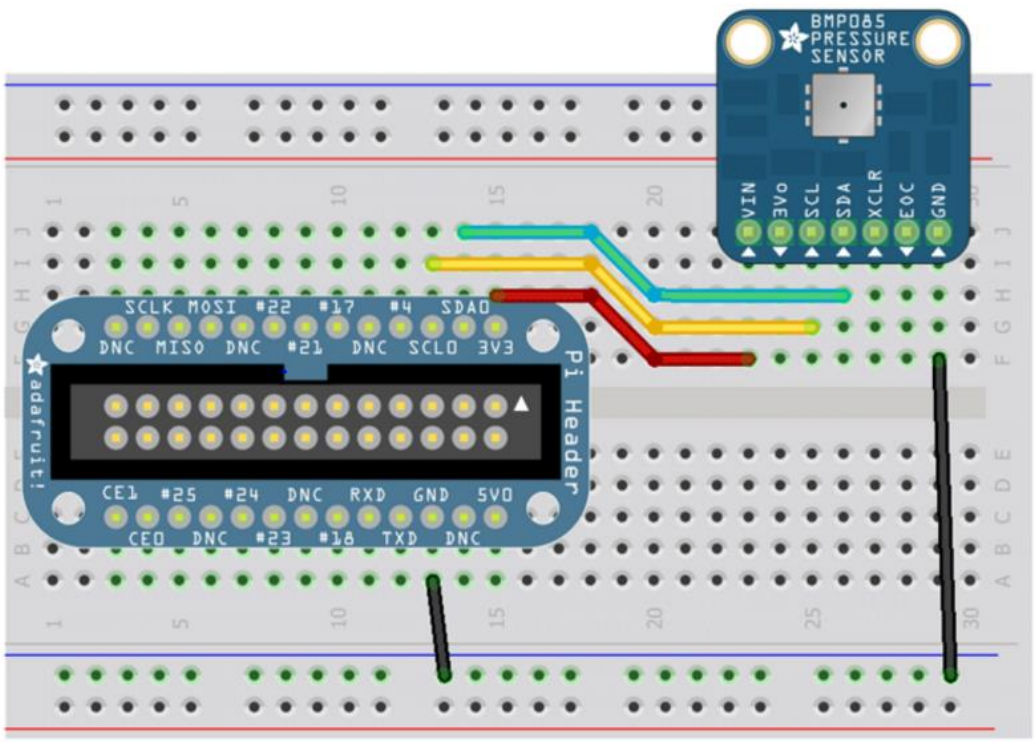
Server: localhost » Database: test » Table: test2

Browse Structure SQL Search Insert Export Import Privileges Operations More

+ Options

ID	Date	Time	Humidity	Temperature	MoistureStatus	Rainstatus	ActionTaken
19	2017-04-17	01:33:00	43	32	No Water	It is raining!!	Farmer Notified
18	2017-04-17	01:33:00	43	31	No Water	It is raining!!	Farmer Notified
17	2017-04-17	01:31:00	42	32	No Water	Its not raining!!	Farmer Notified
16	2017-04-17	01:31:00	43	31	No Water	Its not raining!!	Farmer Notified
15	2017-04-17	01:23:00	43	32	No Water	Its not raining!!	Farmer Notified
14	2017-04-17	01:13:00	43	31	No Water	Its not raining!!	Farmer Notified
13	2017-04-17	01:12:00	42	31	Water	Its not raining!!	Farmer Notified
12	2017-04-17	01:12:00	42	32	Water	Its not raining!!	Farmer Notified
11	2017-04-17	01:10:00	42	32	No Water	Its not raining!!	Farmer Notified
10	2017-04-17	01:05:00	42	32	No Water	Its not raining!!	Farmer Notified
9	2017-04-17	00:57:00	42	31	No Water	Its not raining!!	Farmer Notified
8	2017-04-17	00:56:00	43	31	No Water	Its not raining!!	Farmer Notified
7	2017-04-17	00:56:00	42	32	No Water	Its not raining!!	Farmer Notified
6	2017-04-17	00:54:00	43	31	No Water	Its not raining!!	Farmer Notified
5	2017-04-17	00:54:00	42	32	No Water	Its not raining!!	Farmer Notified
4	2017-04-17	00:47:00	42	32	No Water	Its not raining!!	Farmer Notified
3	2017-04-17	00:47:00	42	31	No Water	Its not raining!!	Farmer Notified
2	2017-04-17	00:47:00	41	31	No Water	Its not raining!!	Farmer Notified
1	2017-04-17	00:47:00	42	32	No Water	Its not raining!!	Farmer Notified

Interfacing Of Barometric sensor:



The Raspberry Pi and Beaglebone Black include support for Python, which makes it easy to get access to a lot of low-level hardware and software stacks -- USB, TCP/IP, multiple file systems etc. This is a good thing since it means you don't need to wrap your head around all the obscure details that go along with these complex stacks or the implementation details of various serial buses: you can focus on getting your data off your sensor and into your project as quickly as possible. Hurray for abstraction!

Most sensors tend to communicate with other devices based on one of three well-defined mechanisms: **I2C**, **SPI** or good old **analog output**. There are dozens of other serial buses and communication protocols out there (CAN, 1-Wire, etc.), and they all have their strengths and weaknesses, but I2C, SPI and analog cover the overwhelming majority of sensors you're likely to hook up to your development board.

I2C is a particularly useful bus with the for two main reasons:

- It only requires two shared lines: **SCL** for the clock signal, and **SDA** for the bi-direction data transfers.
- Each I2C device uses a unique 7-bit address, meaning you can have more than 120 unique I2C devices sharing the bus, and you can freely communicate with them one at a time on an as-needed basis.

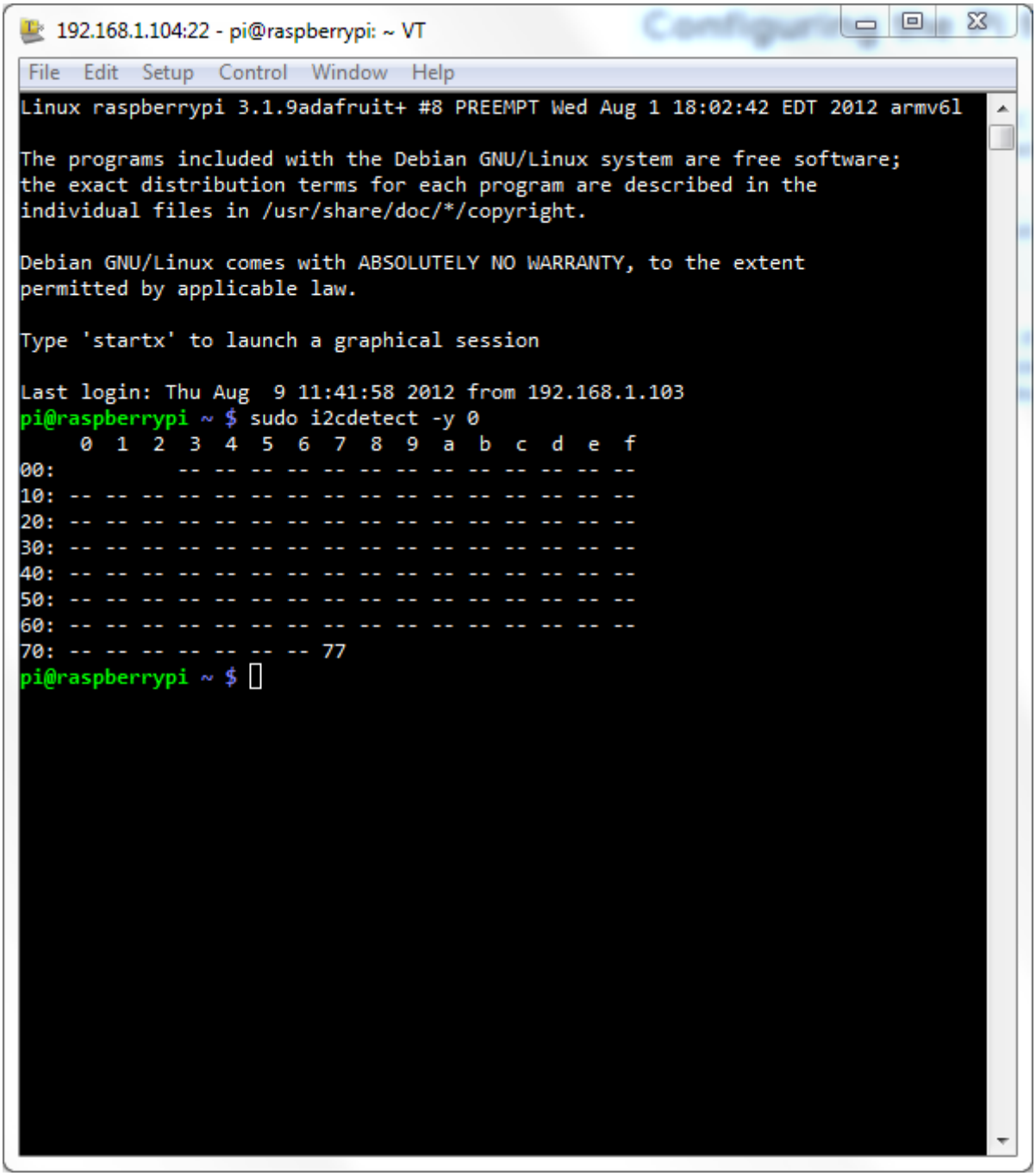
This tutorial will show you how you can read data from the I2C-based BMP085 or BMP180 Barometric Pressure Sensor using Python on a Raspberry Pi or Beaglebone Black.

Configuring the Pi for I2C

If you're using a Raspberry Pi, follow the steps below to configure it to use the I2C interface.

```
sudo i2cdetect -y 0 (if you are using a version 1 Raspberry Pi)
sudo i2cdetect -y 1 (if you are using a version 2 Raspberry Pi)
```

This will search /dev/i2c-0 or /dev/i2c-1 for all address, and if an Adafruit BMP085 Breakout is properly connected it should show up at 0x77 as follows:



Once both of these packages have been installed, you have everything you need to get started accessing I2C and SMBus devices in Python.

Programming for BMP180(Barometric Sensor):

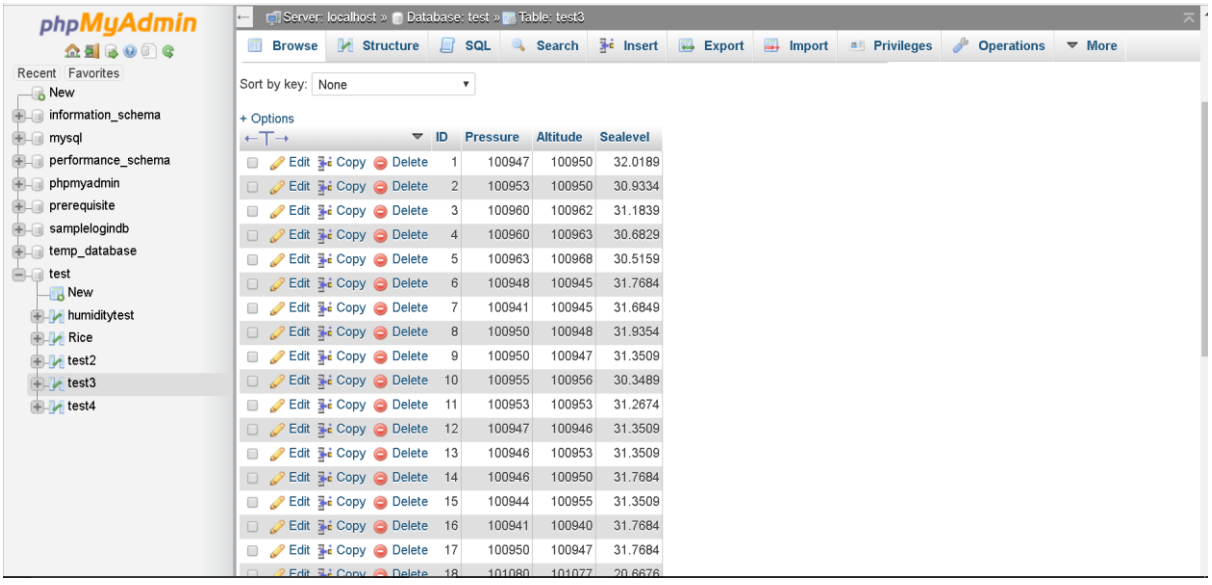
def read_bmp():

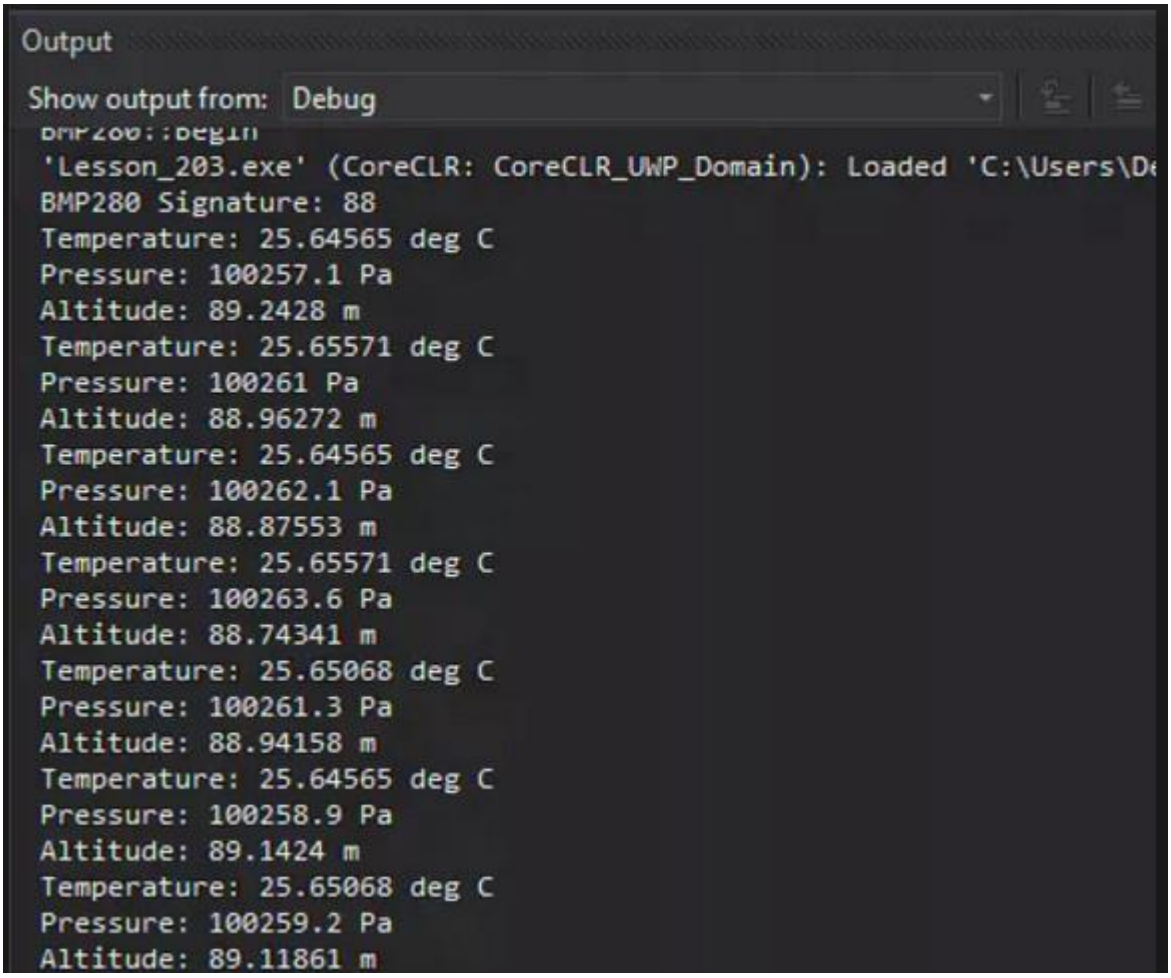
global press,sl,alt

print('Temp = {0:0.2f} *C'.format(sensor.read_temperature()))
print('Pressure = {0:0.2f} Pa'.format(sensor.read_pressure()))

```
print('Altitude = {0:0.2f} m'.format(sensor.read_altitude()))
print('Sealevel Pressure = {0:0.2f} Pa'.format(sensor.read_sealevel_pressure()))
press = sensor.read_pressure()
sl = sensor.read_sealevel_pressure()
alt = sensor.read_altitude()

db = MySQLdb.connect(host="localhost", user="root",passwd="raspberry", db="test")
sql = "INSERT INTO test3(Pressure,Altitude,Sealevel) VALUES('%s','%s','%s')" % \
          (press,sl,alt)
cur = db.cursor()
try:
    cur.execute(sql)
    print "values successfully inserted"
    db.commit()
except:
    print "failed to insert"
    db.rollback()

db.close()
```

Output of Database:





Php code Sample For graphical data representation:

Live Data Of Humidy Sensor:

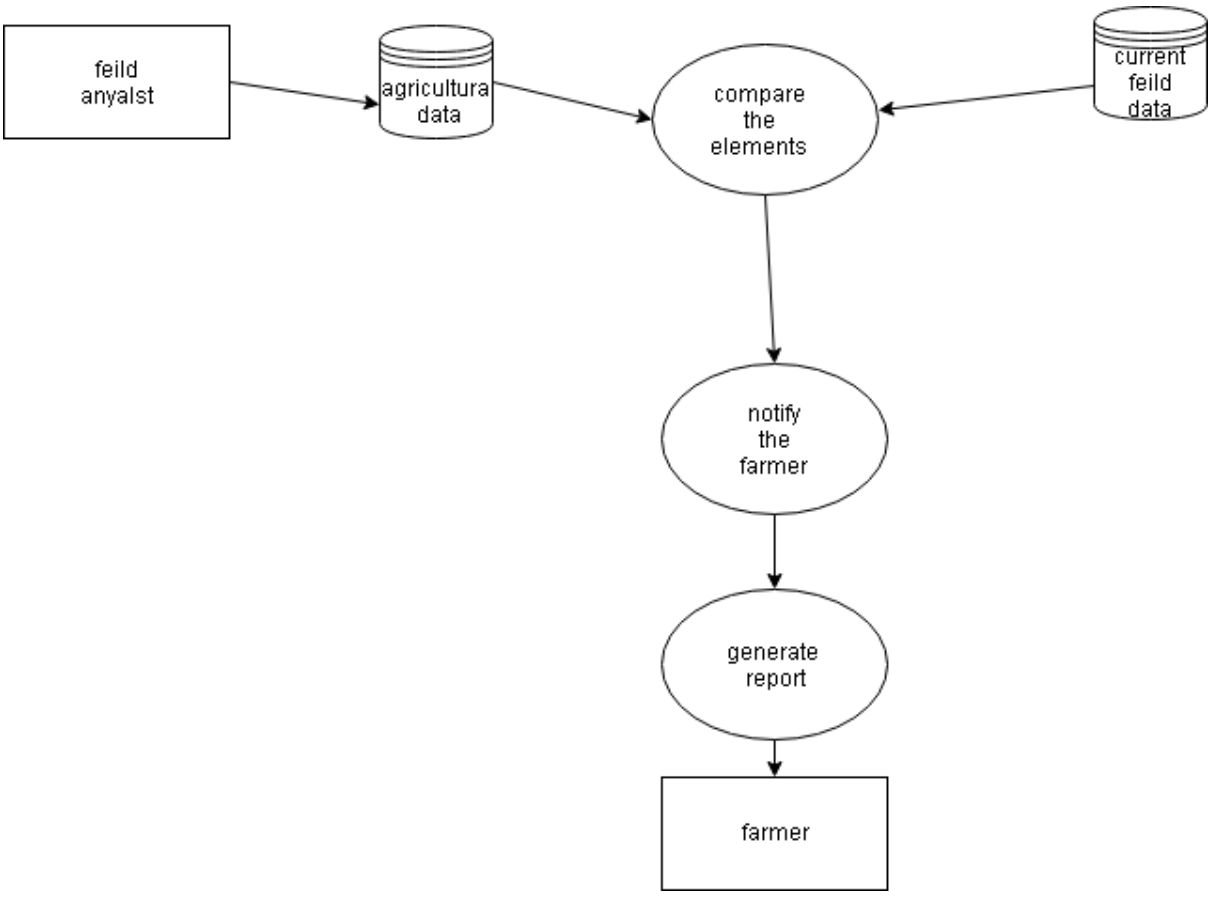
```
<?php
//setting header to json
header('Content-Type: application/json');

//database
define('DB_HOST', '127.0.0.1');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', 'raspberry');
define('DB_NAME', 'test');
```

```
//get connection
$mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_NAME);

if(!$mysqli){
	die("Connection failed: " . $mysqli->error);
}

//query to get data from the table
$query = sprintf("SELECT ID,Humidity  FROM test2");

//execute query
$result = $mysqli->query($query);

//loop through the returned data
$data = array();
foreach ($result as $row) {
	$data[] = $row;
}

//free memory associated with result
$result->close();

//close connection
$mysqli->close();

//now print the data
print json_encode($data);
```

6.3 Building prediction logic

This process is the most crucial process of our system in this process we are going to match the values obtained by the sensors with the values of the agricultural data now this agricultural data is the data obtained from the field analyst or agricultural expert research and this data show the optimum amount of moisture, humidity, temperature and salinity required to grow good crop and produce profitable cultivation.

The data collected from sensors are stored in test2 table and data collected from field analyst is stored in the test3 table now the we compare these two tables in mysql database and if the values is less then the optimum values specified in the agricultural data we will show that the is the deficiency of that particular element in the field and if the values of the sensors matches the optimum values of agricultural data we will show that his field are working fine like this way he can keep an eye on his field anytime and from anywhere and the second thing is the prediction logic directs the farmer towards efficient crop growth and profitable cultivation.



PHP code for Building Prediction Logic

This program checks whether the humidity is in optimum level or not and based on that it predicts whether the crops are in good condition or bad conditons;

```
while ($inRangerow = mysql_fetch_object($resulttest)) {

   //echo "from rice humidity->" .$inRangerow-> humid ;
   echo "<br>";
   if($inRangerow->humid>=$row->minhumidity and $inRangerow->humid<=$row-
>maxhumidity){
      echo "<br>";
      echo "humidity is in good condition";
     // echo "from rice humidity->" .$inRangerow-> humid ;

    }
```

```
    elseif($inRangerow->humid<=$row->minhumidity){
      echo "humidity is less do something";
    }
    elseif($inRangerow->humid>=$row->maxhumidity){
      echo "humidity is more do something";
    }
    echo "<br>";
    echo $inRangerow->humid;

}
```

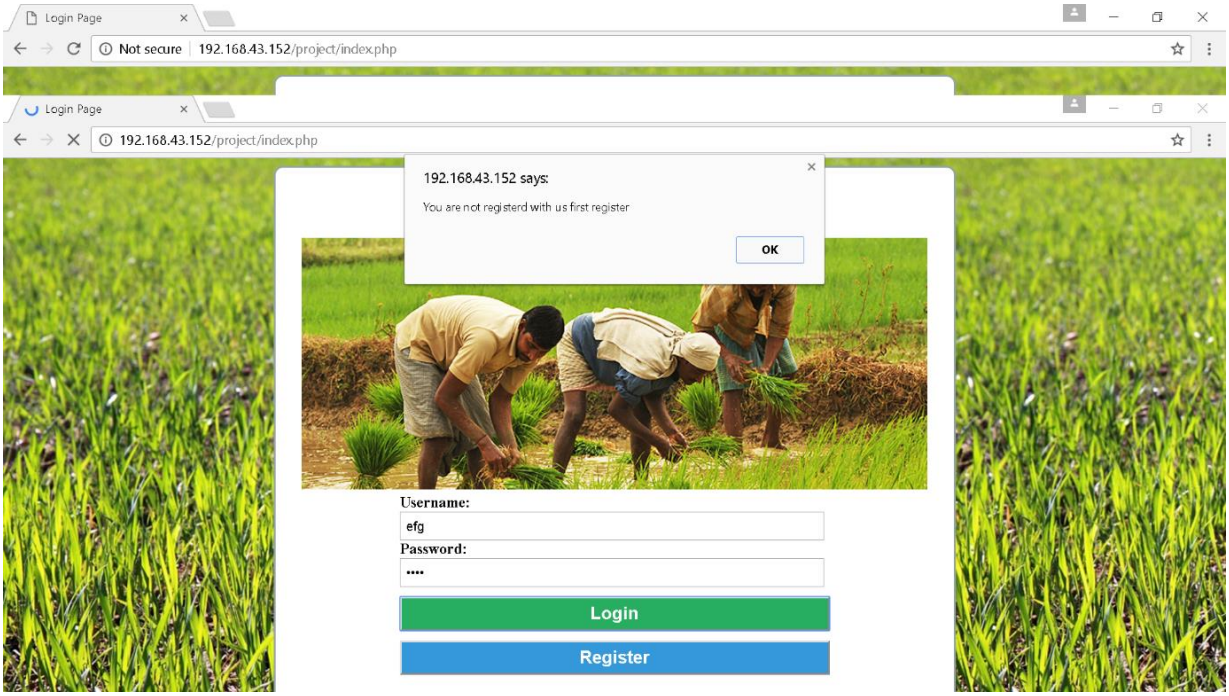
Chapter 7

Testing

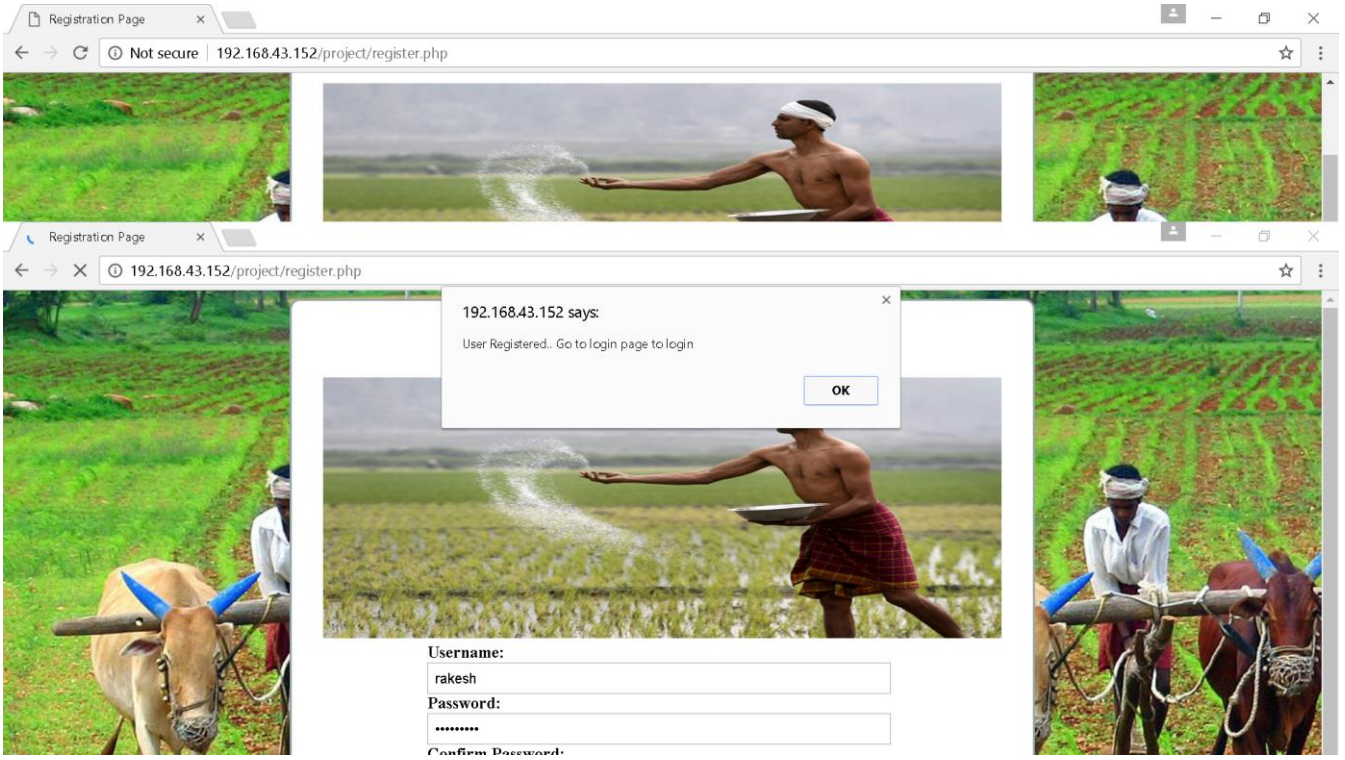
7.Testing

7.1 Test Cases

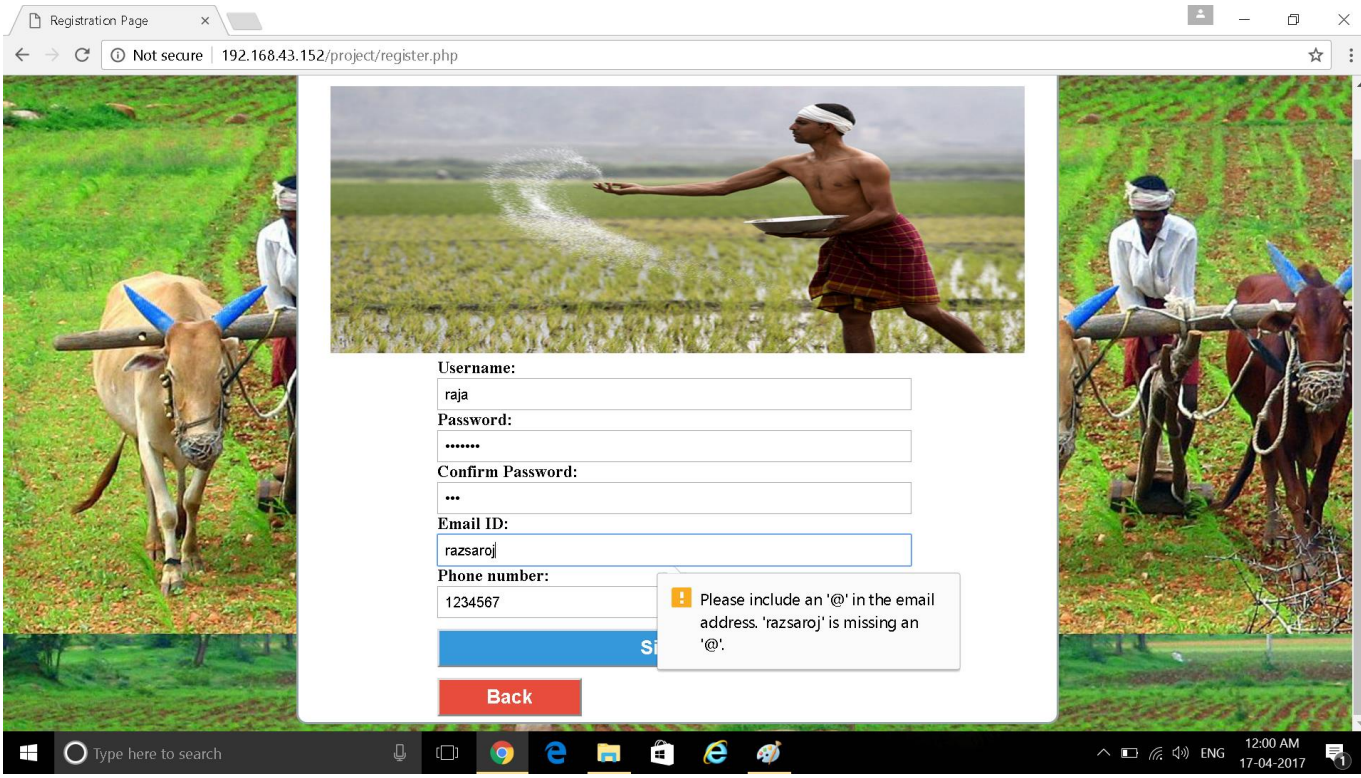
Case 1: User not Registered



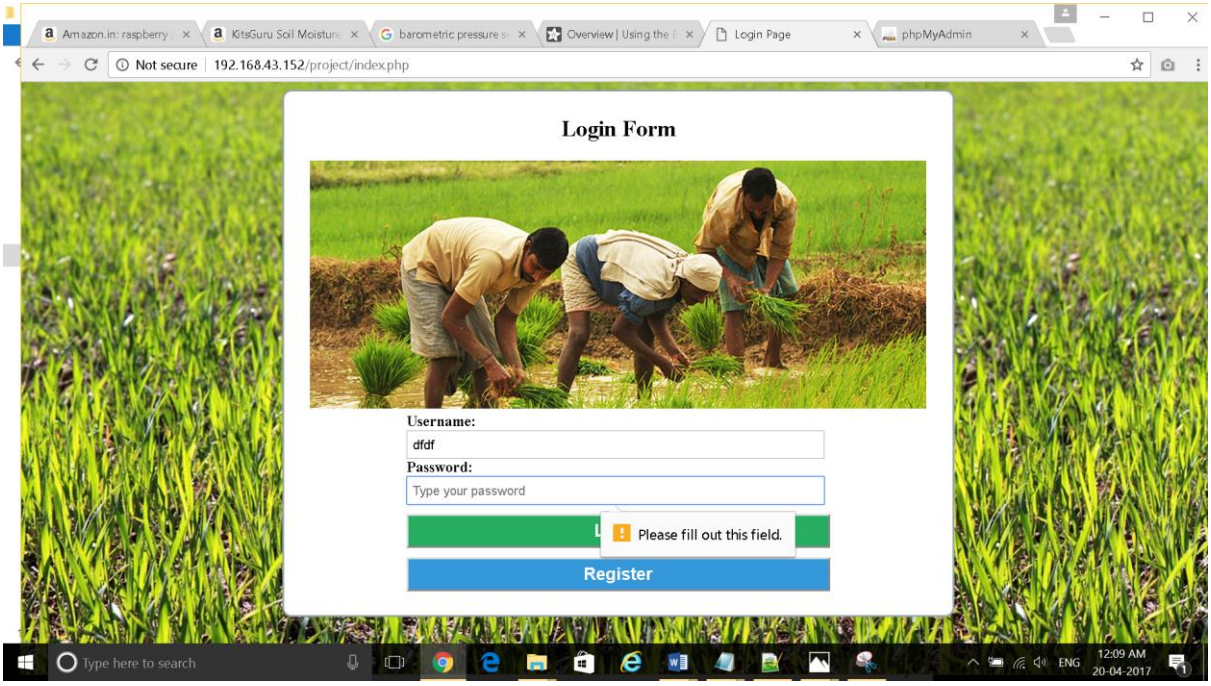
Case 2: User registered



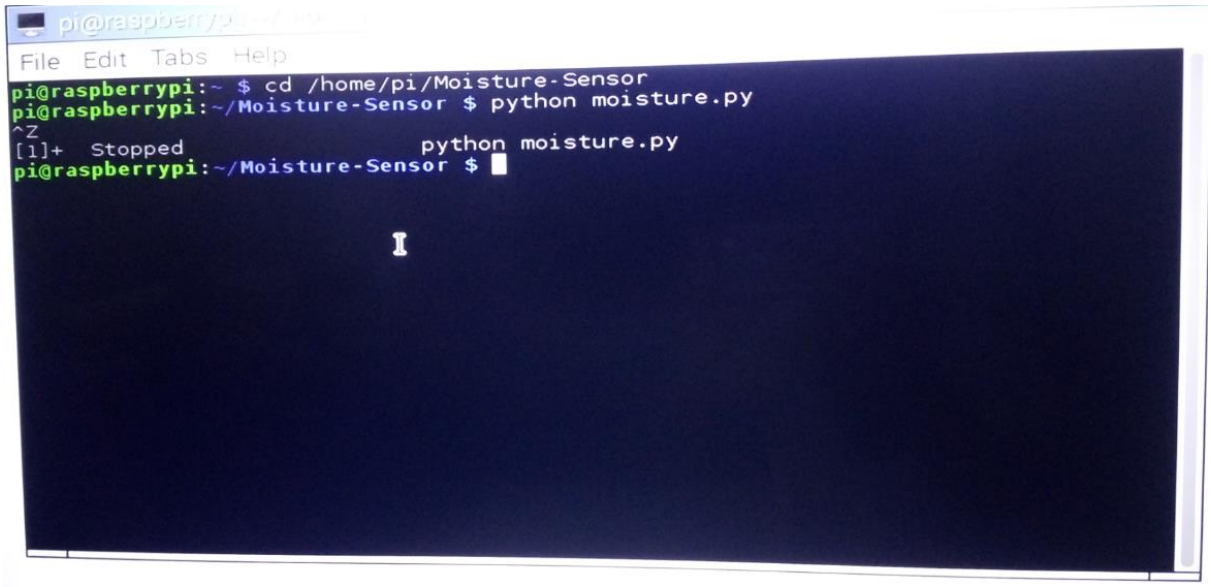
Case 3: Email verification



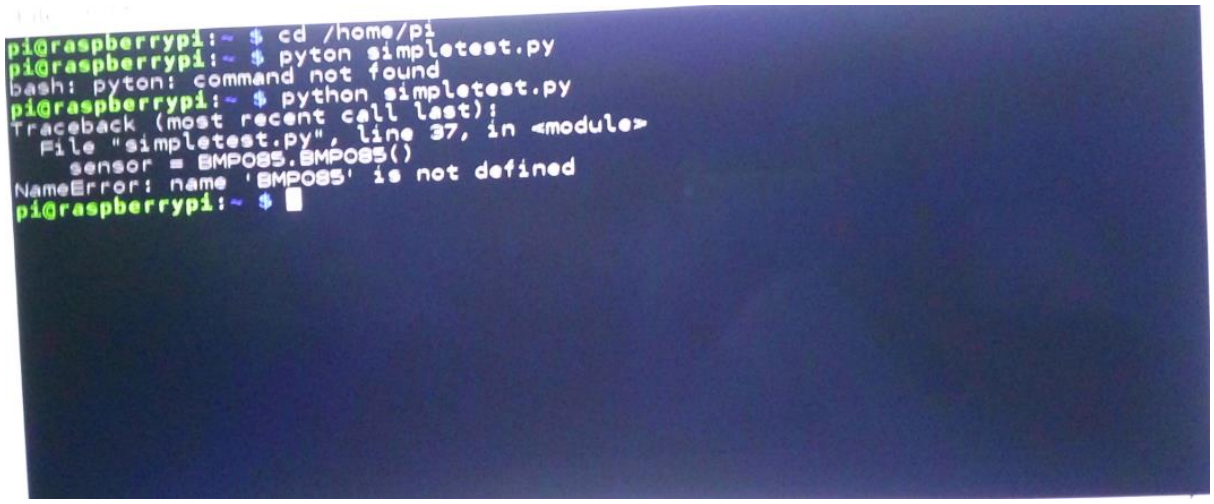
Case 4: Password Field blank:



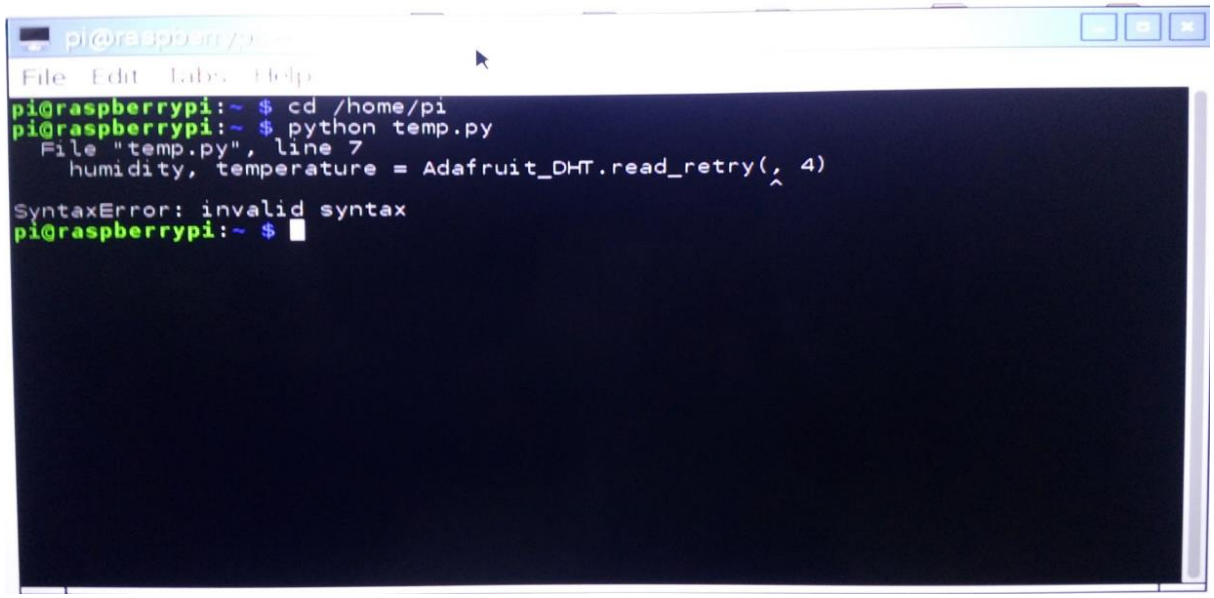
Case 5: Moisture Sensor stopped working



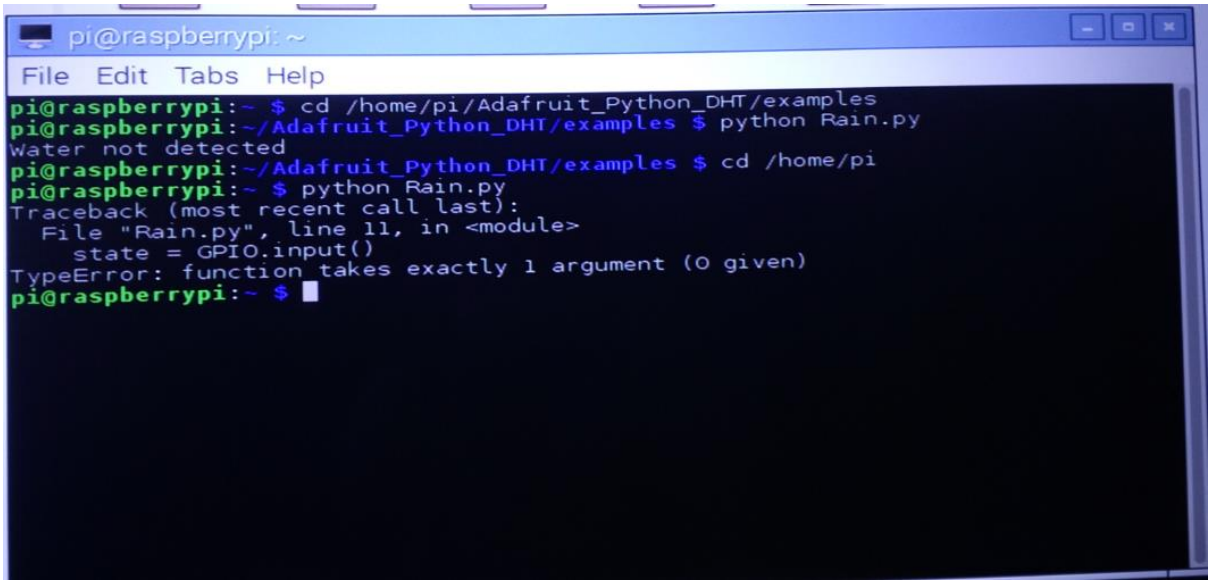
Case 6: BMP085 library file missing



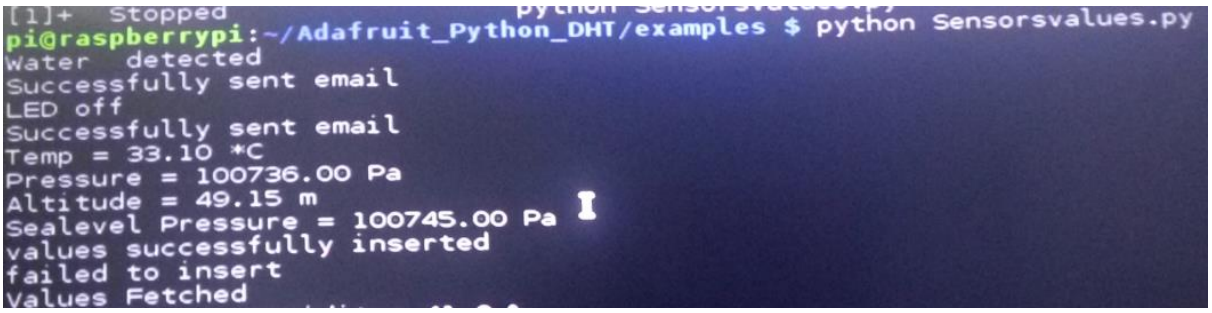
Case 7: SyntaxError



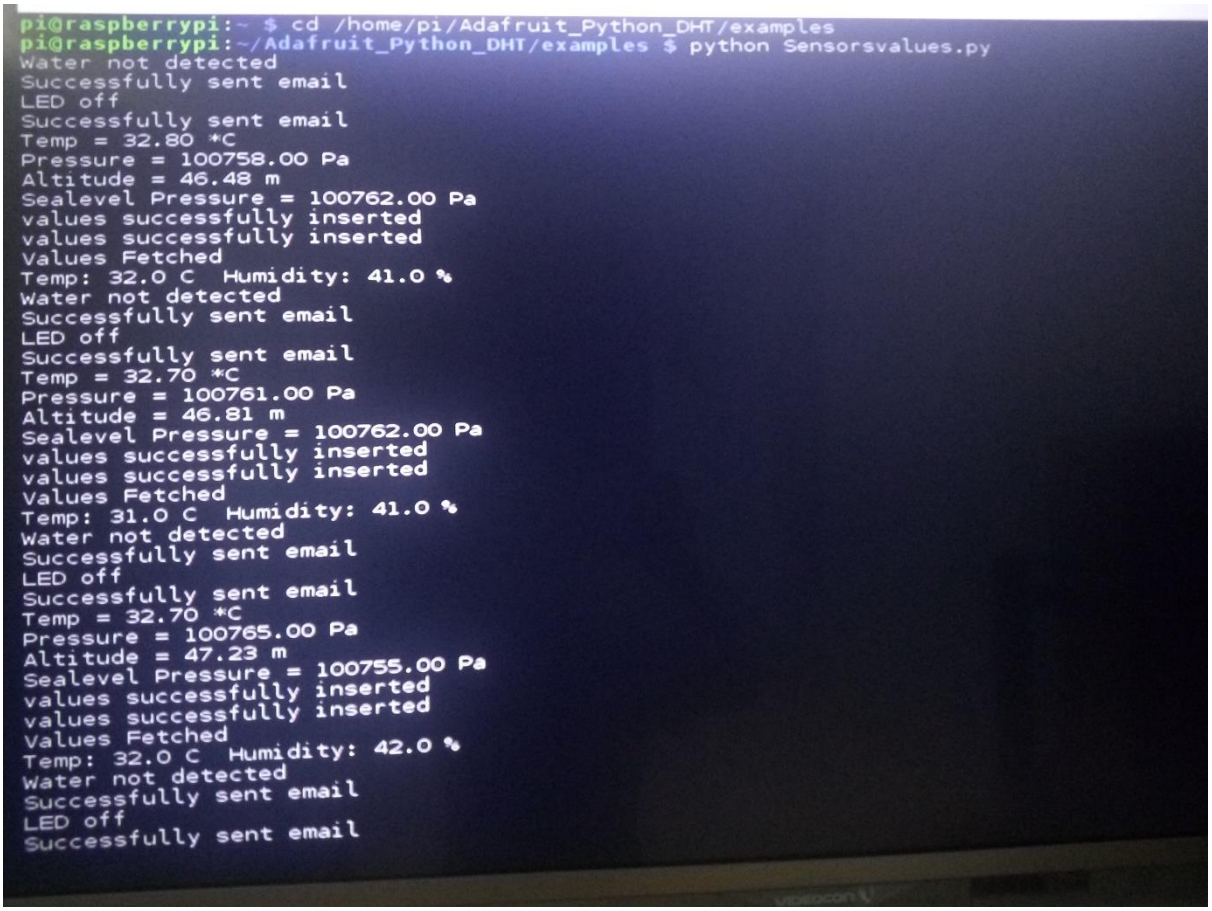
Case 8: type error



Case 9: Unable to insert Values in Database



Case 10: All the values are inserted with email notification

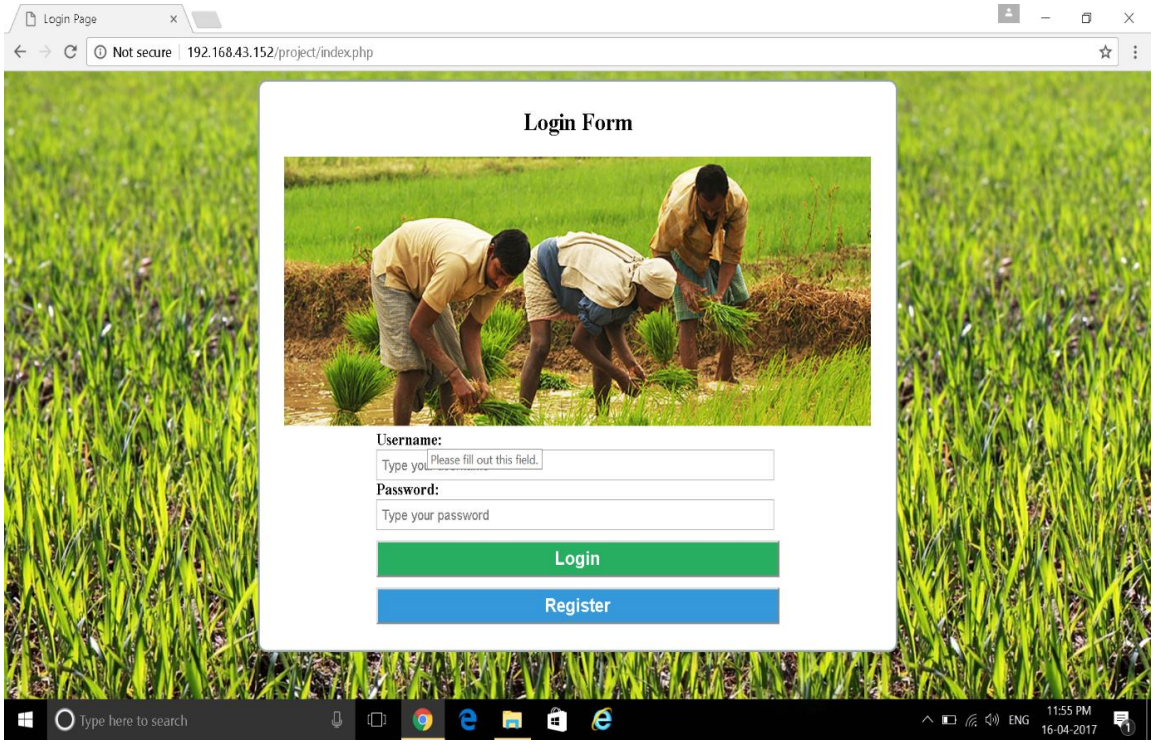

pi@raspberrypi:~ $ cd /home/pi/Adafruit_Python_DHT/examples
pi@raspberrypi:~/Adafruit_Python_DHT/examples $ python Sensorsvalues.py
Water not detected
Successfully sent email
LED off
Successfully sent email
Temp = 32.80 *C
Pressure = 100758.00 Pa
Altitude = 46.48 m
Sealevel Pressure = 100762.00 Pa
values successfully inserted
values successfully inserted
Values Fetched
Temp: 32.0 C Humidity: 41.0 %
Water not detected
Successfully sent email
LED off
Successfully sent email
Temp = 32.70 *C
Pressure = 100761.00 Pa
Altitude = 46.81 m
Sealevel Pressure = 100762.00 Pa
values successfully inserted
values successfully inserted
Values Fetched
Temp: 31.0 C Humidity: 41.0 %
Water not detected
Successfully sent email
LED off
Successfully sent email
Temp = 32.70 *C
Pressure = 100765.00 Pa
Altitude = 47.23 m
Sealevel Pressure = 100755.00 Pa
values successfully inserted
values successfully inserted
Values Fetched
Temp: 32.0 C Humidity: 42.0 %
Water not detected
Successfully sent email
LED off
Successfully sent email

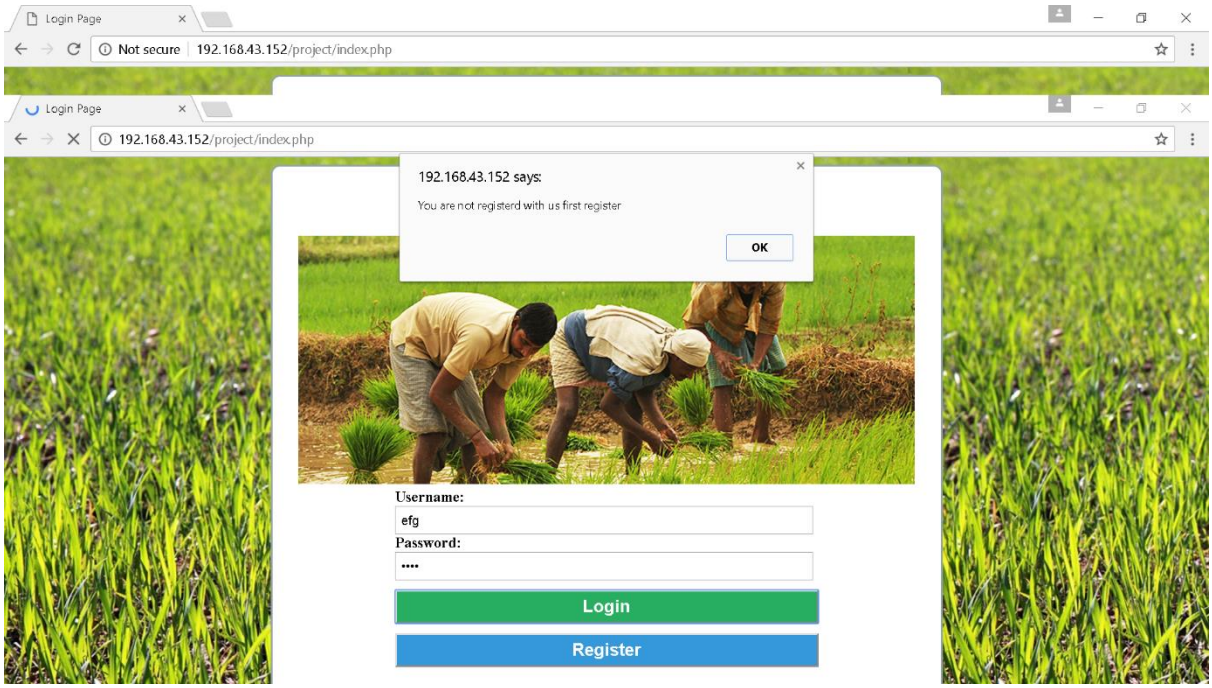
Chapter 8

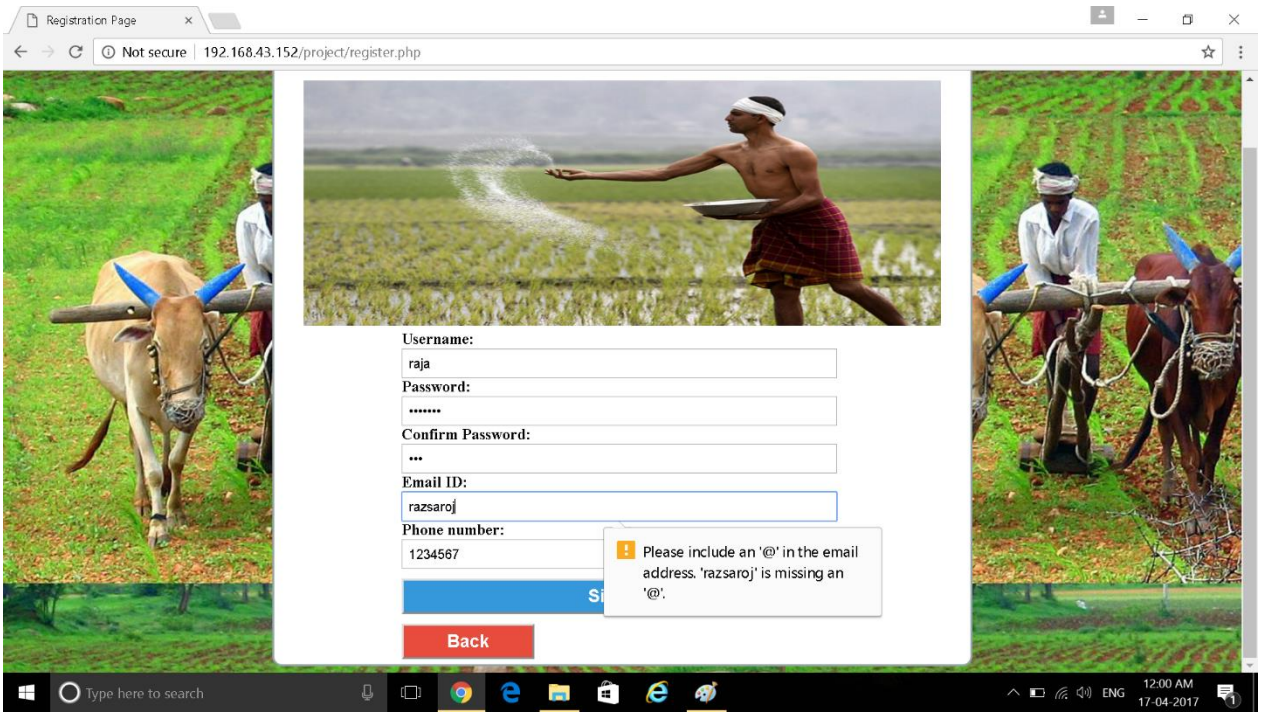
Results

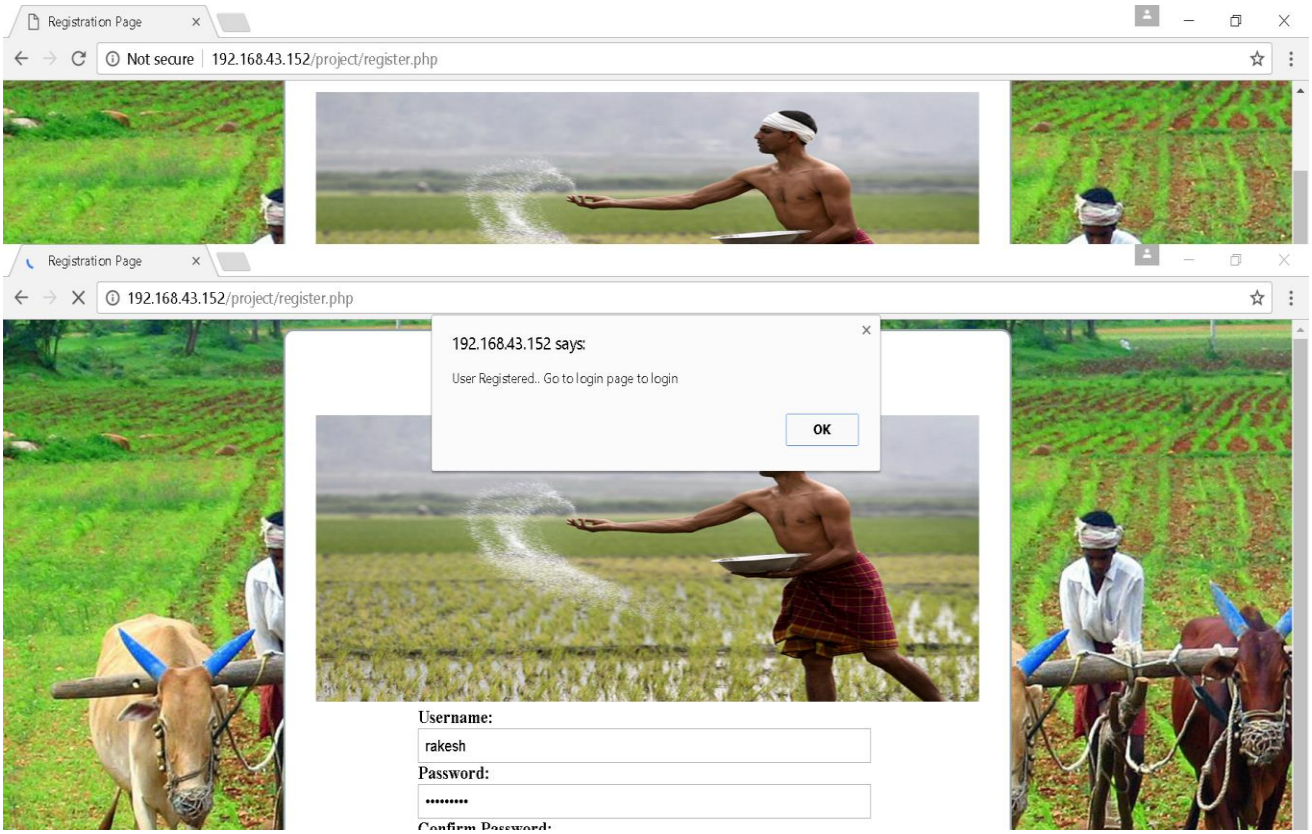
8. Results

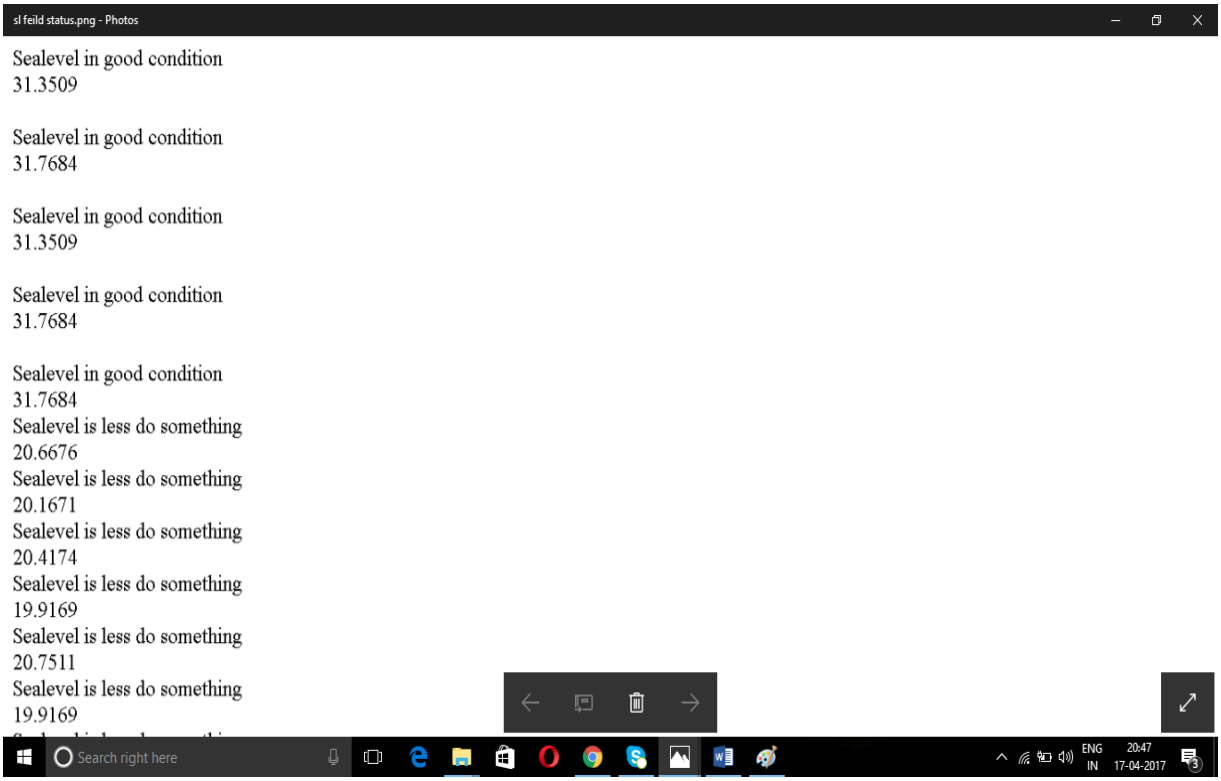
8.1 Graphical User Interface

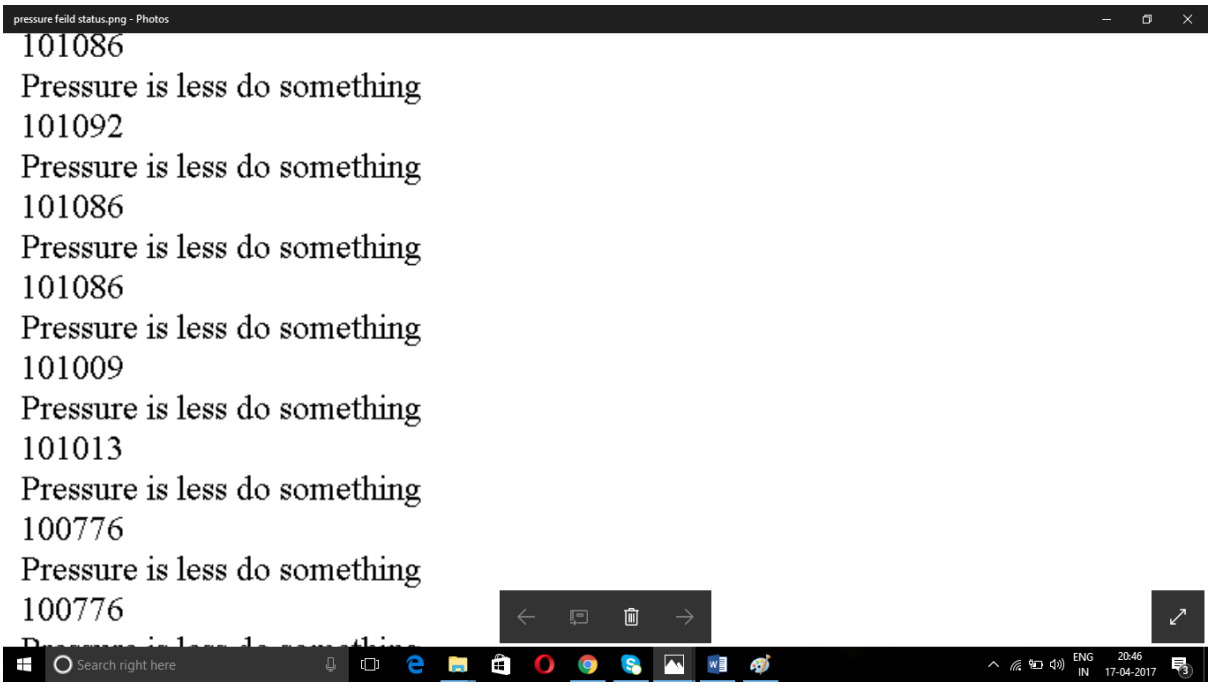


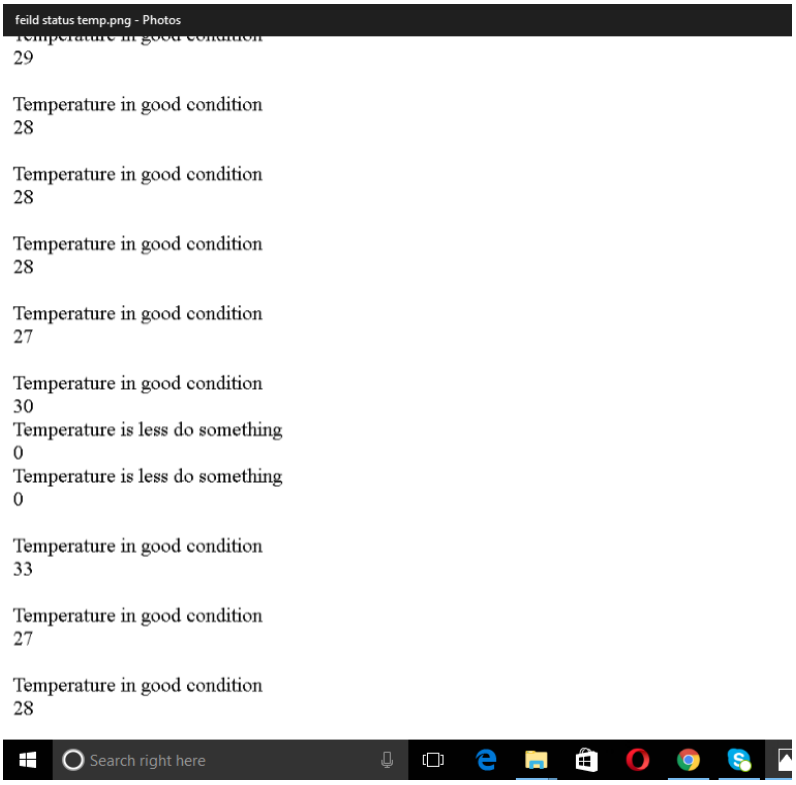


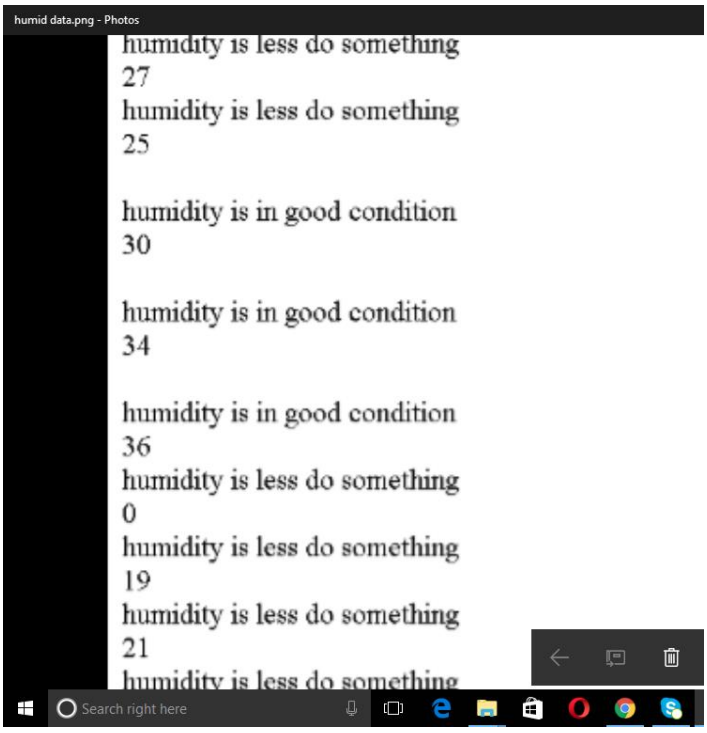

Registration Page
Not secure | 192.168.43.152/project/register.php
Username:
raja
Password:
Confirm Password:
Email ID:
razsaroj
Phone number:
1234567
Please include an '@' in the email address. 'razsaroj' is missing an '@'.
Back
Type here to search
ENG
12:00 AM
17-04-2017

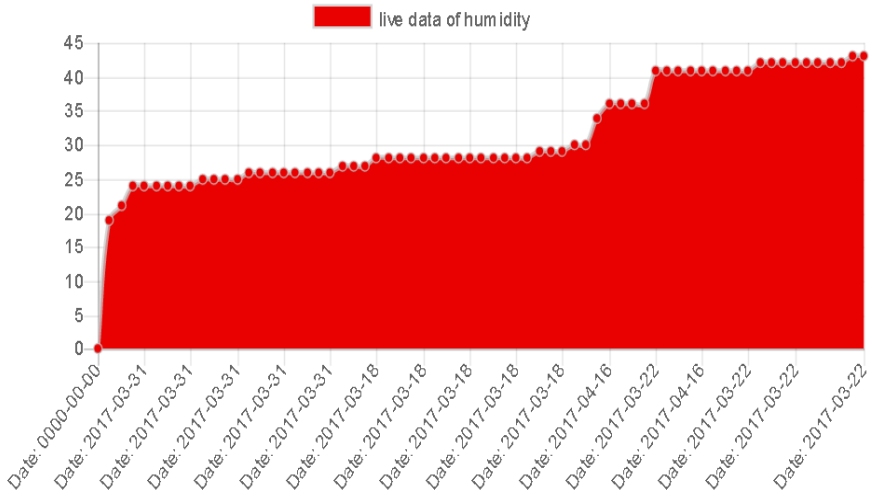

Registration Page
Not secure | 192.168.43.152/project/register.php
192.168.43.152 says:
User Registered.. Go to login page to login
OK
Username:
rakesh
Password:

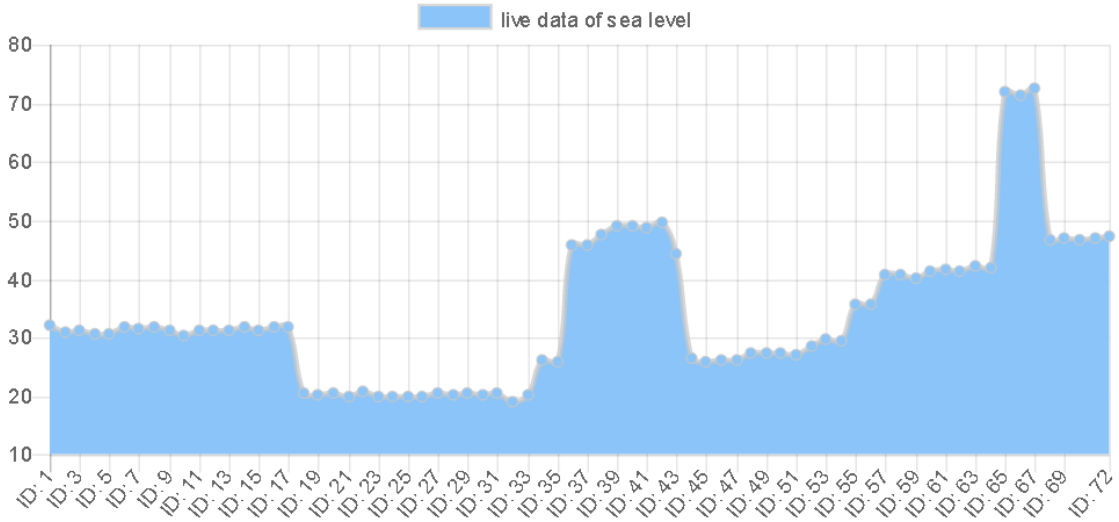

sl feild status.png - Photos
Sealevel in good condition
31.3509
Sealevel in good condition
31.7684
Sealevel in good condition
31.3509
Sealevel in good condition
31.7684
Sealevel in good condition
31.7684
Sealevel is less do something
20.6676
Sealevel is less do something
20.1671
Sealevel is less do something
20.4174
Sealevel is less do something
19.9169
Sealevel is less do something
20.7511
Sealevel is less do something
19.9169
Search right here
ENG
IN
20:47
17-04-2017

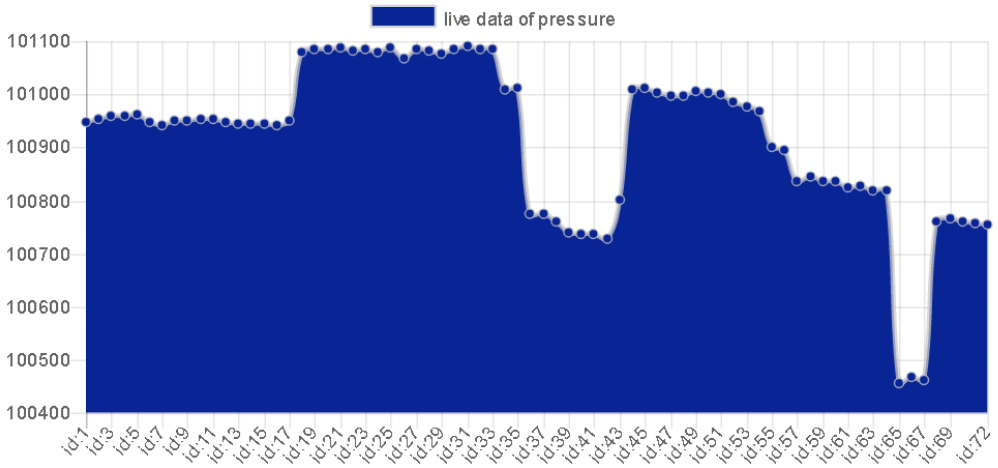

pressure feild status.png - Photos
101086
Pressure is less do something
101092
Pressure is less do something
101086
Pressure is less do something
101086
Pressure is less do something
101009
Pressure is less do something
101013
Pressure is less do something
100776
Pressure is less do something
100776
Search right here
ENG
IN
20:46
17-04-2017

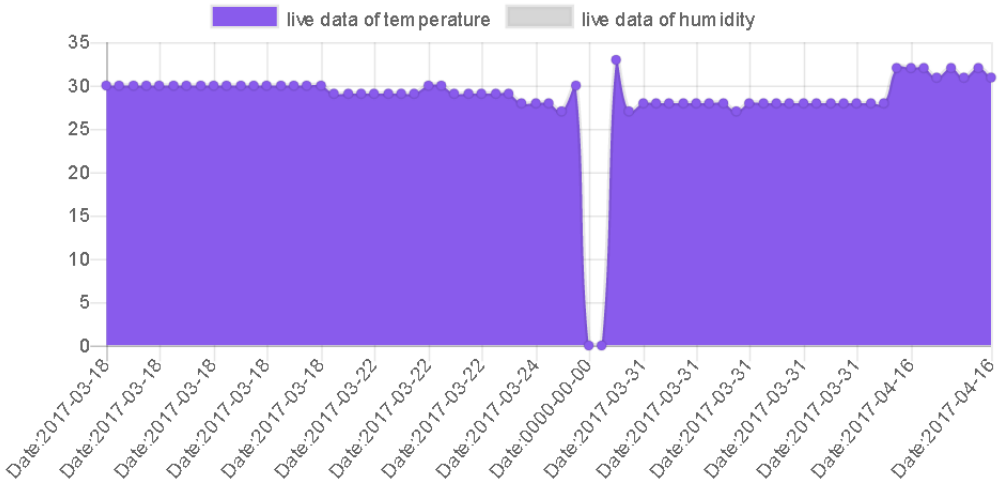

feild status temp.png - Photos
29
Temperature in good condition
28
Temperature in good condition
28
Temperature in good condition
28
Temperature in good condition
27
Temperature in good condition
30
Temperature is less do something
0
Temperature is less do something
0
Temperature in good condition
33
Temperature in good condition
27
Temperature in good condition
28
Search right here


humid data.png - Photos
humidity is less do something
27
humidity is less do something
25
humidity is in good condition
30
humidity is in good condition
34
humidity is in good condition
36
humidity is less do something
0
humidity is less do something
19
humidity is less do something
21
Search right here


live data of humidity
45
40
35
30
25
20
15
10
5
0
Date: 0000-00-00
Date: 2017-03-31
Date: 2017-03-31
Date: 2017-03-31
Date: 2017-03-31
Date: 2017-03-31
Date: 2017-03-31
Date: 2017-03-18
Date: 2017-03-18
Date: 2017-03-18
Date: 2017-03-18
Date: 2017-03-18
Date: 2017-04-16
Date: 2017-03-22
Date: 2017-04-16
Date: 2017-03-22
Date: 2017-03-22
Date: 2017-03-22


live data of sea level
80
70
60
50
40
30
20
10
ID: 1
ID: 3
ID: 5
ID: 7
ID: 9
ID: 11
ID: 13
ID: 15
ID: 17
ID: 19
ID: 21
ID: 23
ID: 25
ID: 27
ID: 29
ID: 31
ID: 33
ID: 35
ID: 37
ID: 39
ID: 41
ID: 43
ID: 45
ID: 47
ID: 49
ID: 51
ID: 53
ID: 55
ID: 57
ID: 59
ID: 61
ID: 63
ID: 65
ID: 67
ID: 69
ID: 72


live data of pressure
101100
101000
100900
100800
100700
100600
100500
100400
id:1
id:3
id:5
id:7
id:9
id:11
id:13
id:15
id:17
id:19
id:21
id:23
id:25
id:27
id:29
id:31
id:33
id:35
id:37
id:39
id:41
id:43
id:45
id:47
id:49
id:51
id:53
id:55
id:57
id:59
id:61
id:63
id:65
id:67
id:69
id:72


live data of temperature
live data of humidity
35
30
25
20
15
10
5
0
Date:2017-03-18
Date:2017-03-18
Date:2017-03-18
Date:2017-03-18
Date:2017-03-18
Date:2017-03-22
Date:2017-03-22
Date:2017-03-22
Date:2017-03-24
Date:0000-00-00
Date:2017-03-31
Date:2017-03-31
Date:2017-03-31
Date:2017-03-31
Date:2017-03-31
Date:2017-04-16
Date:2017-04-16

Chapter 9

Conclusions

9.Conclusion

9.1 Conclusion

Hence, we have implemented a smart device which will make sure precision farming is ensured by taking proper readings from the sensors and displaying the live data to farmers by comparing it to historic database and displaying the data in the form of graph, thereby achieving the aim.

9.2 Future Scope

- This module can be further extended to have support for various other sensors
- Data can be sent to the farmer after data mining and detailed statistics.
- The device can be made mobile.
- Various quality improvements can be done by taking feedbacks.

Bibliography

[1] I. **Auernhammer, Hermann. "Precision** Farming — The Environmental **Challenge."** *Computers and Electronics in Agriculture* 30 (2001)

[2] **Barroquillo, Van. "User Precision Agriculture Interview." Personal interview. 30** Mar. 2013.

[3] **Berry, Joseph. "Technology of Precision Agriculture Email Interview." Interview** Question Via Email

[4] **Berry, J.K., J.A. Delgado, R. Khosia, and F.J. Pierce. "Precision Conservation for Environmental Sustainability."** *Journal of Soil and Water Convervation* 58.6 (2003):

[5] Borgelt**, S.C., J.D. Harrison, K.A. Sudduth, and S.J. Birrell. "Evaluation of GPS For Applications in Precision Agriculture."** *Applied Engineering in Agriculture* 12.6 (1996):

[6] **Brown, Rachael M., Carl Dillon, Jack Schieffer, and Jordan Shockley. "T**he **Impact of Precision Agriculture Techniques on Kentucky Grain Farmers' Carbon Footprint."***The University of Kentucky* (2012): http://ageconsearch.umn.edu/bitstream/119802/2/2012%20SAEA%20RMB.pdf.

[7] https://www.thingworx.com/ecosystem/markets/smart-connected-systems/smart-agriculture/.

[8] http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

Publication:

ISBN NO. : 978-93-5258-745-2

ICEMTE-2017

International Conference on Emanations in Modern Technology & Engineering

March 4th - 5th - 2017

Organized by

Shree L. R. Tiwari College of Engineering

Mira Road (E), Maharashtra

In Association with

University of Mumbai

IETE, Mumbai Centre

Certificate

This is to certify that Prof. / Dr. / Mr. / Ms. / Mrs. SANTOSH SHAH of SLRTCE has participated / Orally presented / Published a paper entitled SMART FARMING TECHNIQUE USING INTERNET OF THINGS in "International Conference on Emanations in Modern Technology and Engineering [ICEMTE-2017]" held on 4th - 5th March 2017 at Shree L. R. Tiwari College of Engineering, Mira Road (E).

In Association with

Publication Partners

Dr. S. Ram Reddy
Convenor & Principal

Shri. Rahul L. Tiwa
Secretary

ISBN NO. : 978-93-5258-745-2

ICEMTE-2017

International Conference on Emanations in Modern Technology & Engineering

March 4th - 5th - 2017

Organized by

Shree L. R. Tiwari College of Engineering

Mira Road (E), Maharashtra

In Association with

University of Mumbai

IETE, Mumbai Centre

Certificate

Prof. / Dr. / Mr. / Ms. / Mrs. ARUN SHARMA of SLRTCE has participated / Orally presented / Published a SMART FARMING TECHNIQUE USING INTERNET OF THINGS "International Conference on Emanations in Modern Technology and Engineering [ICEMTE-2017]" held on 4th - 5th March 2017 at Shree L. R. Tiwari College of Engineering, Mira Road (E).

In Association with

Publication Partners

Dr. S. Ram Reddy
Convenor & Principal

Shri. Rahul L. Tiwari
Secretary