Contents

[Project Objectives 2](#_Toc26661577)

[System architecture design 2](#_Toc26661578)

[Hardware Requirements 2](#_Toc26661579)

[Operating System chosen 6](#_Toc26661580)

[Programming Languages and Database 6](#_Toc26661581)

[**System Implementation** 8](#_Toc26661582)

[9.1. Screenshots of the final system 8](#_Toc26661583)

[9.1.1. Screenshot for home page 8](#_Toc26661584)

[9.1.2. Screenshot for login page 8](#_Toc26661585)

[9.1.3. Screenshot for real-time control 10](#_Toc26661586)

[9.1.4. Screenshot for adding new schedule 11](#_Toc26661587)

[9.1.5. Screenshot for view schedule 11](#_Toc26661588)

[9.1.6. Screenshot for edit schedule 13](#_Toc26661589)

[9.1.7. Screenshot for delete schedule 13](#_Toc26661590)

[9.1.8. Screenshot for add new user 15](#_Toc26661591)

[9.1.9. Screenshot for view user 15](#_Toc26661592)

[9.1.10. Screenshot for the normal user home page 17](#_Toc26661593)

[9.2. Screenshots of the source code 17](#_Toc26661594)

[9.2.1. Explanation of the sample code for adding new schedule 18](#_Toc26661595)

[9.2.2. Explanation of the sample code for turn on/off lights based on scheduling 19](#_Toc26661596)

[9.2.3. Explanation of the sample code for Real-time control 20](#_Toc26661597)

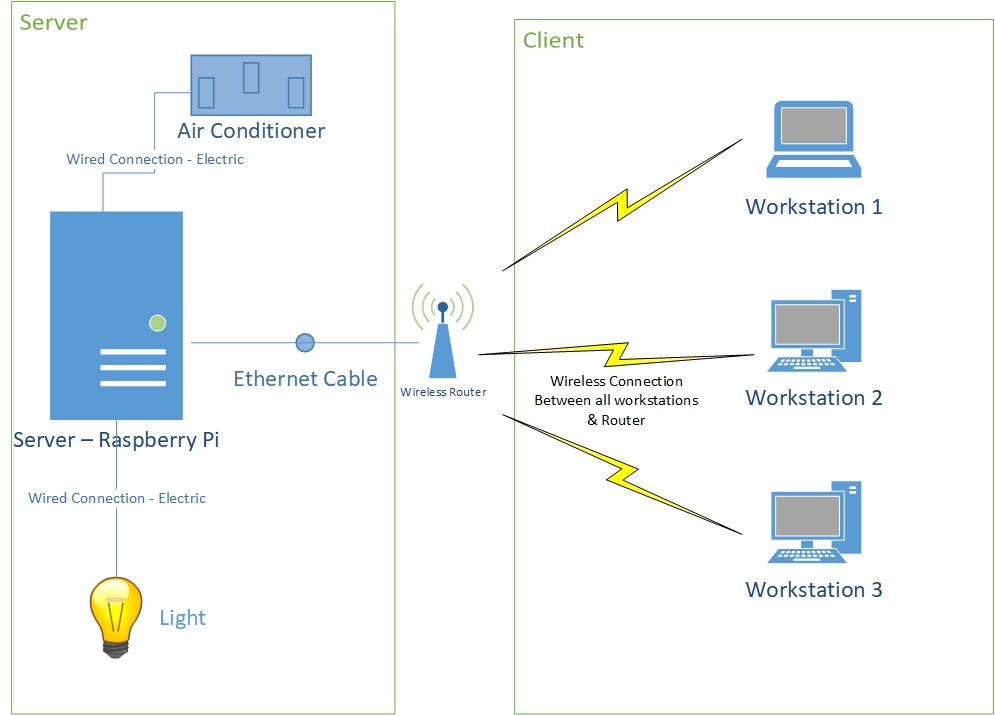
[9.2.4. Explanation of the sample code for turning on pin 21](#_Toc26661598)

[9.2.5. Explanation of the sample code for turning off pin 21](#_Toc26661599)

## Project Objectives

1. To provide an optimized Web application system which allows to remotely control the power of “air-conditioners and lights” for each classroom.
2. To provide an intelligent Web application system which can automatically turn ON/OFF the power of “air-conditioners and lights” for each classroom.
3. To provide a Web application system that can automatically control classrooms’ temperature environment “air-conditioning”.
4. To provide an optimized system to assist for reducing the consumption of electrical power and save money in any educational Institution.

## System architecture design



## Hardware Requirements

The minimum requirements for hardware for this project to be successful to carry out and meet the objectives of the system are as follows:

PC

* Processor – Core i5
* Random Access Memory (RAM) – 512MB
* Keyboard & Mouse
* Other components

:

* Router Wireless (Wi-Fi)

A router is a small electronic device which allows to join multiple networks deceives together via using wired or wireless connections. The wireless Router will be used to connect to the Raspberry pi via using an ethernet cable in order to setup network TCP/IP network for the server side. The wireless router will also provide a Wireless LAN connectivity, which allows the users to connect as clients from their PCs to the Server which is installed on the Raspberry Pi thus they can remotely browse to the system’s web page.

* 330-ohm resistors

A resistor is a small electrical component which is designed to regulate the flow of the electrical current in any electronic circuit (Rouse, 2017). Resistors must be used to connect the LED indicators to the GPIO pins of the Raspberry Pi. The fact that Pi can only supply small current, whereas LEDs want to draw more. As a result, resistors have to be used to ensure that the Pi will not be damaged (The Pi Hut., 2017).



Figure 9: 330-ohm resistors - Source (JIMB0, n.d.)

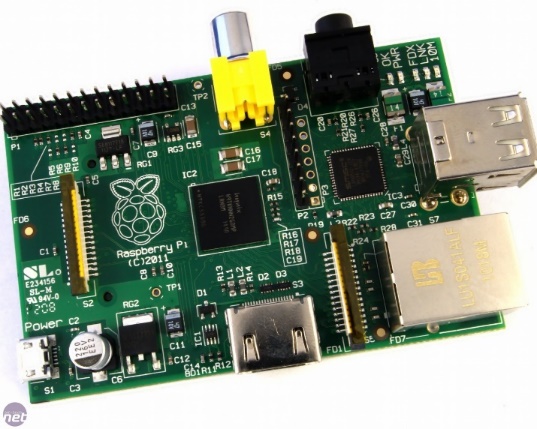
* Ethernet cable

It is a wire cable which will be used for connecting the PI server to the router.



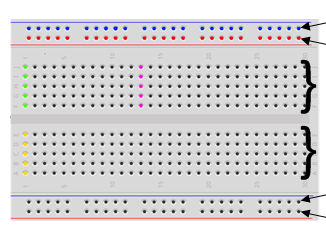
* A microcontroller such [Raspberry Pi Model B](https://www.allaboutcircuits.com/electronic-components/?p=raspberry+pi+model+b) with memory card

The Raspberry Pi which is also known as Pi, it is a very small computer device which is popularly known as an alternative to the Arduino, BASIC Stamp, and various models of microcontrollers. In fact, Pi is much more than a microcontroller, it is a full-blown computer system, implemented on a single small card. In addition, Pi is supported with the common features which can be found on normal desktops or laptop devices. Nevertheless, it has a small size with other extra features which can’t be found on devices such as laptops or desktops. Pi has the ability to directly control digital I/O pins. As a result, it can be used to be integrated with other external devices such as LEDs, potentiometers, various types of sensors and others more (Lowe, n.d.).



* Breadboard

It is a board which can be used for connecting electronic devices to each other without needing to solder them together. This board will be used to connect Resistors to LEDs then to GPIO pins of the Pi (The Pi Hut., 2017).



* LED Indicators

LED indicators will be used as a prototype for representing the operation of the actual lighting and air-conditioning system of classrooms. The LED indicators will be connected to the breadboard as well as the resistors which will be connected to the Raspberry Pi’s GPIO Pins via using jumper wires.



* Male and female wires

It is also known as jumper cable or DuPont cable which is an [electrical wire](https://en.wikipedia.org/wiki/Electrical_wire) which is provided with connectors and pins at each end. Jumper cables are usually used to interconnect the components of a [breadboard](https://en.wikipedia.org/wiki/Breadboard)  with other equipment or components. In this project, Jumper wires will be used for connecting the GPIO Pins of the Pi to the breadboard, which will assist in controlling the LED indicators.



## 

## Operating System chosen

* Linux Rasbian.

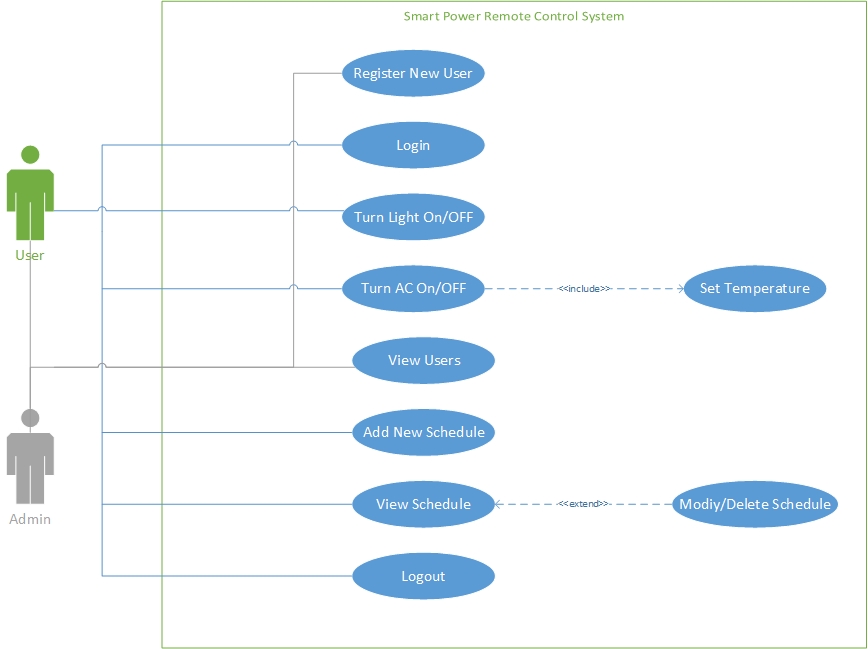
It is the foundation’s official supported operating system and it is a free OS based on Debian optimized for the Raspberry Pi hardware (Raspberrypi.org, 2017). This operating system will be installed on the server side on the Raspberry pi.

## Programming Languages and Database

* **PHP**
* **CSS**
* **HTML**
* **Python**
* **MySQL**

#### 

#### System Use Case Diagram



# **System Implementation**

## Screenshots of the final system

This section of the report will elaborate of the final user interface and its detailed explanation. It can help the user to under the flow of each segment of the system. The user interface itself provides an ease to all the user, but this is an alternate way to learn more about each component of the screen.

### Screenshot for home page

#### Screenshot

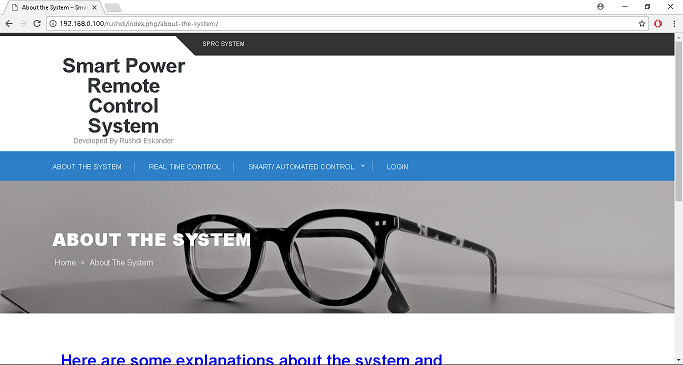


Figure 39: Screenshot for home page

#### Description

The above-mentioned figure illustrates the very starting appearance of the web application where it displays main functionalities in the navigation bar and a bit of details of the Smart Power Remote Control System. Home page will only allow the user to know about the system or simply login into system, whereas the rest of the menu items which are the core functionalities will be not allowed before login and they are all disabled, it will not work until the user logs in, hence it shows a restriction message to the user instead where it ask the user to login or ask the administrator to grant you the access.

### Screenshot for login page

#### Screenshot

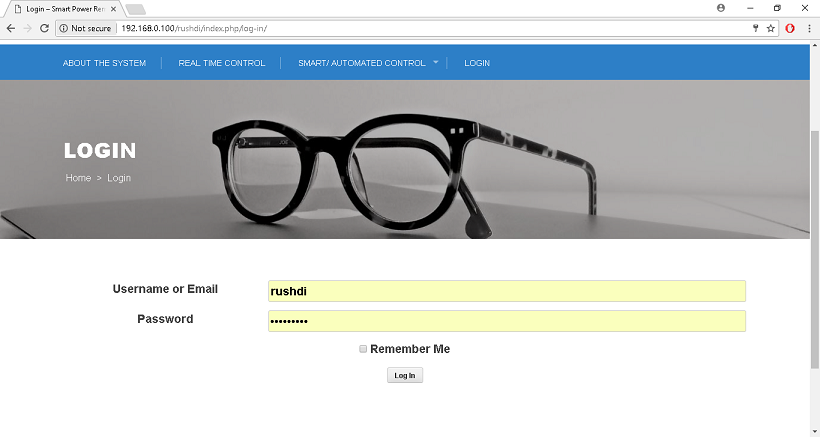


Figure 40: Screenshot for login page

#### Description

The figure above illustrates the login page of the system where it clearly displays the two fields where are required by the user. Hence the login only works once the credentials are valid, if not then an error message appears. The interface is easy to understand the user can input its username or email and password.

### Screenshot for real-time control

#### Screenshot

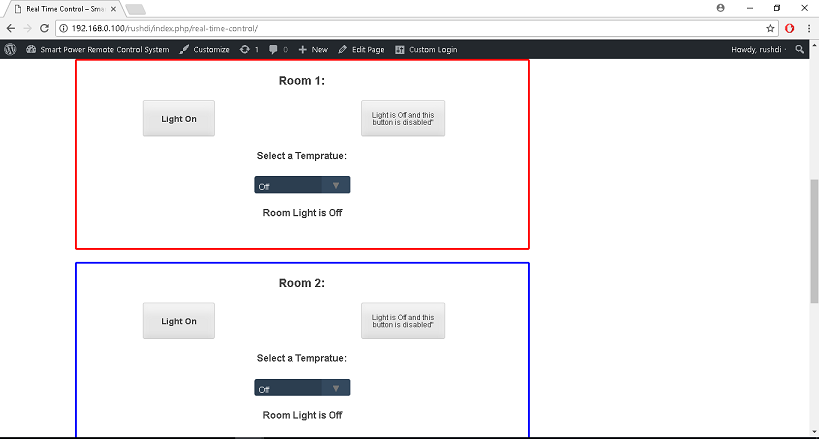


Figure 41: Screenshot for real-time control

#### Description

The figure displays an essential interface of the web application which is the core function of the proposed system. Real-time control facility is available for both normal users and administrator. It is clear to understand that the screen appears with the control of light and AC for each classroom which is very user friendly. In simple words, any user can use this control as it is very clear to understand about its workings. This screen shows the light on and light off button and both work accordingly. There is one dropdown list by the temperature of air conditioner can be set and by default its off. Under the AC function the status bar appears which show the status of lights and AC as well.

### Screenshot for adding new schedule

#### Screenshot

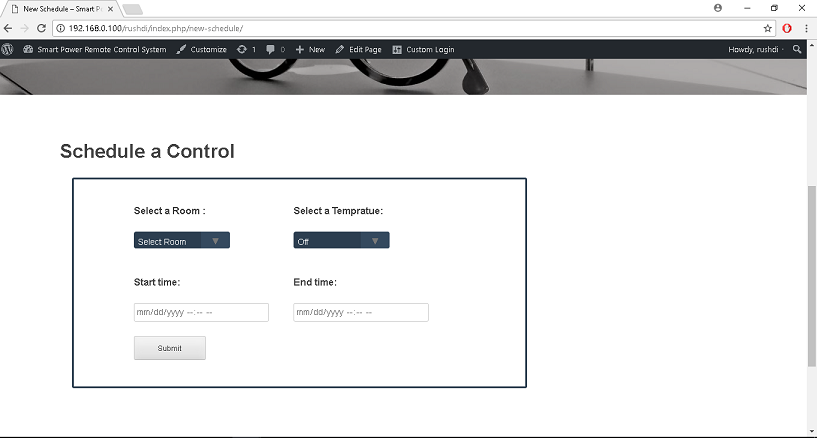


Figure 42: Screenshot for adding new schedule

#### Description

The figure illustrates the functionalities of the new schedule which appears with two different panels side by side, first shows the selection of room and the other allows the user to select the temperature of the AC if required, respectively. Then comes the start time function that allows the user to set the date and time when it is supposed to operate and next to it is the end date time setting which will turn off the lights and AC according to the end time. Once the submit button is clicked the record will be stored in the database and the operation will take place accordingly.

### Screenshot for view schedule

#### Screenshot

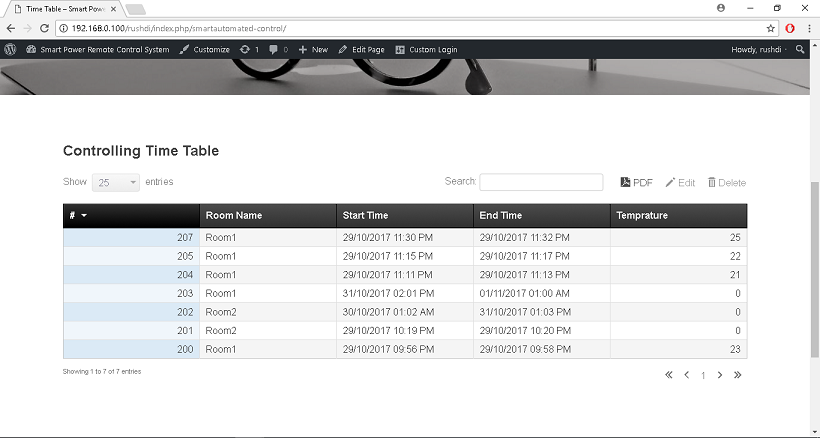


Figure 43: Screenshot for view schedule

#### Description

The figure displays the screen of the viewing schedules set by the users. Hence it allows to view the schedule as well as there are options at the left top corner that allows to edit, delete or print. At the bottom of schedule table, it is obvious to understand that the pagination facility is being provided so that scroll can handle large number of records at once. Inside the schedule table the user can clearly notify the details of each classroom that are being set.

### Screenshot for edit schedule

#### Screenshot

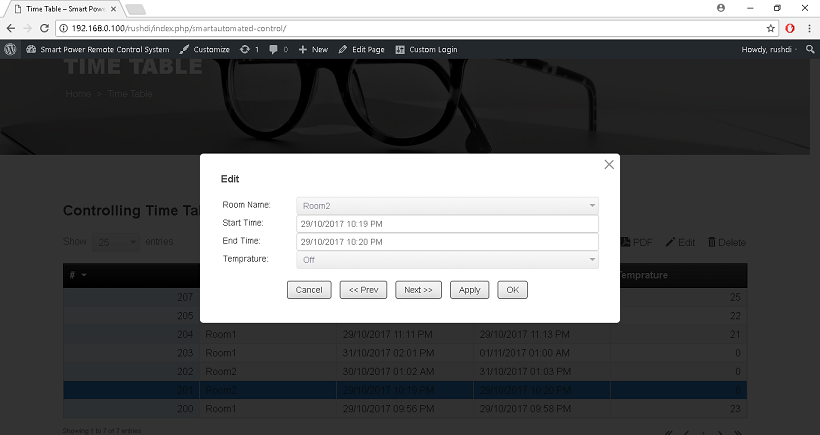


Figure 44: Screenshot for edit schedule

#### Description

The figure above mentions the modification of the schedule within the system, only that record will be modified that is selected. To modify any record, the user must select the records and click edit above to modify hence the screen appears as popover panel where it allows the user to modify that particular record. Once the changed are being made and the user should simply click the button below, the record will be updated, and the operations will be now performed accordingly, and the schedule will also show the updated record.

### Screenshot for delete schedule

#### Screenshot

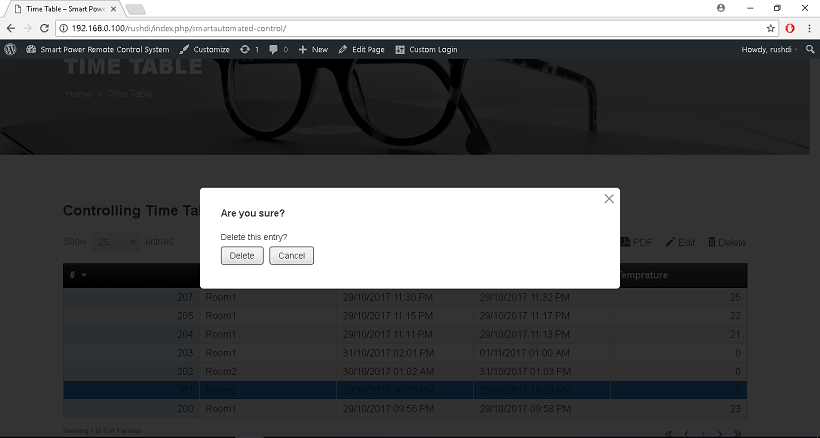


Figure 45: Screenshot for delete schedule

#### Description

The figure above shows the screen to delete the schedule record from the database. The workflows need the user to select any of the record that is to be deleted, after selecting the record simply click delete, the system will ask for the confirmation. If the confirmation is positive the records will be deleted, and it will not be displayed anymore in the schedule.

### Screenshot for add new user

#### Screenshot

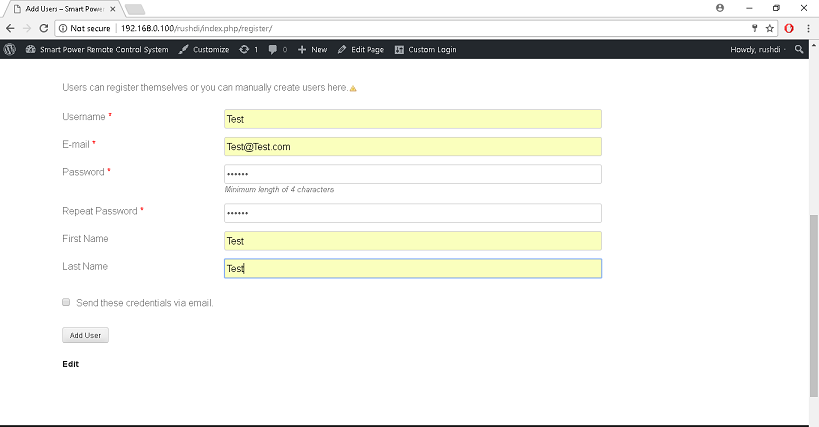


Figure 46: Screenshot for add new user

#### Description

The figure above illustrates the options to add new user, yet it is mentioned in the navigation bar, but it cannot be accessed by all users, only the admin can create the user. The fields that appear in this page are all highly validated and each of them are required as it reduces the chance of errors. Once of the inputs are valid, click register button to create this user. Once the user is created, it can be logged in, but that user will not have the access to the users’ menu.

### Screenshot for view user

#### Screenshot

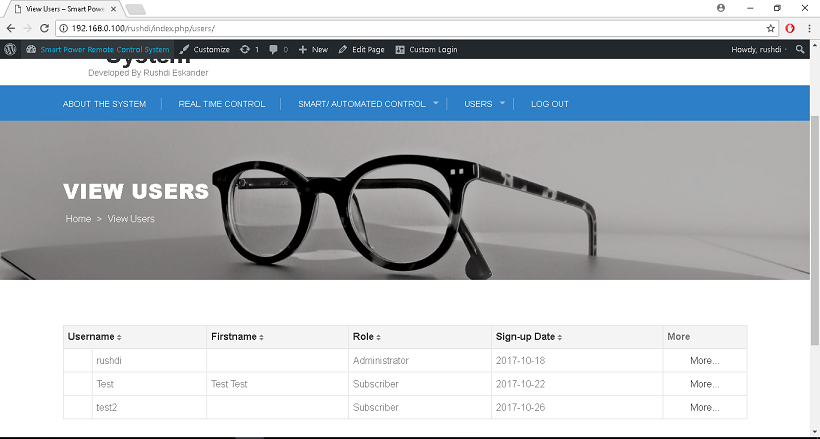


Figure 47: Screenshot for view user

#### Description

The figure above shows that screen to view the users, it can be accessed through the menu item as Users whereas it directly navigates the administrator to view the users. However, this option is also only available for the admin, hence no other user can use this, the restriction message is displayed instead by accessed by other users.

### Screenshot for the normal user home page

#### Screenshot

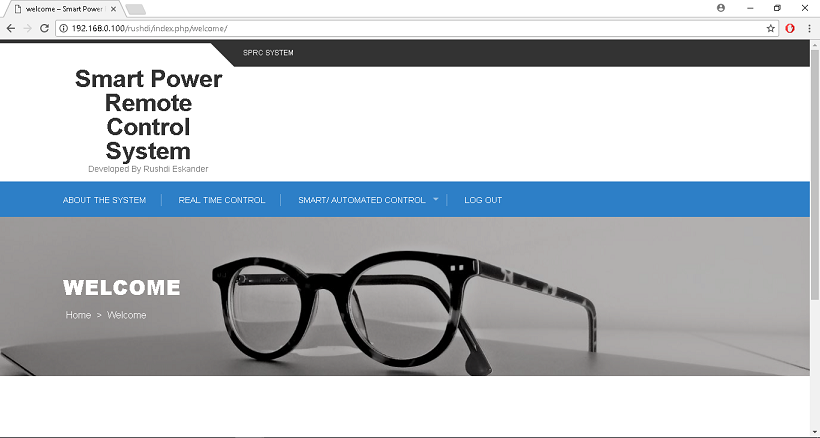


Figure 48: Screenshot for the normal user home page

#### Description

The figure above displays home page for the normal users who have been added by the main admin. Normal users can use all the system’s functionalities same as main admin; however, they can’t add or view new registered users. The fact that, adding or view new registered users’ menu items will be invisible in normal user home page.

## Screenshots of the source code

Source code is main backend workflow which is a guide for the user to understand the sequence of the programming in this system. This explanation and screenshots can help the reader to know more about source code and programming as different developers have logics. The developer has decided to choose few sample codes from the system itself, which will be elaborated in detail below.

### Explanation of the sample code for adding new schedule

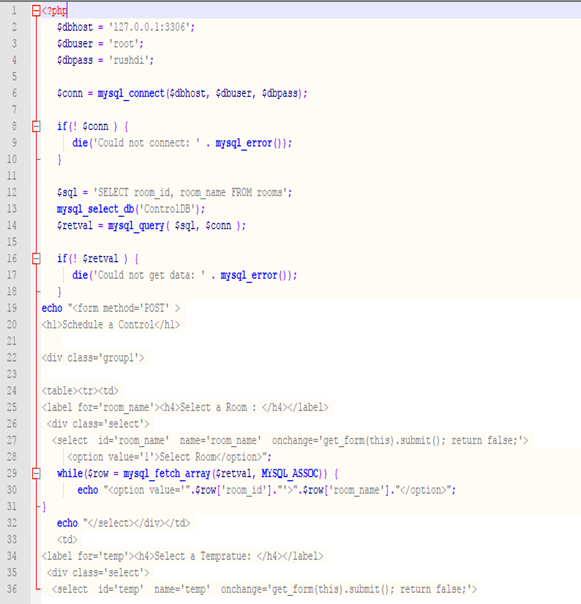


Figure 49: Sample Code - Adding New Schedule

In the figure above, the developer has written the code to establish a connection to the MySQL database and make sure there is no errors in the connection. The developer then passed a query for the room’s name from the room’s table (entity). The code flows directly pass the rooms query to the entity itself, then getting all rows values and populating all rooms’ names and IDs in the dropdown list to be viewed to the user in HTML standard.

### Explanation of the sample code for turn on/off lights based on scheduling



Figure 50 Sample Code - Checking Timetable Control

In this code, the system sends a query to check the timetable row by row. The developer queries for every row and save the room name, python light on and off commands in variables. At each row, the system will compare the start date and time to the current date and time, if it is true then lights on for the current room. Otherwise, it will compare the end date and time to the current date and time, if it is true, then lights off for the current room and so on for every row in the timetable. Finally, closing the connection to the MySQL database.

### Explanation of the sample code for Real-time control

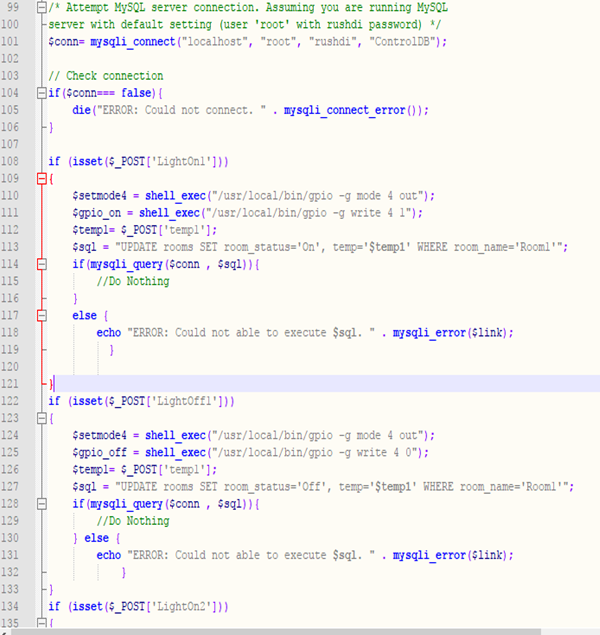


Figure 51: sample code - Real-time control

The code above shows the code of a connection establishment to MySQL database and check if the connection is done without returning any error. The rest of the code works for comparing the posted the values from the previous button press action in the same page. After getting the true results from the comparison, it will perform a shell execution to the Raspberry Pi to a chosen input to be light on or light off. Once the shell execution is done, it will perform an update for the temp and status of the light in table rooms for the selected room.

### Explanation of the sample code for turning on pin

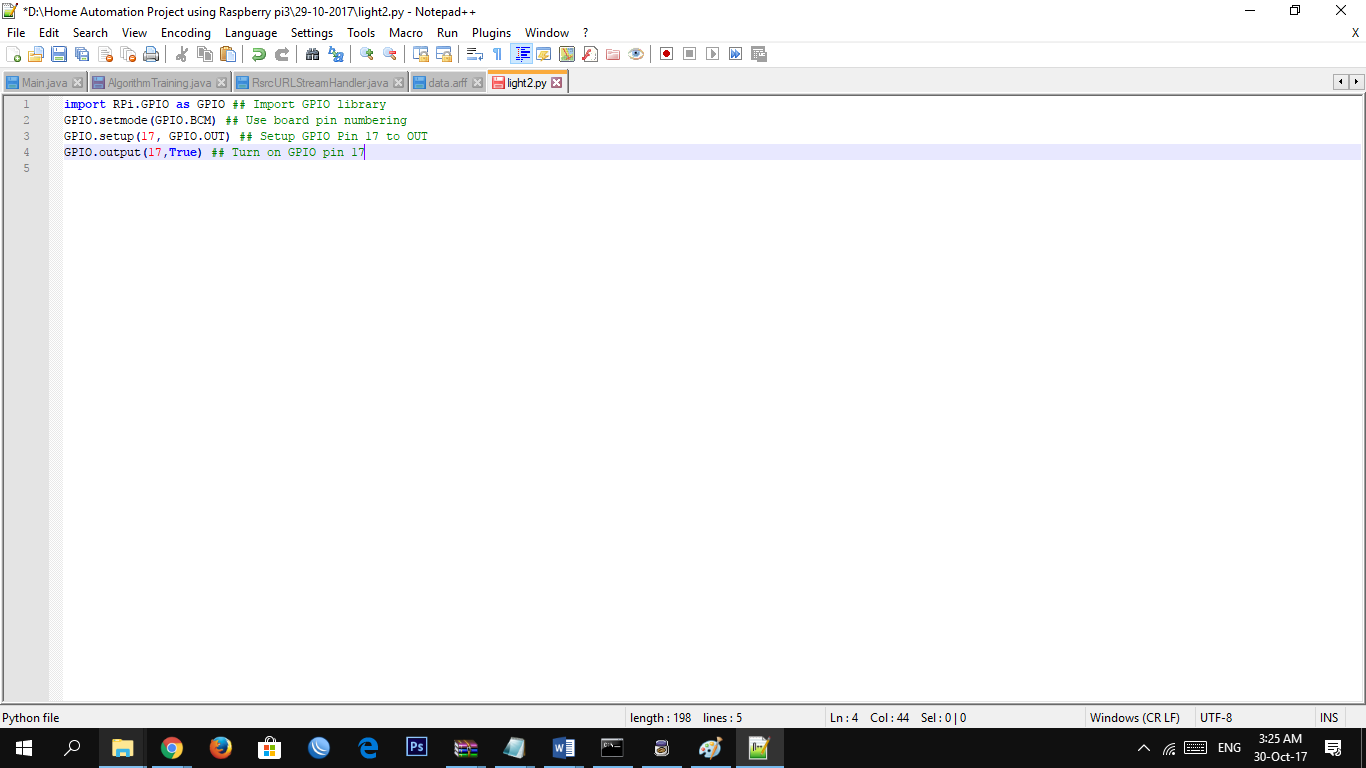


Figure 52: Sample Code - Turning On Pin

The above figure shows the python code that performs a turn on to pin number 17 as shown. First, the developer imports a raspberry pi GPIO library to recognize raspberry pi commands. After importing library, it sets the mode to BCM that allow sends commands to the Raspberry Pi. Thus, the Raspberry Pi will be ready to receive commands which the developer setup and specify the pin number by the command OUT, then he sends a true Boolean value to open and turn on the pin for power.

### Explanation of the sample code for turning off pin

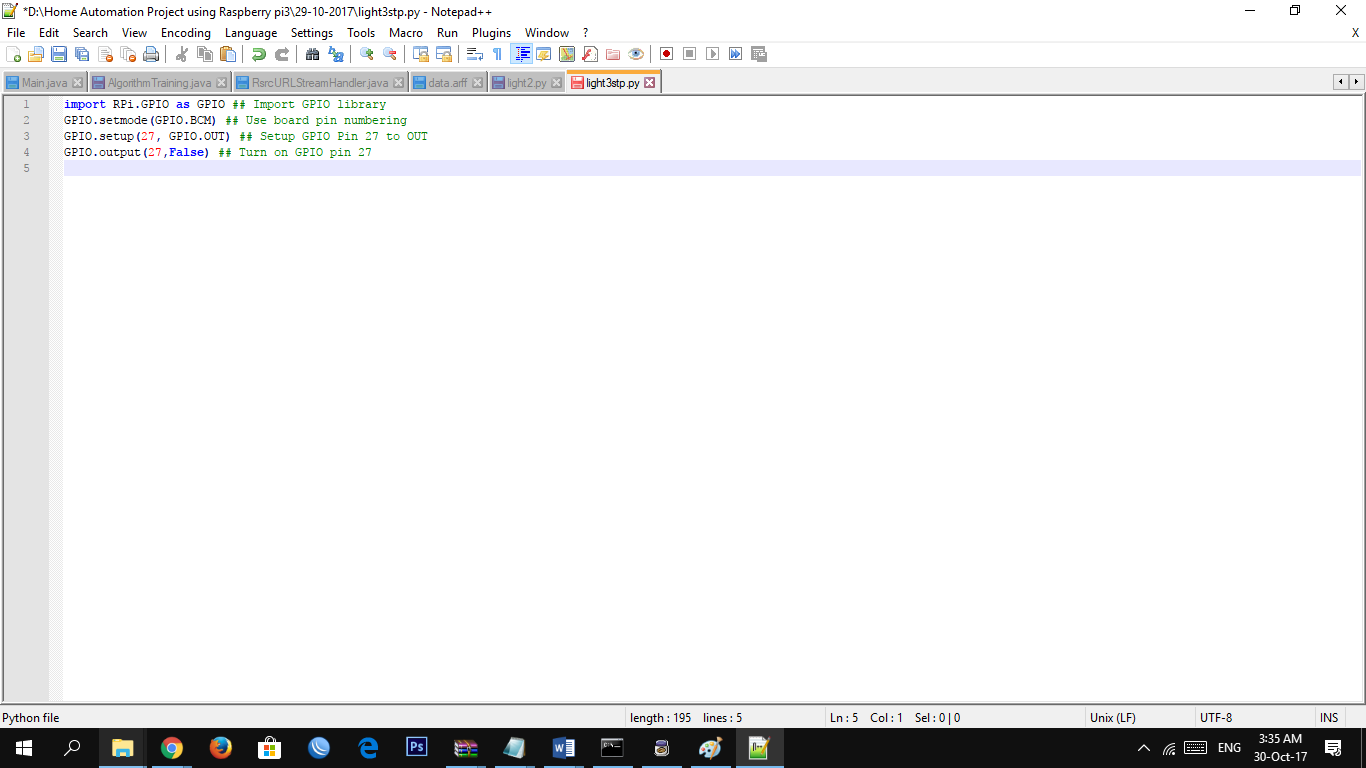


Figure 53: Sample Code - Turning Off Pin

The above python code is for the turning off command for pin number 27. It is the same for the previous turn on python code, except that the developer in this code changed the pin number to control pin number 27, then he sends a false Boolean value to the specific pin to tell the raspberry pi to turn on the power from pin number 27.