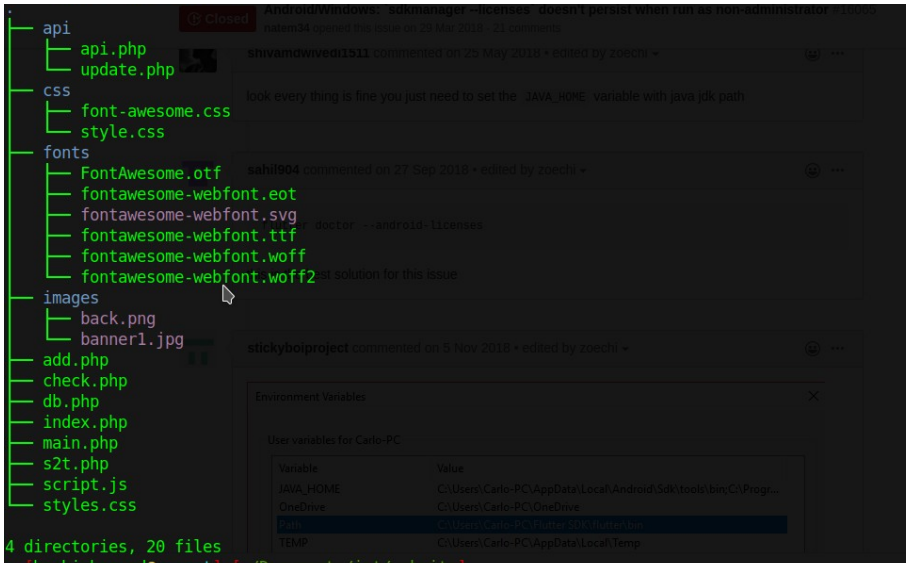
IOT BASED HOME AUTOMATION WITH VOICE RECOGNITION

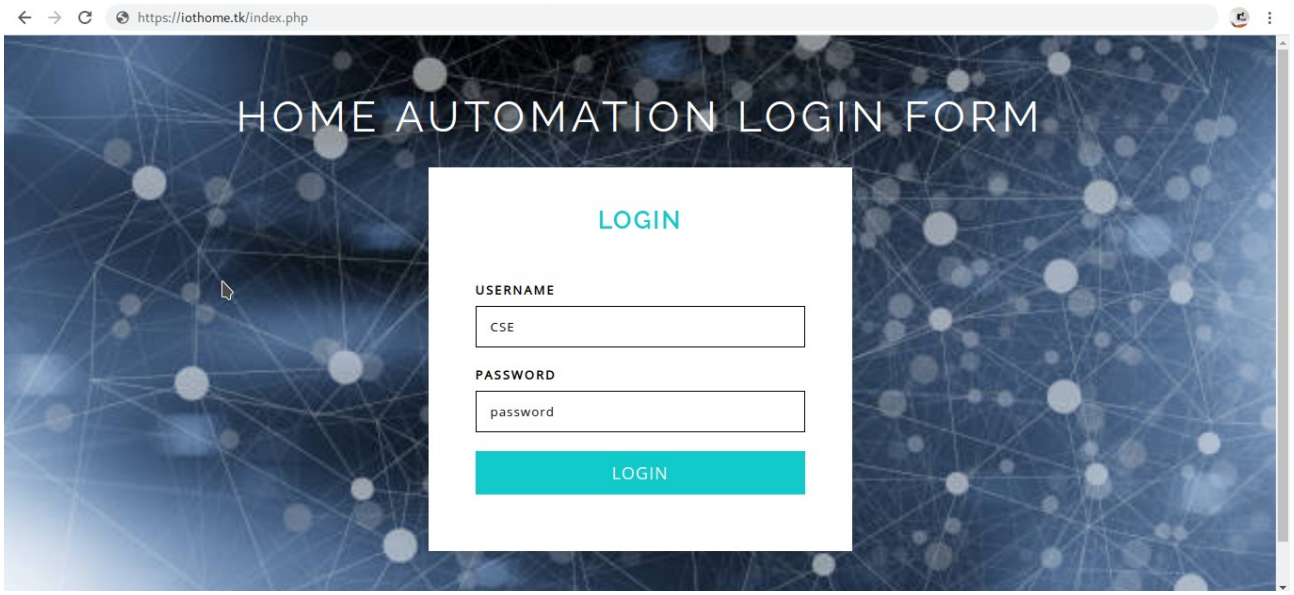
Website / Application:




WEBSITE URL: iothome.tk

>> index.php – Landing page of the website

This page contains a simple form to give username and password and action="" to same page. Contain php code to check the given credentials are correct. If its corect it redirects to check.php page

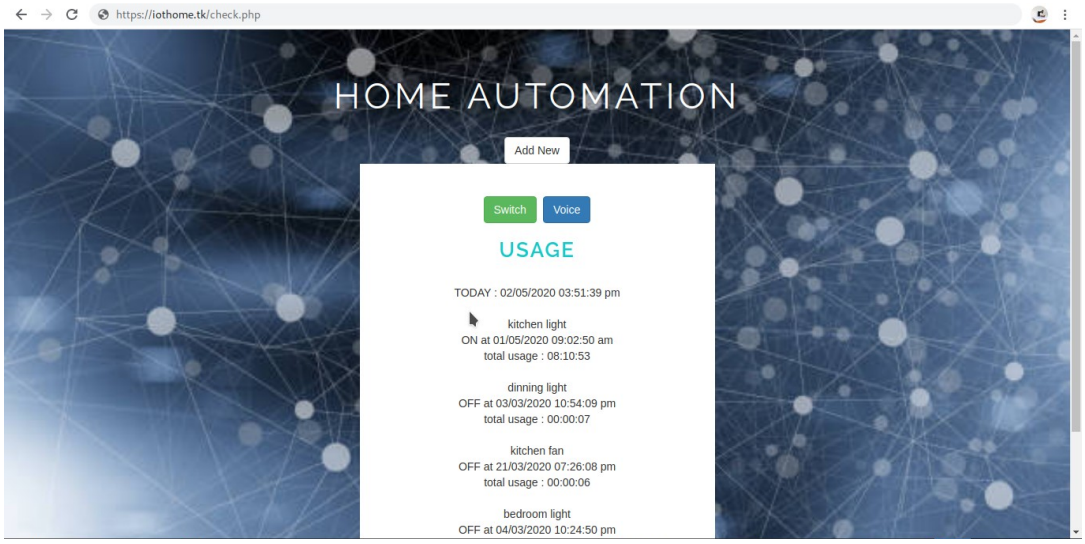



USERNAME : CSE
PASSWORD : password

>> check.php – once you logged in .. you will reach this page..

Here you can view the status of all the switches connected to controller..
format- device name, On/Off , last On/Off time-stamp , total usage
this page has 3 buttons / link – Add new , switch , voice

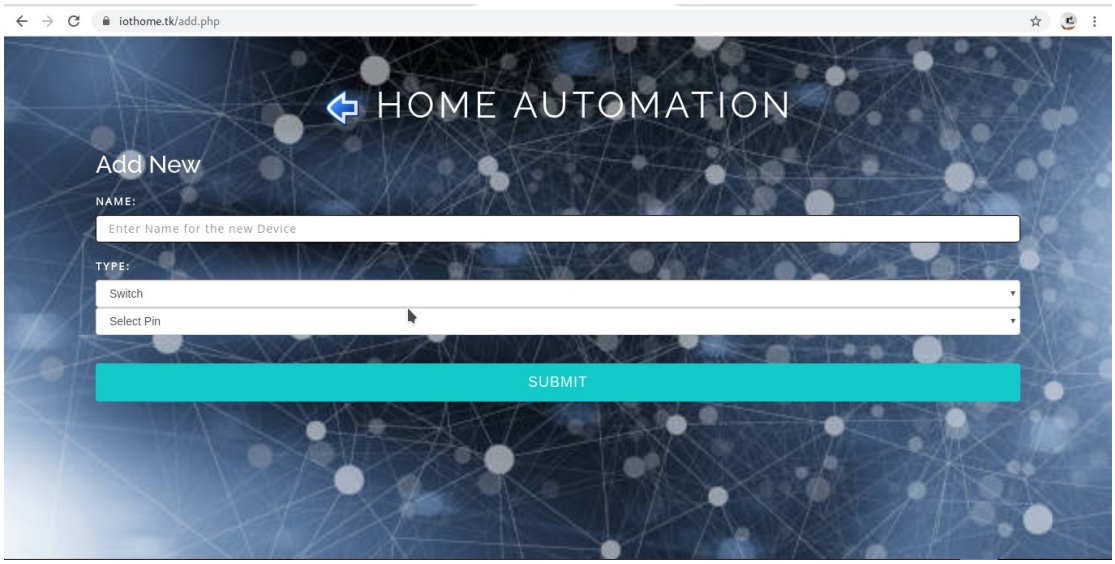
This select all the data from database such as name,status,time,totalusage and display in the page




>> add.php – Once you click “Add new” button from check.php, you will reach this page

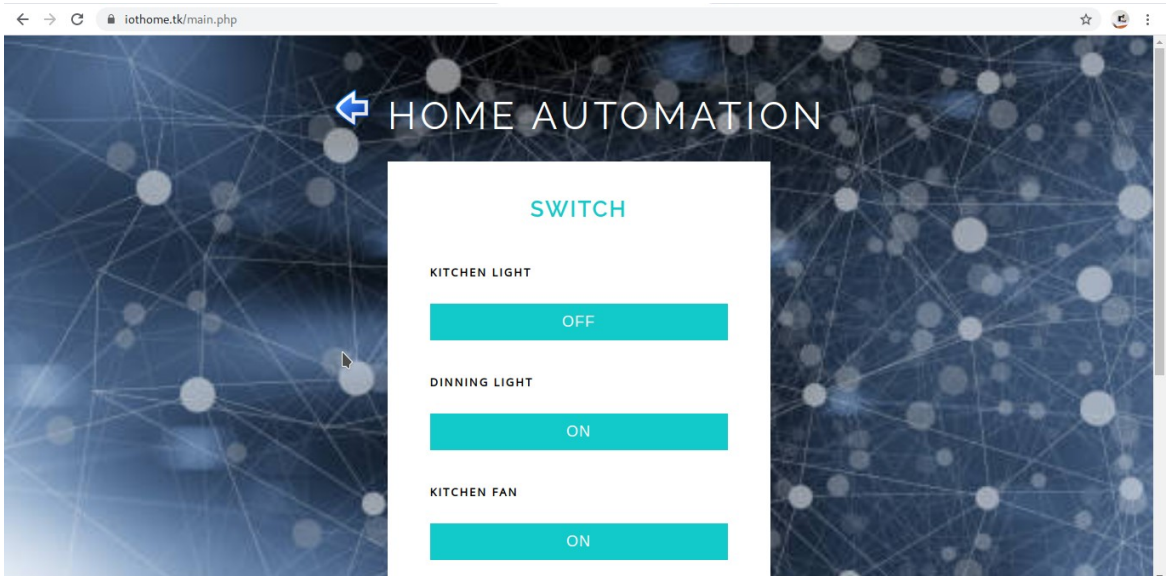
Here you can add new device to the device.. by giving details –
name of the device, device type(switch/regulator), pin value(gpio pin number)

This page contain a simple form and update the data in database



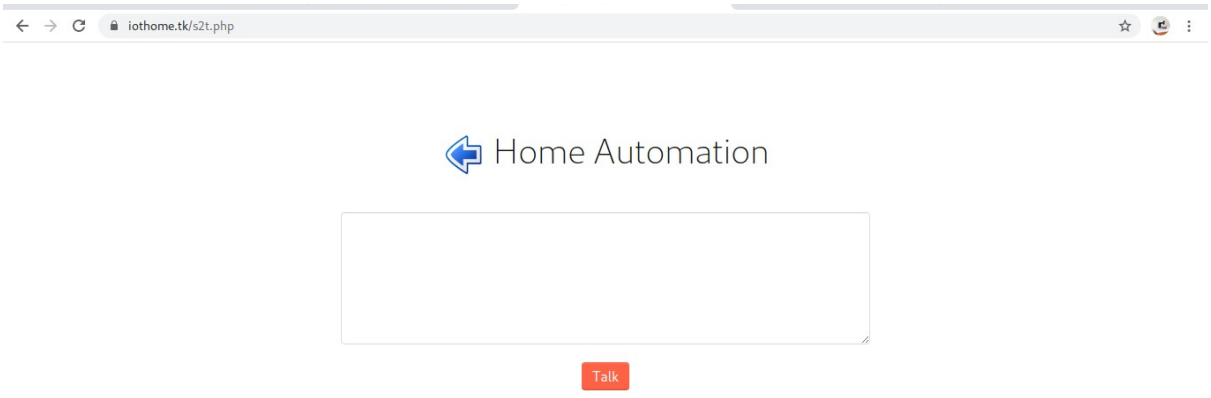
>>main.php – Once you click “switch” button from check.php you will reach this page
Here you can see the switches of each devices. You can On/Off the device

This page contains button.. based on the button value and pin number.. this updates the status of the button in database




>> s2t.php – Once you click “voice” button from check.php you will reach this page.
Here you can give the voice command by clicking the “Talk” button
command will be “<device name> on” / “<device name> off”
example : “kitchen light on” .. it will reply success
if you given wrong command.. it will say no command found

This page include the script.js file which process the command..



>> script.js – The voice command from s2t.php file will be send to this page and here the voice will be converted to text, and this call the api- update.php by passing the voice command. The responds from the api will be either “success” or “no command found”. This response will be converted to voice in the same page.

>> css folder – contains the css files regarding the design of the webpages.

>> fonts folder – contains font related scripts files

>> images folder – contains images used in webpages , background image and icon to goback

>> api folder – contains the api files

>>api.php – Its an get api which returns the pin number and status of each device from database to the requested page.
This api is used by raspberry pi.. (final.py)

>>update.php – its an post api - which update the status of the device in database based on command given. This api accept the command and check the device name from database, and if the command contains on- it updates status of that device as on in database else it update off in database.
If the update happens, it is an valid command, thus it returns “Success”.
Else it returns “No Command Found”
This api is used in raspberry pi (s2t.py) and website (script.js)

**Database :**

>> Consist of one table “new” which has 8 colomns

Table Structure

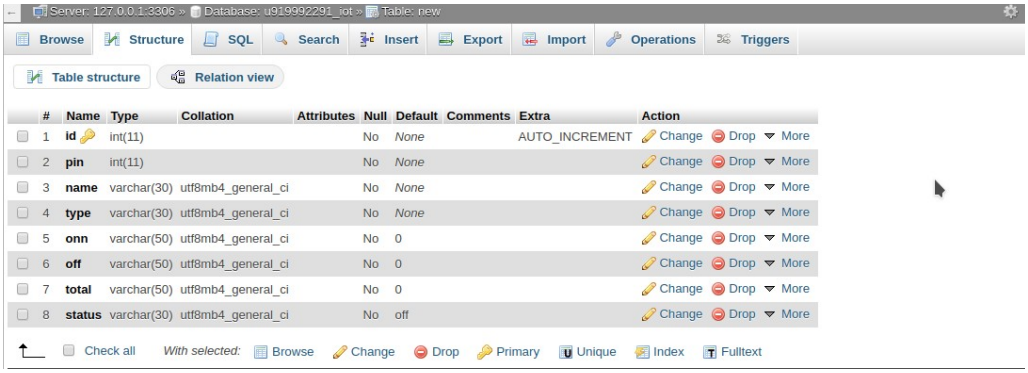
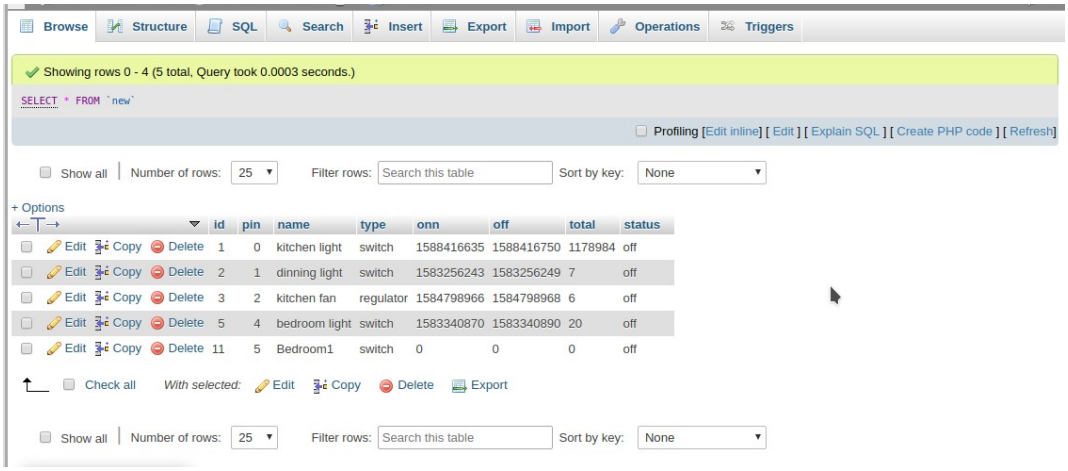



Table Sample Data




the colomn “onn” / “off” stores the timestamp value which will be converted to dd/mm/yyyy h:m:s format in frontend

“total” stores the total usage of the device.. which is calculated by once the device becomes off as total=total+(off-on)

Raspberry pi contains 2 files running in the device continuesly

>> rasb.py – this python file execute and continusly calling the api – “api.php” and based on the response data of each pin and status, the gpio pins of raspberry pi become enable or disable.

>>s2t.py – this python file also executing and continuesly record the commands from out side.
This file mainly convert the voice command to text using google api feature,
and do a post request to api – update.php with parameter -command
the response from the api will be converted to voice and played.

we hav given the device name as “siri” thus it activates once we call “hey siri”
and it respond and say “i am hearing”. Then we can say our command and based on
the response success or failure it tells the output.