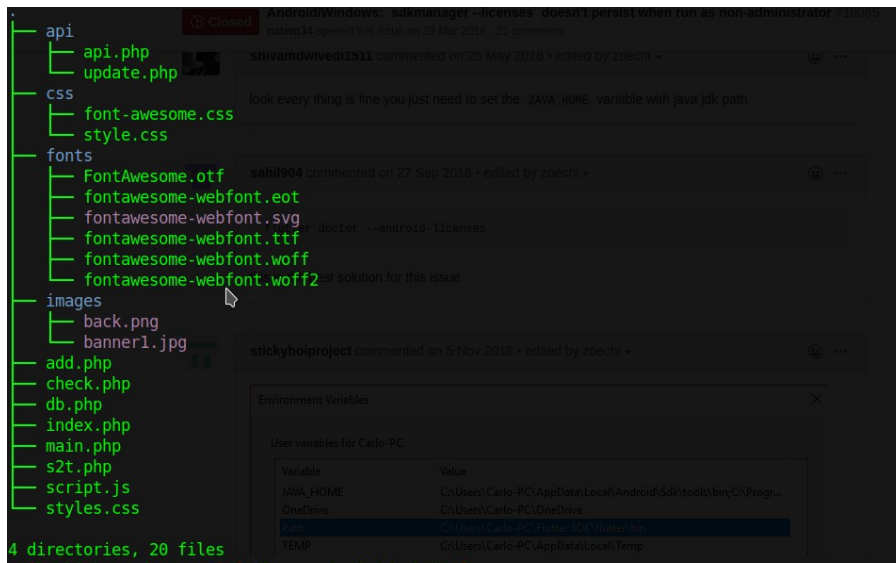


IOT BASED HOME AUTOMATION WITH VOICE RECOGNITION

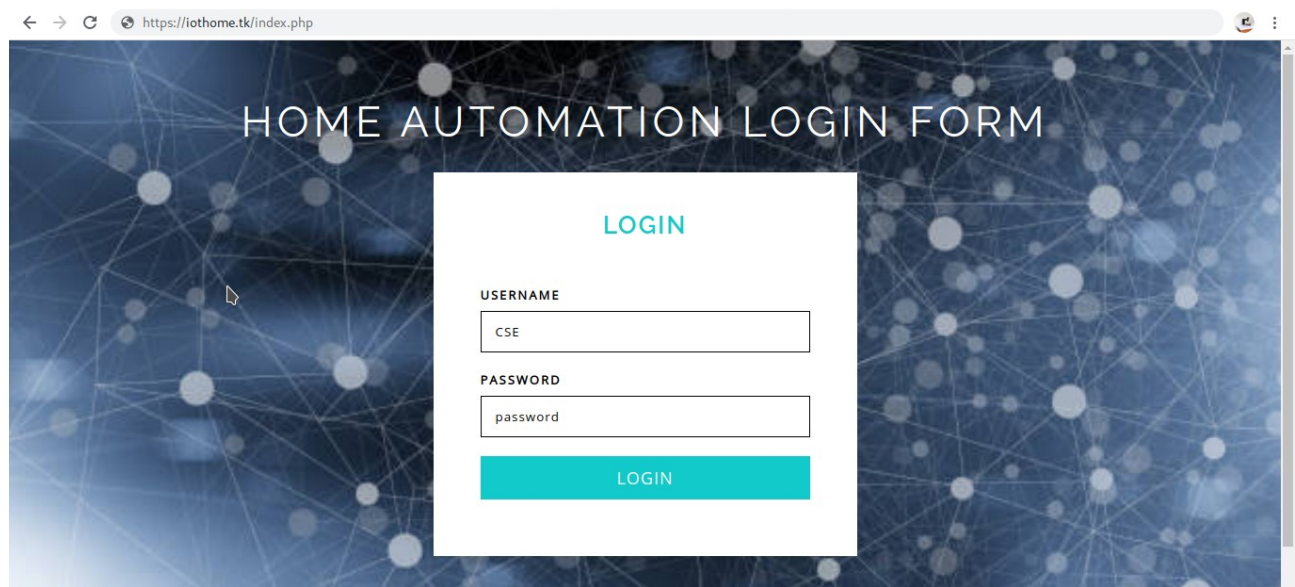
Website / Application:



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

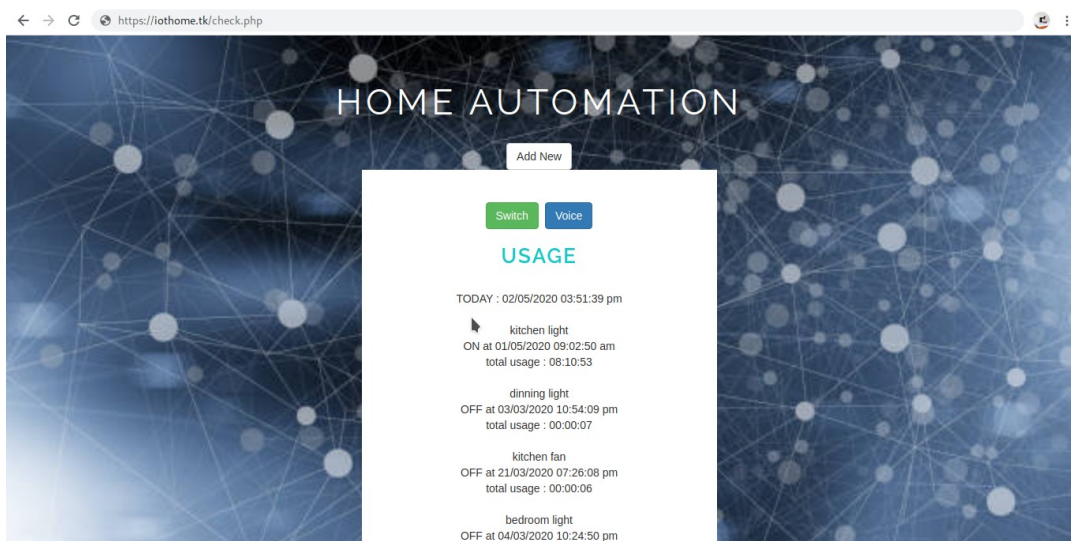
>> check.php – once you logged in .. you will reach this page..

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

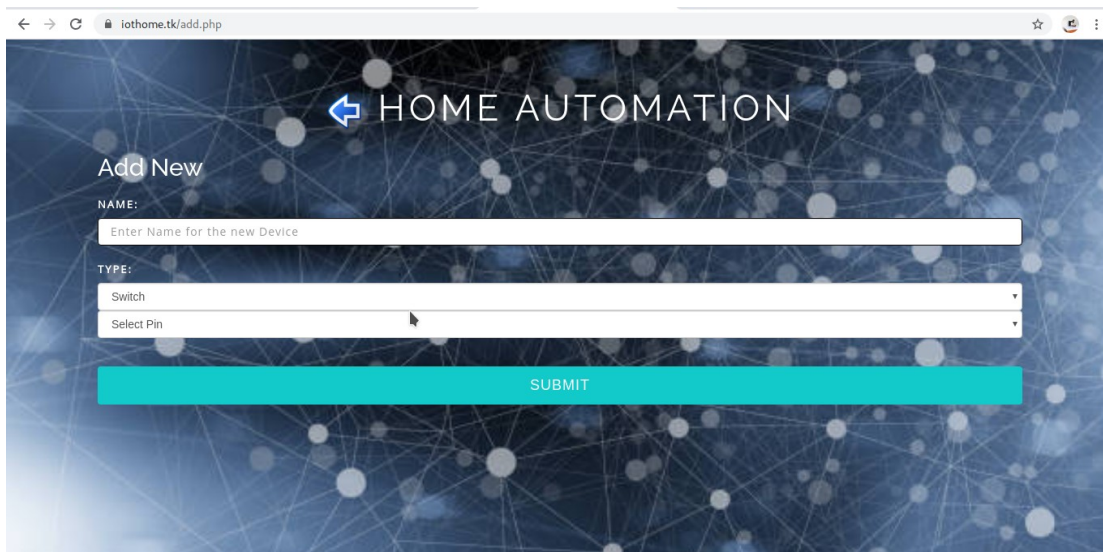


>> add.php – Once you click “Add new” button from check.php, you will reach this page

Here you can add new device to the device.. by giving details –

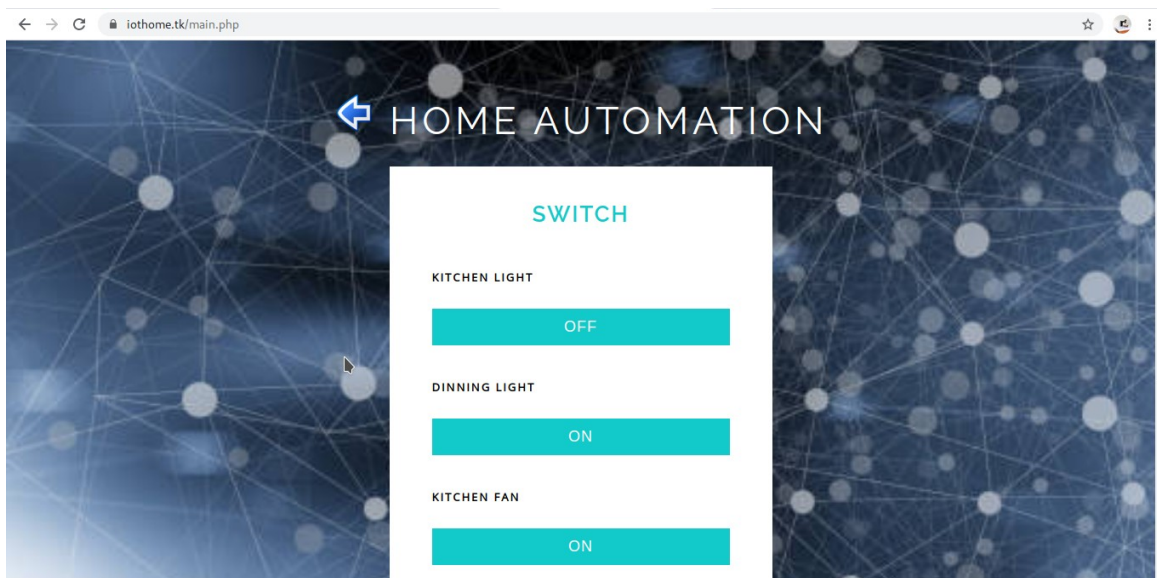
name of the device, device type(switch/regulator), pin value(gpio pin number)

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



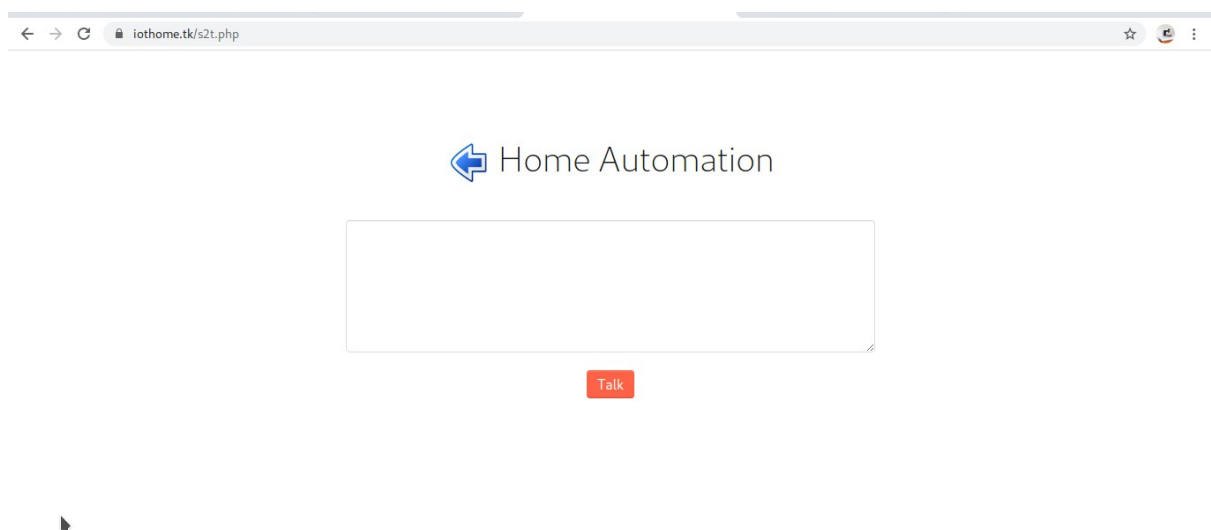
>>main.php – Once you click “switch” button from check.php you will reach this page
Here you can see the switches of each devices. You can On/Off the device

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..

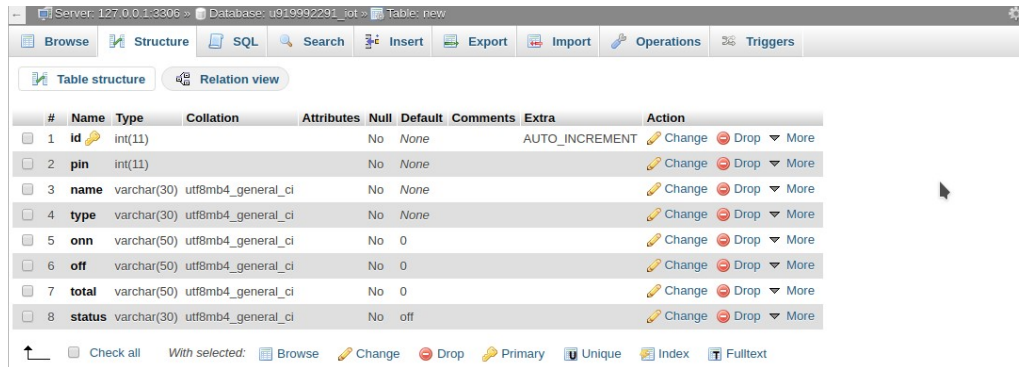


- >> script.js – The voice command from s2t.php file will be send to this page and here the voice will be converted to text, and this call the api- update.php by passing the voice command. The responds from the api will be either “success” or “no command found”. This response will be converted to voice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

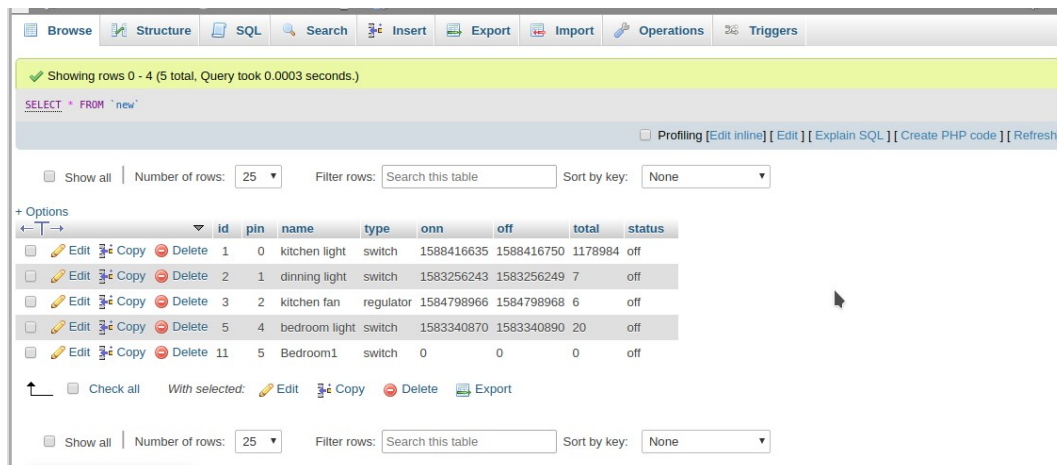
Table Structure



The screenshot shows the 'Table structure' view for a table named 'new'. The table has 8 columns: id (int(11), AUTO_INCREMENT), pin (int(11)), name (varchar(30)), type (varchar(30)), onn (varchar(50)), off (varchar(50)), total (varchar(50)), and status (varchar(30)). All columns are using the utf8mb4_general_ci collation and have no attributes, null values, or defaults. The 'id' column is the primary key.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	pin	int(11)			No	None			Change Drop More
3	name	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
4	type	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
5	onn	varchar(50)	utf8mb4_general_ci		No	0			Change Drop More
6	off	varchar(50)	utf8mb4_general_ci		No	0			Change Drop More
7	total	varchar(50)	utf8mb4_general_ci		No	0			Change Drop More
8	status	varchar(30)	utf8mb4_general_ci		No	off			Change Drop More

Table Sample Data



The screenshot shows the 'Table Sample Data' view for the 'new' table. It displays 5 rows of data. The columns are: id, pin, name, type, onn, off, total, and status. The data is as follows:

id	pin	name	type	onn	off	total	status
1	0	kitchen light	switch	1588416635	1588416750	1178984	off
2	1	dinning light	switch	1583256243	1583256249	7	off
3	2	kitchen fan	regulator	1584798966	1584798968	6	off
5	4	bedroom light	switch	1583340870	1583340890	20	off
11	5	Bedroom1	switch	0	0	0	off

the column “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 continuously

>> rasb.py – this python file executes and continuously calls 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 executes and continuously records the commands from outside.

This file mainly converts the voice command to text using google api feature, and does a post request to api – update.php with parameter -command. The response from the api will be converted to voice and played.

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