IOT BASED HOME AUTOMATION

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Vatsal Shah

(15MECE23)

Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

May 2017

IOT BASED HOME AUTOMATION

Major Project Report

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Vatsal Shah
(15MECE23)

Under the guidance of

External Project Guide:

Mr. Jaydeep Dhrangdharia

Senior Software Engineer

Glide Technology Pvt. Ltd.,

Ahmedabad

Internal Project Guide:

Prof. Jayesh Patel

Assistant Professor, EC Department,

Institute of Technology,

Nirma University, Ahmedabad.

Electronics & Communication Engineering Department

Institute of Technology

Nirma University

Ahmedabad-382 481

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgment has been made in the text to all other material used.

- Vatsal Shah

15MECE23

Certificate

This is to certify that the Major Project entitled **"IOT BASED HOME AUTOMATION"** submitted by **Vatsal Shah (15MECE23)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge,haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Prof. Jayesh Patel

Internal Guide Program Coordinator

Certificate

This is to certify that the Major Project (Phase- I) entitled **"IOT BASED HOME AUTOMATION "** submitted by **Vatsal Shah (15MECE23)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Mr. Jaydeep Dhangdhariya
Senior Software Engineer
Glide Embedded Technologies Pvt.LTD
Ahmedabad

Acknowledgements

I would like to express my gratitude and sincere thanks to PG Coordinator of M.Tech Embedded Systems program for allowing me to undertake this thesis work and for his guidelines during the review process.

I take this opportunity to express my profound gratitude and deep regards to **Prof. Jayesh Patel**, guide of my major project for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I would take this opportunity to express a deep sense of gratitude to **Mr. Jaydeep Dhrangdhria**,Sr. Software Engineer,Glide Technology Pvt. Ltd. for his cordial support, constant supervision as well as for providing valuable information regarding the project and guidance, which helped me in completing this task through various stages.

Lastly, I thank almighty, my parents, brother and friends for their constant encouragement without which this assignment would not be possible.

- Vatsal Shah

15MECE23

Abstract

Advancement in smart system for daily household is require. People in busy routines mostly forget to switch off household or office appliances at no use which make power bills costlier. Their is a need of technology to know the state of each appliances and also to control them from any place with Internet. Meanwhile Security in the office premises and homes also becomes a challenging task at economical rates. People are in worry that an unauthorized person should not enter in its office premises or home in absence of him. There are swiping RF ID cards available to provide security but their is a fear of getting lost and get access to unauthorized person and also employees have to keep with them all time. So their is a technology needed to provide more Door Lock security. Here I have developed IOT based Home automation where user can control appliances with Internet. Further their is an additional feature of IOT is Door lock which is using Bluetooth ID of employees mobile phones or house member mobile to unlock the door. People are always carrying smart phones with them, so they have to just open the application and punch the entry into it. While when owner is far away and if he has to give access to some guest to enter in premises, he can give access to them with Internet from anywhere.Here, I have developed an android Application where user can see the state of appliances and can also control them. Their is also a feature to set destination of user's home where appliances can be on/off when user is within 1Km to destination automatically. Here I have developed web based User Interface for on board storage of house members/employees entry in out details where user can able to see from anywhere its entry exit time. In this User Interface user can able to add new person, delete person or edit person detail. All these functionality are implemented through python with use of mysql database. For storage of these staff data and details i am using MYSQL database and XML files. In this project, efforts are given to work with voice assistant JASPER platform to control applications through user voice command. I have used PUBNUB and OPENHAB framework to work this device with wifi and make this device smart.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Need of Advancement in smart systems for daily household is require.Their is a requirement for system integration and increased extensibility.Demand of resources is increased.Increased concern about resources and management.

1.2 Objective

The main objective behind this project is to develop new features in door lock and make lock smart by accessing through wifi with greater security.

1.3 Requirements

Require learning of C, Python language while it also requires knowledge of html, basics of java script and fundamentals of flask.

1.4 Project Work Flow

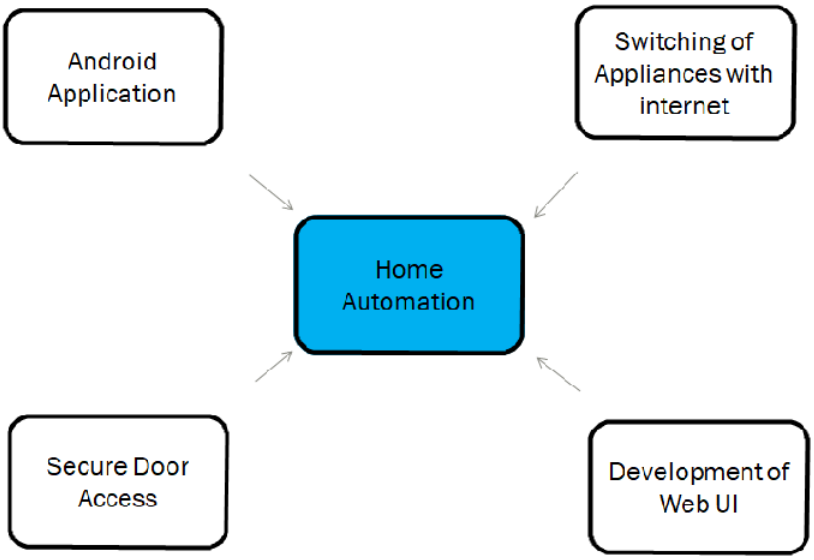

Figure 1.1: Project Flow Chart

Figure shows the flow or features supported in IOT based Home Automation. Their is a feature to control home or office appliances with help of Internet irrespective of any location in world. Their is an android application is developed which allows user to control appliances and to see the state of particular appliances. Their is also provision to secure the door of home or offices and can also controlled open/-close with Internet. Owner can be able to see who enters or exits home or office premises at respected time.

Chapter 2

Framework and Platform

2.1 OpenHAB

OpenHAB is a open source software for integrating different home automation systems and technologies into one single solution that allows over-arching automation rules and that offers uniform user interfaces.
OpenHAB is designed to be a vendor independent in which it is runs on any hardware platform which is capable to run Windows, Linux, MAC based on Java Virtual Machine(JVM). OpenHAB gives us an integration platform in which all home automation technologies can be embedded in one technology.
Openhab provides different web based User Interfaces to user which can be supported in Android OS as well as IOS. OpenHAB platform provides various APIs for different technology to integrate with real time Home automation services.
OpenHAB supports many technologies like MQTT, Z-wave, TCP-UDP, XBMC, OSMC, Lifi and all other various technologies.
OpenHAB provide APIs for Raspberry Pi3 which can easily be installed and started working over with it.
OpenHAB provide various bundles which can be installed in Raspberry Pi and calls them based upon requirement.

2.1.1 Installation Of OpenHAB

OpenHAB can be installed in Raspberry Pi from download section of OpenHAB.org. After unzip the file there will be two folders named "addons" and "Configuration". Configuration folder have subfolders named "items", "Persistence", "rules", "sitemaps", "scripts".
There are different types of debugger provided by windows. They all are integrated into windows driver kit which provides the driver development environment. There are six different debugging environments that you can use after installing windows driver kit and visual studio.

- Items : This file consists of all contents of every control devices we want to interface like each bulb, fan, AC, lights, room1light, room2light. Each content can be assigned in groups like room1, office and device can also be interfaced with specific bindings.

- Sitemaps: Sitemap file concerned with the user interface when user opens in the android or IOS mobile.In this file we can define the specific groups to be laid down and by clicking on this group it will displays the control devices for that particular group.

- Rules: Rules file is the actual file that comes to action by clicking on the particular device. This file defines the action to be performed like turning on the LED by clicking ON button on toggle LED switch or turn off the power by switch off the toggle LED in user Interface.

- Script: Script file can be used to define complex script, while it gives same functionality as Rules.

Among all these files sitemap file and items file consider to be an important file for configuration.

Here, In "**demo.items**" file insert
Switch RaspiLED gpio : "pin:4"

while in "**demo.sitemaps**" file insert
sitemap home lable ="home"
Switch item = RaspiLED

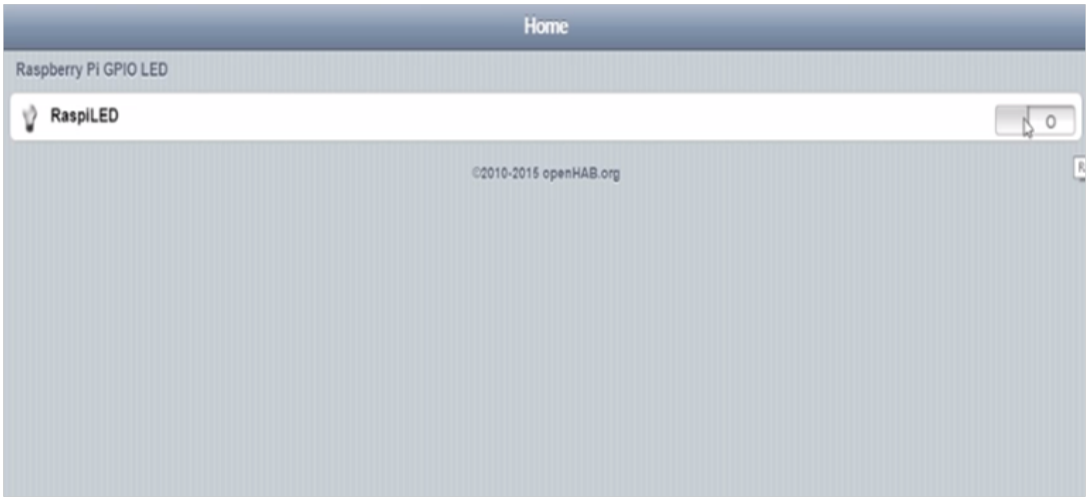

Figure 2.1: OpenHAB Demo

Figure 2.1 represents the demo example of Openhab, where we can see that respective slider of LED , bulb image on left side of html page, background color appears which come with its bundle at the time of installation.
This example of web page can be accessed locally with local host but to access this bundles through internet their is cloud service available of Openhab named as myopenhab.org

2.2 Pubnub

Pubnub is a global Data Stream Network (DSN) and real time infrastructure-as-a-service (IaaS) company (San Francisco, California) and provide solution for IoT device control, iot secure messaging, real time dashboard and many more. Pubnub works on the principle of MQTT Protocol. Pubnub provides real-time publish/subscribe messaging service built on data stream network. Pubnub also provides API for various boards line raspberry pi, Arduino, Texas instruments, Microchip.

2.2.1 Publish/subscribe key features

User can get unique API key by getting an account this publish and subscribe keys are used at publisher and subscriber side. Publisher publish messages to channels and through same API keys subscriber get data through respective published channel in just 0.25 seconds. There are no limitation on number of channels.

2.2.2 Pubnub Working Flow

In this basic example publisher publish message on channel named Test and published message is Demo and subscriber is subscribed to channel Test get Demo as a callback message and this is done with python, So in the beginning pubnub module is imported and both publisher and subscriber having same pub/sub keys which makes communication possible.
Pubnub works on the fundamentals of Message Queue Telemetry Transport protocol where public key is shared between sender and receiver.

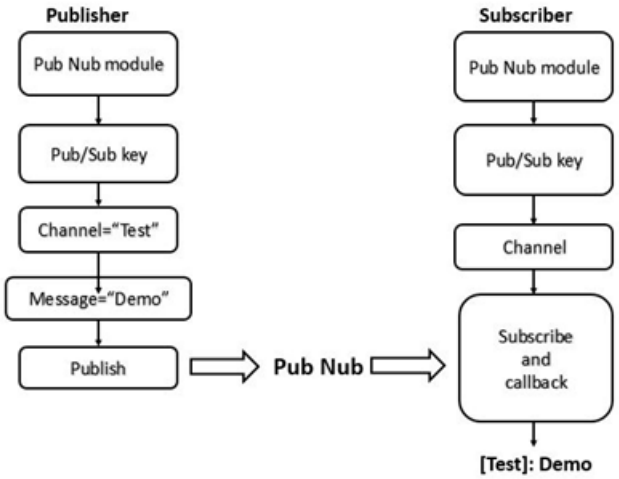

Figure 2.2: Pubnub Working Flow

2.2.3 Home Automation Example

This example contains simple Web UI and through that devices are controlled. The sliders in UI is to control brightness (by changing duty cycle value) of leds. Here from html source file(index.html) java-script is called in which public key is placed through which this page communicates with Pubnub cloud.
While in server side python script is going to run continuously to publish message into pubnub cloud and javascript subscribe to this messages through Pubnub at a time of reloading the webpage.

2.3 Hosting Service

As seen from the above Pubnub example we can access web pages locally or by localhost in linux platform. Now if we want to access this web pages globally or access from any place we require platform who host our webpages.

There are paid platforms available who host our websites with unique domain

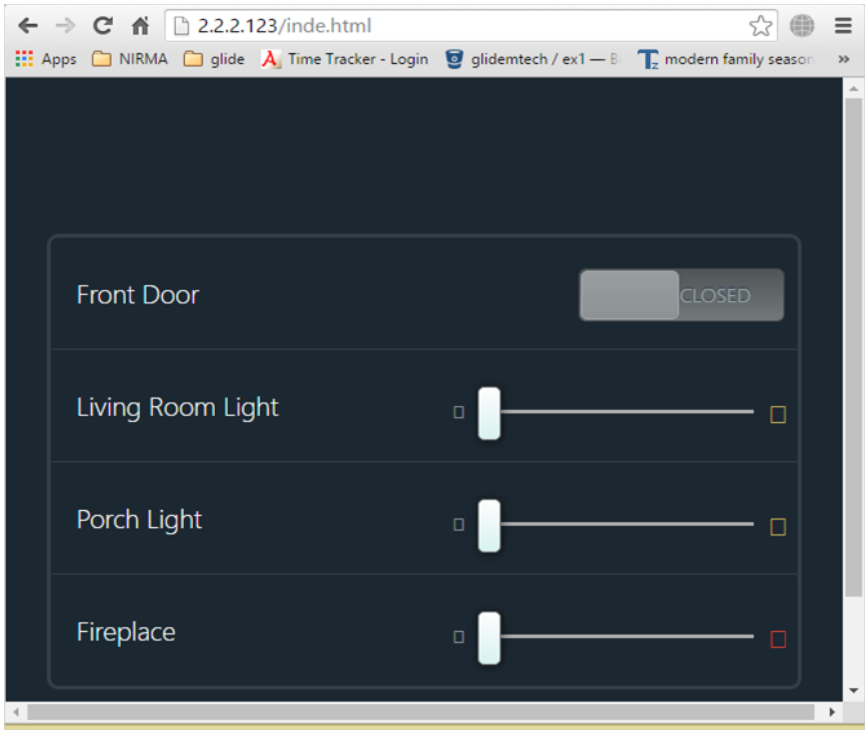

Figure 2.3: Pubnub Example

name. **Byet Host** is the service who gives service to host web pages freely. We simply have to upload web pages and related css and .js files online.

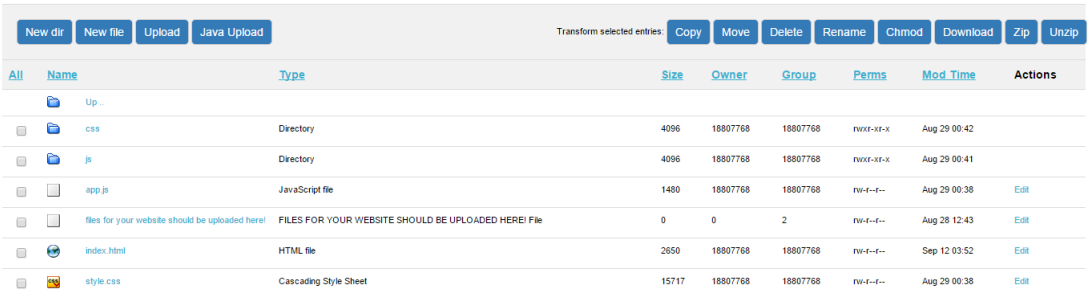

Figure 2.4: Byethost upload web Pages

As seen from the figure index.html style.css and app.js files have been uploaded in the website panel.byethost.com Byet provides ad free subdomains like MySQL,

PHP, FTP, Vista Panel and many more. byet have their own clustered hosting network having 24core CPU xenon webserver, mail server, FTP server, hardware load balancer, MySQL server etc. we are going for VistaPanel: free hosting control panel.

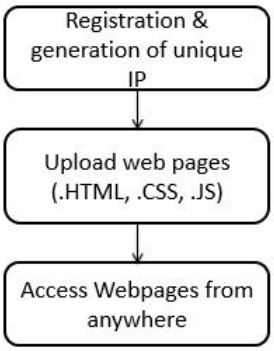

Figure 2.5: Byethost Working Flow

Byethost provides 1gb of free space to the user account in the form of 5Mb of space to the html files.

Figure 2.5 represents the procedure to upload webpages in byethost services. Firstly we have to do the registration in byethost, byethost provider gives us the ip that we can access from anywhere.

Secondly we have to upload webpages in html doc of online file manager in website panel.byethost.com. Respective css file and js file can also be upload by making css and js folder in html doc.

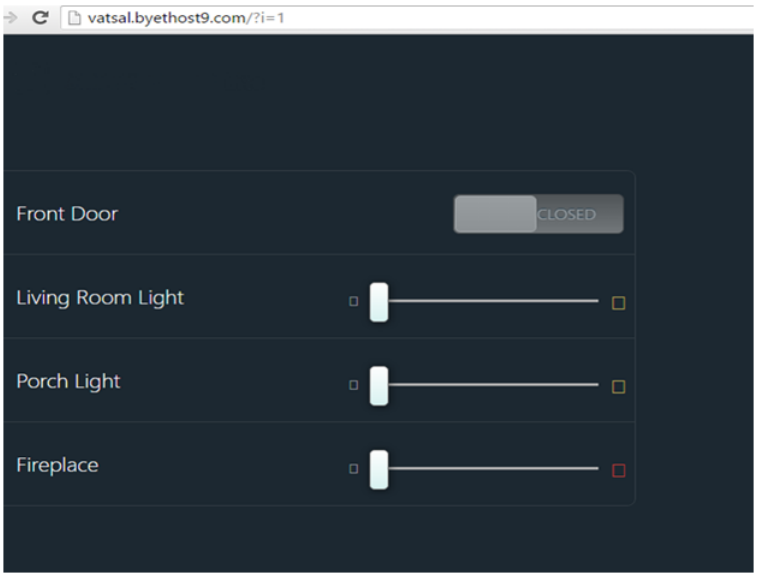

Figure 2.6: Byethost example

Here in figure 2.6, i implemented project to control brightness of LEDs with help of python and PUBNUB framework.

Chapter 3

Protocol And API

3.1 Protocol And API

Raspberry pi3 comes with inbuilt wifi and Bluetooth version 4.1. The bluetoothctl tool is require to use inbuilt bluetooth functionality of Raspberry pi. Various commands can be executed in bluetoothctl tools like scanning as well as pairing.
To launch bluetoothctl tool, type *bliuetoothctl* in command terminal, while to connect the device:

- From the bluetoothctl tool, type in *power on* to make sure Bluetooth is on.
- Type in *agent on* to make sure Bluetooth is running.
- Type in *scan on* to start searching for nearby Bluetooth devices.
- we should see all our various Bluetooth devices pop up with their MAC address (its the number shown on the screen next to the device name, usually something like 50:54:B4:45:00:EB). Make a note of the MAC address for the device you want to pair with your Raspberry Pi.

3.1.1 Interfacing with RPi

To work bluetooth with python, we need to install python-bluez which is APIs which can be use directly. **Psudo code**

import bluetooth
import RPi.GPIO as GPIO
//continuous loop
bluetooth.lookupname('MAC Address of Device')
//if connected: device or LED turned ON
//else: device or LED OFF

Sudo code represents the basic code to test bluetooth in Raspberry Pi 3 through python script. Firstly i called the bluetooth library and GPIO library to work bluetooth with GPIO of Raspberry Pi3.
In infinite loop Raspberry Pi continuously looking for specific bluetooth device MAC address and trying to connect it. As long as blutooth connection occur Raspberry Pi switch on off the GPIO pins to on off devices.

3.2 Voice Recognition API

There are voice recognition technologies available in the mobile phones or iphones. Siri is the one of the voice assistant technology developed by Apple company, while for android phones Google assistant is available. While these technologies are for smart phones, there are platforms available or developed which allows people to develop their own voice assistant through that people can control electrical appliances with their voice.

Amazon Echo, JASPER are such open source platform for developing voice controlled Home Automation.It allows people to use their own voice to listen updates from social media, get to know about the daily emails, weather condition of living area and many other things.

In our project, i am using open source platform of JASPER which always listens command from few meters also. Jasper does not work on the smart phones, it require only a stand alone unit with microphone, a internet connection and a small speaker.

To get started with Jasper platform user require Raspberry Pi module, 4gb sd card, microphone and speaker unit and internet connection.

Their is also a precompiled image available in the official site where we can just dump into memory card and can execute the script. There are also other voice assistant platforms are available which is supported in Raspberry Pi 3 but JAsper platform gives higher accuracy compare to them. JASPER gives options to user to configure the command file about how to speak specific commands.

Addition of any feature like switching LEDs apart from time, weather command is easy in JASPER platform which is based on python script. Interfacing or to work with Raspberry pi 3 GPIO is easy as ready libraries are available for JASPER platform.

3.2.1 Jasper Platform

Jasper platform uses speech apis and text apis to convert speech to text and text to speech. As shown in figure USB microphone which attached to the USB ports of Raspberry Pi captures the commands of the people.

Various speech to text engines available and supported in jasper platform like Google speech to text engine, WITAI STT engine, Pocketspinx STT engine, Julius STT engine.

Text to speech engine like MAry TTS engine, eSPEAK TTS engine, Flite TTS engine, Pico TTS engines are supported in JASPER platform.

In our project we are using espeak TTS engine, where we can get its unique id through registration in official site and can place it in a profile.yml configuration file of JASPER platform. Figure 3.1 represents the requirements to operate voice commands through jasper.

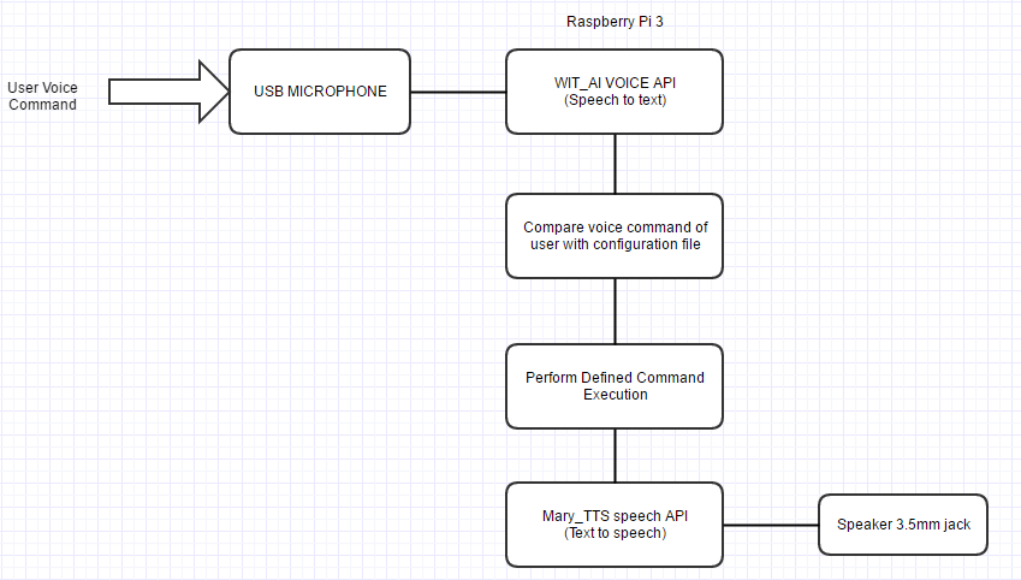

Figure 3.1: Voice Controlled Assistant

This STT engine coverts speech into text and compare this text command with command configuration file. Configuration file checks the valid command is their in the directory or not. As if the command is valid, it performs the defind function related to that command.

Jasper comes with pre installed various commands like "what's the time?" which indicates the real time, "Weather" command gives weather condition of area, "Tell me a Joke" command speaks the joke with TTS engine.
The results of this commands will be heard in speaker which is connected in 3.5mm jack in raspberry pi3. TTS engines Google TTS and others will convert results of commands into speech from text

```
carrier: PHONE_CARRIER_NO_QUOTES
first_name: Vatsal
#gmail_address: postmaster@sandboxe1411c78f2ec4b388f13
#gmail_password: f78475411f7164b0b0efe7d3d5036980
gmail_address: vatsal.shah97@gmail.com
gmail_password: Enter Password
last_name: Shah
location: '380063'
phone_number: '9712862669'
prefers_email: true
timezone: US/Eastern
wmo_id: 42647
# TTS CONFIG - Male Voice Coming Soon
## WORKS OUT OF THE BOX, NO CONFIG
#tts_engine: mary-tts
tts_engine: espeak-tts
stt_engine: witai
witai-stt:
  access_token: PIT3YJGSUYDSRCK3XWQWSDMS7DM7IWRV


## PLUGINS  ##
```

Figure 3.2: Jasper file

Their is a configuration file in JASPER in which developer can change the configurations like email id and password for email updates, area location WMOID for weather location and country setting for specific time zone.

Figure Jasper file shows the "profile.yml" file in which user can change or edit the configurations.
After installing all speech to text engines and text to speech engines configuration of .yml file should be made.

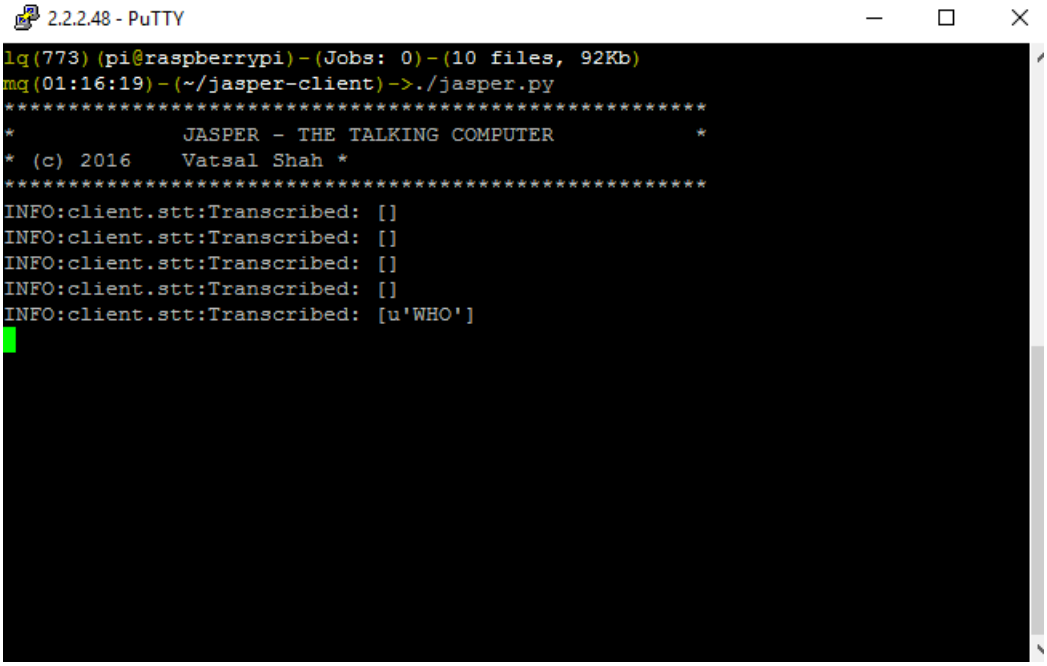

Figure 3.3: Working of Jasper

After configuration python script should be executed, here screen shot shows that whatever speech speaks by person displays in terminal.
As long as the command matches with the commands with the configuration file small beep sound will be heard that this is the valid command.

Chapter 4

REST Web Services

Web services are used to exchange data between applications using different standard and open protocols. Applications are written in different language and running on different environment so to provide interoperability web services are used to exchange the data.

4.1 What is REST?

- Client-Server:
  Server who offers services are isolated from the client who consume that services and vice versa.
- Stateless:
  To carry out the request by server each client side request requires to contain all the information that may be needed to server.
- Cache-able:
  Server notifies to client that the request is cached or not.
- Layered System:
  Rest follows layered approach where server does not directly communicate

with client there is another server present rather than end server so it allows intermediate request responses.

- Uniform Interface:
  Communication method between client and server must uniform.

- Code On Demand:
  For clients request server can provide optional script or executable code.

4.2 What is RESTful Web Services?

Restful web services are based on rest architecture (design to fit http protocol). Restful web services is used to create apis for web based applications because restful services are light weight, maintainable and highly scalable. In rest service between client and end server uris present as mediator and client send request to uris using different http methods. The methods of http request affect different resources in standard way:

Table 4.1: RESTful Methods

HTTP Method	**Action**	**Example**
GET	Retrive Resource Information	http://example.com/api/order
GET	Retrive Resource Information	http://example.com/api/order/123
POST	Create resource	http://example.com/api/order
PUT	Update resource	http://example.com/api/order/123
DELETE	Delete Resource	http://example.com/api/order/123

4.3 Flask Web Framework

Flask, a micro web framework which is specifically written in python language.It does not require any particular library or any apis to work with it, while flask supports various extensions that can add applications features which can be executed in Flask itself. Flask has its own development server and debugger which can be implemented in python.

It has integrated support for specific unit or module testing.

It can be use with restful web services to call specific unit without reloading the web page. The functionality can be called by "@app.route("/")"

Flask supports various functionality to interact with various databases. Here in my project i accomplished to fetch the data from the database using flask. Database data fetched using python language and displayed in web pages through java script.

Psudo code

import Flask

@app.route("/hii")

function hii():

// return "hello vatsal"

//main function

//run flask(localhost)

Below represents the python code to call specific function from web page using flask. @app.route line represents the path or header to be called on web page. As long as user will enter "127.0.0.1:portno./hii" function hii will be called and "hello vatsal" will be appear and can be seen on web page

Chapter 5

Android Application Development

Figure indicates the screen shot of application which is made. An application is created which have features to control appliances with help of Internet. In switching appliances screen shot, their can be seen various devices like light, fan, bulb which can be controlled through this application. In this particular data or state of switching appliance is posted and published on PUBNUB framework.

In client side or customer side there will be a hub in the form of Raspberry Pi board. Raspberry pi board subscribed to the PUBNUB framework. It will continuously fetch the data from PUBNUB. Appliances are differentiated with channel name in PUBNUB framework like light appliance is classified as channel "Appliance1" and Fan or bulb appliance is classified as channel name "Appliance2". Their is an additional feature included where appliances can be controlled base on physical location of mobile. For Ex: when an user is within 100meters or 1km to destination, light or AC can be switched ON automatically. Their is an feature provided to enter latitude longitude of respected destination and can be stored in file storage of mobile as a binary file or text file. Their is an button provided "Write to file" and "Start" where user can easily write destination to file location and can start the process by clicking on start button.

As shown in screen shot, their is a background process running which continuously fetch the location through location service in android application and displaying

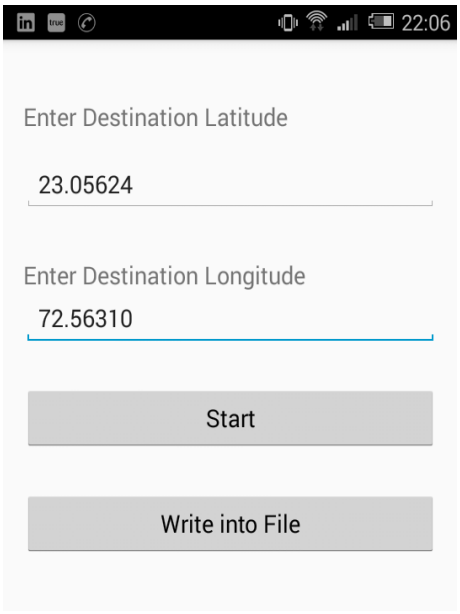

Figure 5.1: User Configuration

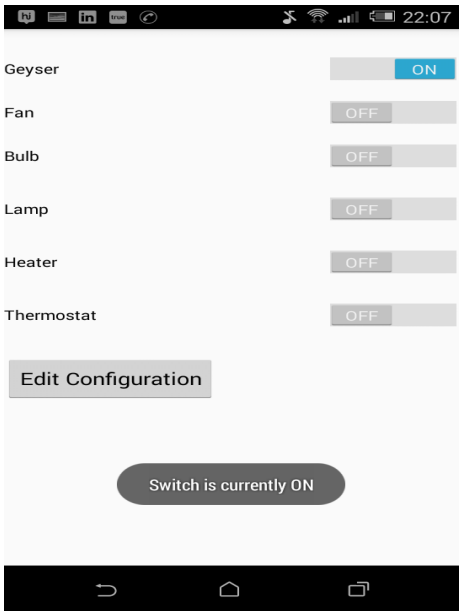

Figure 5.2: Home Page

pop-up of distance to destination on period of 1minute. Their is also an feature to start application automatically. Their is a feature added in android which starts the application at the starting of mobile. In some cases, if mobile switches off then at the time of boot mobile will start the process to fetch the location. At the time of boot, application will run the process to calculate the distance to location and display the distance as pop up.

Figure shows that process is continuously running all time. Their is no need to constantly open the application by the user. Process will run as a background service "All time ON". Process will make the use of Internet to fetch the latitude longitude coordinates. At the time of installation, application will ask to get request for access location as well as to make use of INTERNET or data service. At the time of boot as an app window opens, it will ask to change the destination of object if user wants, otherwise it will fetch previously edited coordinates of destination from file location of smart phone.

Figure 5.3: Application as Background Process

Here, figure shows the application functionality diagram where it describes how our application working and how interface between client side and server side is accomplished with help of PUBNUB framework. Functionality diagram represents

the functionality of each section front end which is Android App, Communicating medium which is PUBNUB, and server side which is Raspberry Pi board.

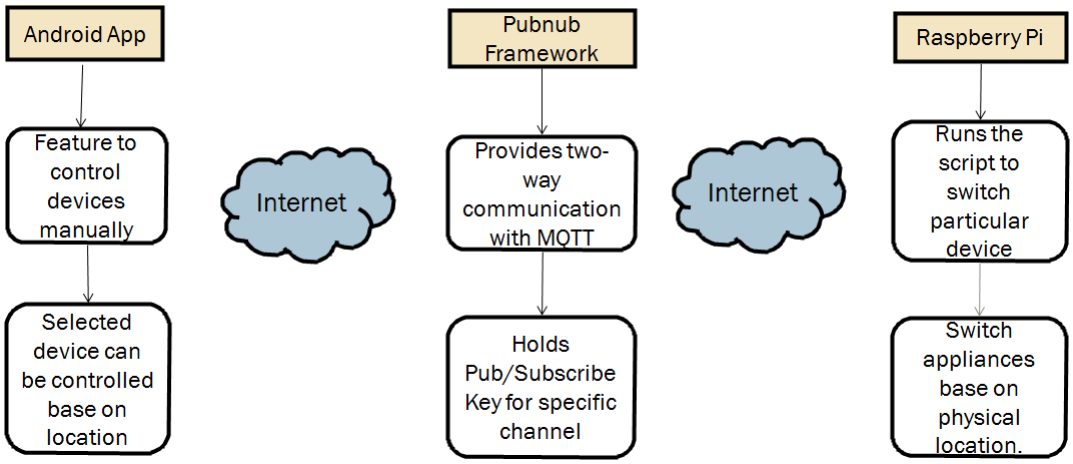

Figure 5.4: functionality diagram

All sections in functionality diagram are communication with medium of Internet.

- **Front end Android App**:

  Android application have features to control home or office appliances with long distance support or from any location with the help of Internet. It also have features to control the device automatically by detecting the location of user to the destination. User can set the location of destination where server is kept.

- **PUBNUB Framework**:

  Pubnub framework provides two way communication between client side and server side with MQTT protocol. Communication with PUBNUB framework can be done with Publish key and subscribe key. User in client side have to subscribe with particular channel of PUBNUB while in server side Raspberry pi will publish the message with publisher key.

Chapter 6

WebUI Implementation

Here i have made Graphical User Interface where records and data of office employees can be seen. GUI displays the on board storage for employees and office staff. On the back end python script is running through flask where the person entry is registered in mysql database.

Here to store the person's data i am using of XML file and mysql database where users data can be fetch through python script and push into the java script file through Ajax web services.It has subsection of Department, Staff, Device, Parent Data, Attendance Report, Shift Management.

- Department: This section displays the various department information in company like Hardware Department, Software Department, Marketing Department, Sales Department. By selecting the particular department, its manager name, number and other information will be fetched from database and displays in web page. Additional Service of Add Department to the company is also their where user can enter Department's name and other related information. Edit or Modification of existing database services also provided where user can edit particular department's data. Here all Department's data is stored or edit in XML file in back end and python scripts fetch data from xml file and push data into html file.

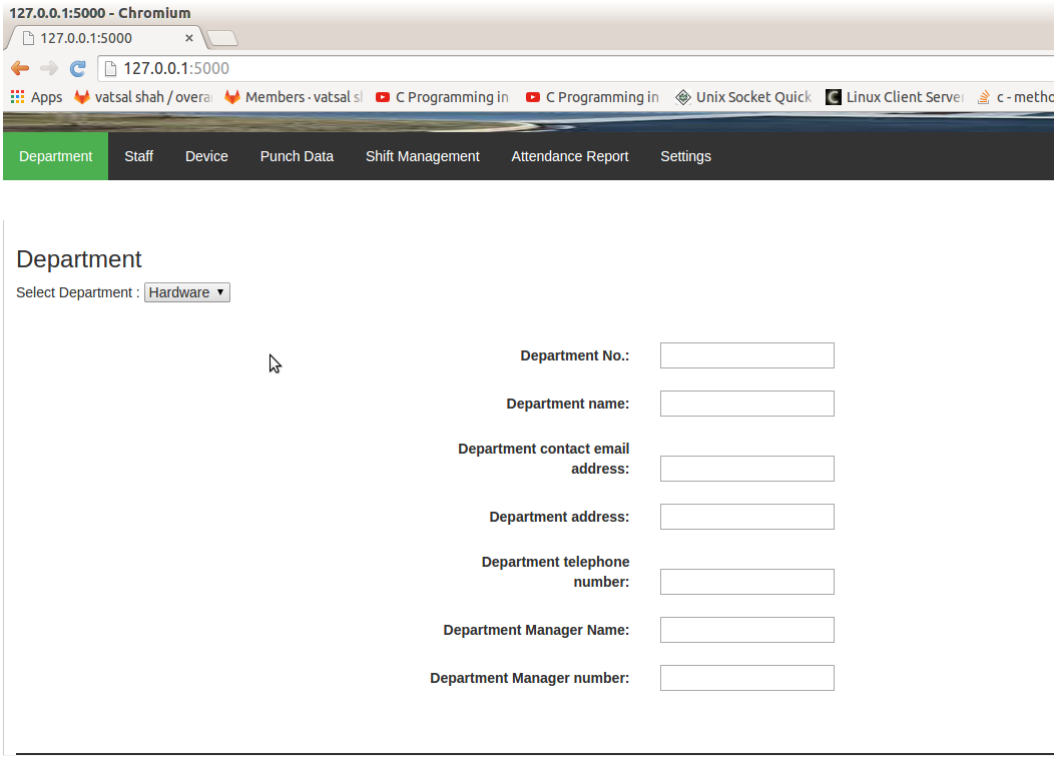

Figure 6.1: Department Webpage Window

- Staff: This section displays the information about staff and employees. A table is created in html where various data of staff is fetched from xml database. In python script this data is fetched and returned with json AJAX query, while this data is received at java script and pushed into html script with particular row column id of table. I have included services of "show all staff" where all staff members are displayed in table. It has also an additional functionality of " show resigned staff" where the recent users which are removed or deleted are displays in html. It has also feature of display a staff or employees name as per sub department. As by clicking on "show sub department staff" it displays the staff of particular department.

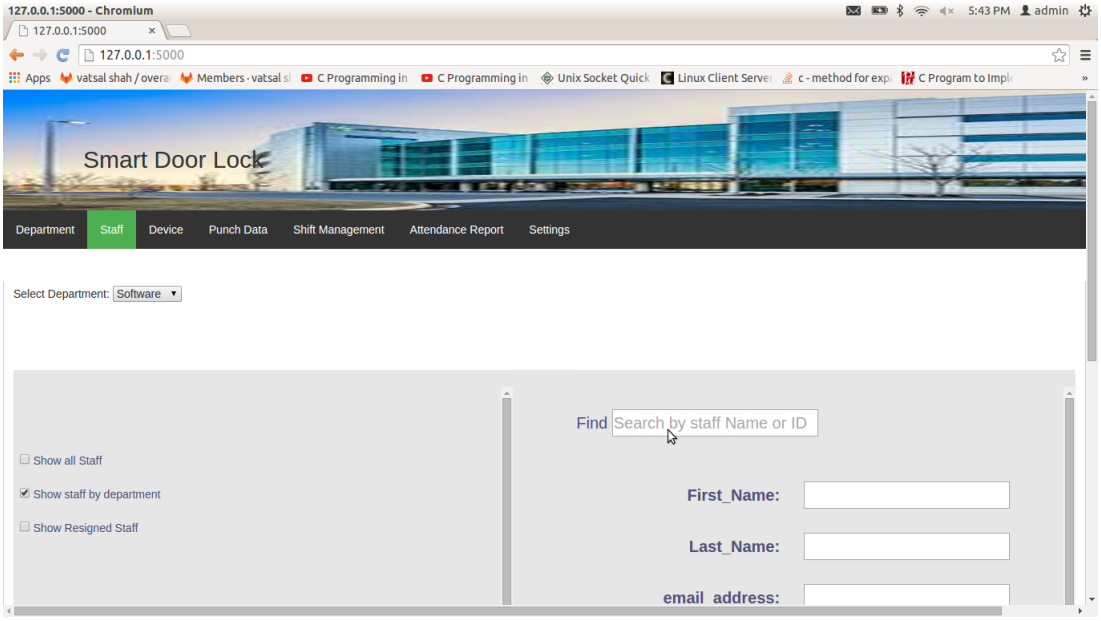

Figure 6.2: Staff Window

Fig. 5.3 represents the python script which is running through the flask. Flask system virtually creating server or act as an server running on local host. As seen from the figure, it is seen that server runs on 127.0.0.1 local host, we can also apply static ip to the board to access this web pages from anywhere. Here in the flask i am calling python script where functional modules of flask is written. Their is need to give path of html page which is to be rendered when flask server runs.I gave and written index.html where i have written my whole html script as a path file name.

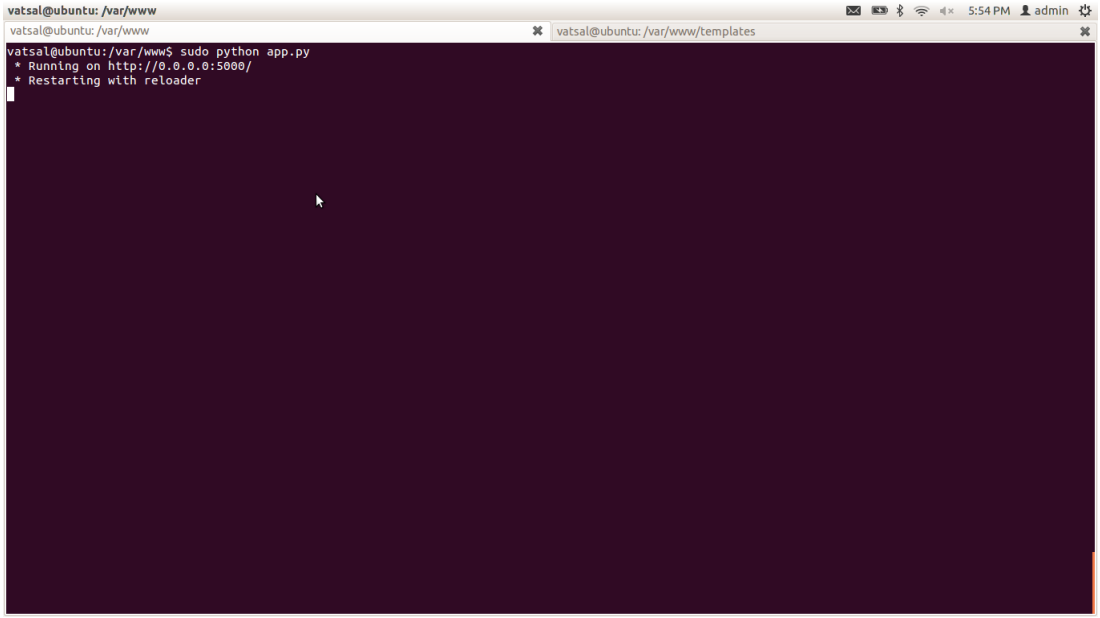

Figure 6.3: Flask Terminal Response

As shown in staff section, their is an services of "add a new staff" where user can enter staff details like name, surname, email, gender, Bluetooth ID, Marital Status, Contact Number, Joining Date. As shown in figure 5.4 i have entered First name as "nirma" and last name as "University" and respective email and other information and click it on ADD button. On back end as long as ADD is clicked, this details are stored in XML file by creating a new children in root parent of XML file. We can see in the XML file that the new data with "NIRMA UNIVERSITY" is created.

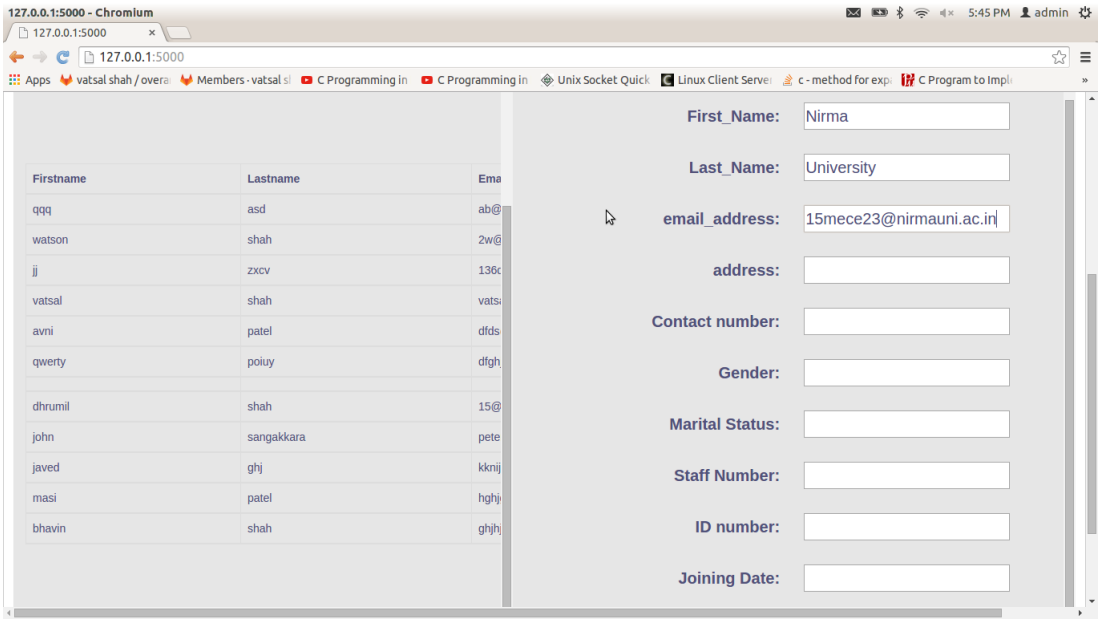

Figure 6.4: Adding Staff input

As long as ADD button is clicked data is entered in XML file, here in figure 5.5 we can see the response of web services. Here i made the functionality in such a way that when i add new staff details, this details should appear on table which is on the left side container in html page shown in screen shot. so, As long as i reload the page, python script will fetch all data from xml file and displays in html. Here in fig 5.5 it can be seen that if data is fetched succesfully i got the response of ":200" which is succesfully fetched othewise i could got a response of ":404" which is forbidden error, while response of ":303" indicates failure of response.

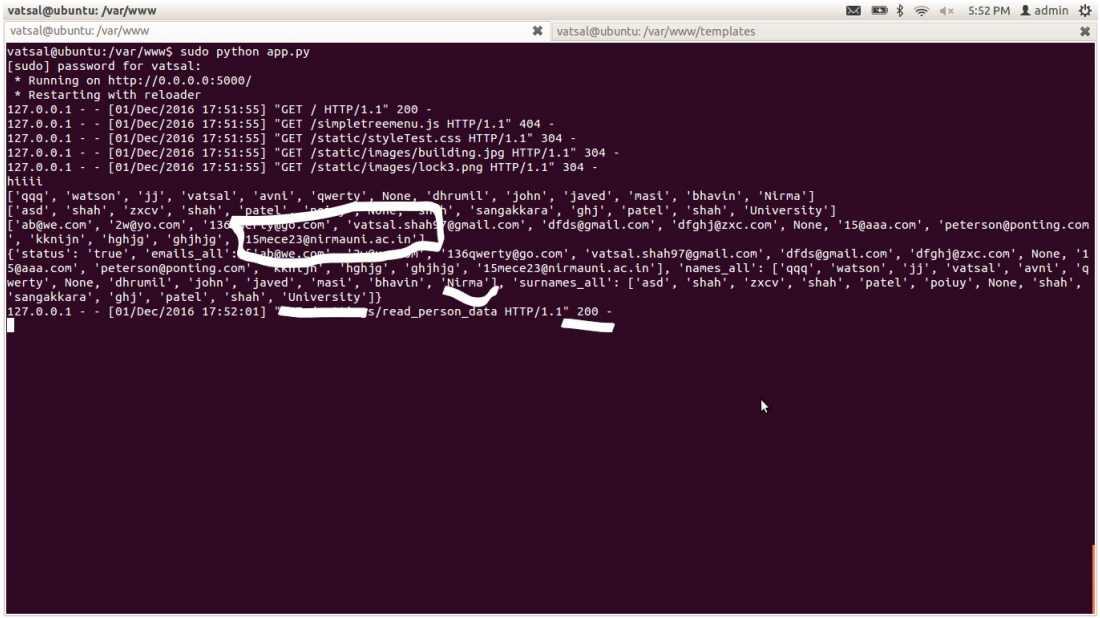

Figure 6.5: Staff Details Response

we can see here in the container table in html page that after getting the response of "200", new added staff information will appear on html table. Figure 5.6 represents the newly added staff details NIRMA and UNIVERSITY appear on table. Similar functionality of Delete the staff, modify or edit the staff details, clear record, quick setup functionality is provided. At the time of reload all data from xml file is fetched through python script by flask and data is pushed into javascript with ajax web services. In the java script i am deleting all rows of table in html and fetch it again through xml file, so same data does not appear on table rows of html.

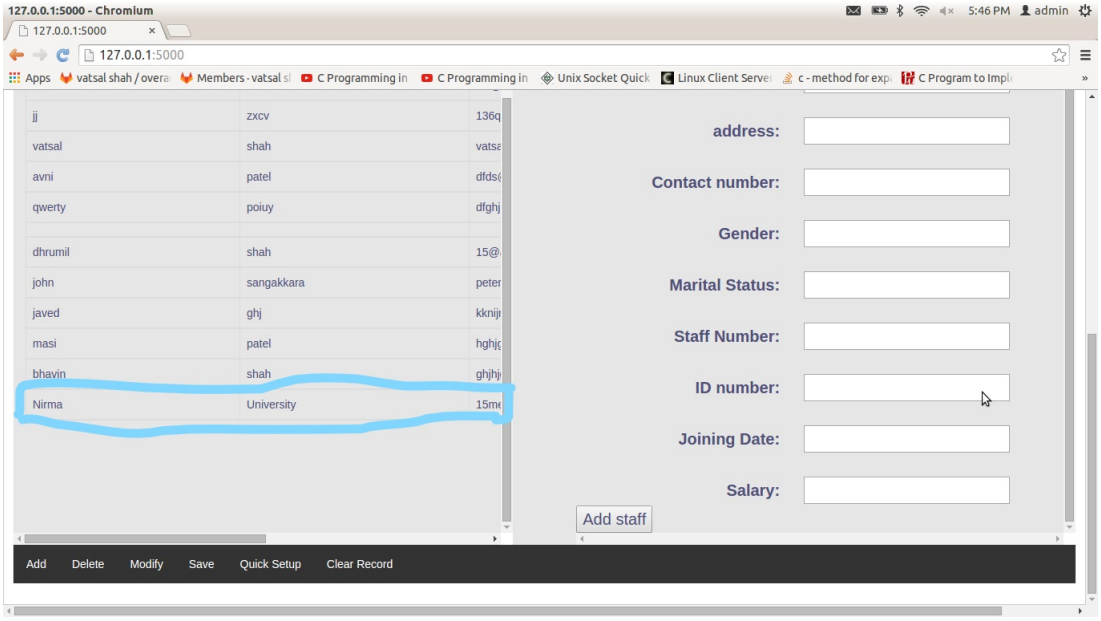

Figure 6.6: Add staff window

- PunchData: In this section it displays the punch data of particular employee, this section indicates the in out time of employees, indicates late time in minutes as well as early out and other related options date wise. As shown in figure as by selecting date and selecting person's name, database will be fetched and displays in html. In python script a functionality is created that when person will enters in office through blue term android application, its time is recorded and stored in mysql database in table format.Python script will fetch the data from database and displays in html through Java script and Ajax web services.In the java script section i am deleting all rows of table in html and fetch it again through database file, so same data does not appear on table rows of html.

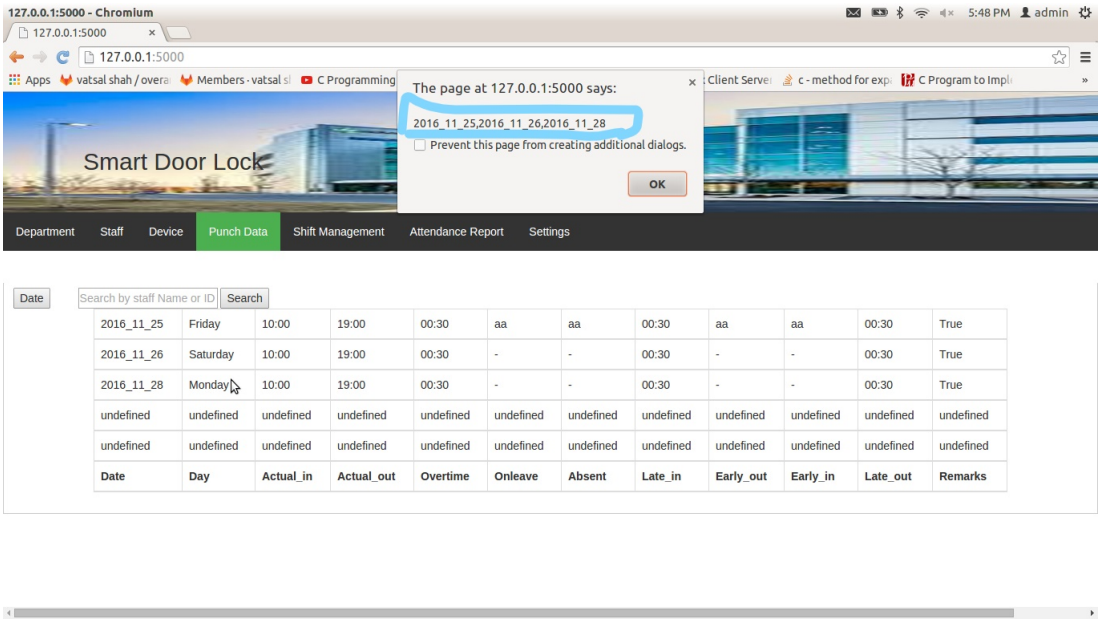

Figure 6.7: PunchData Window

As long as we reload the web page and click on PunchData tab i put an ”alert” of data which can be seen. Here in figure 5.8 we can see the response of web services. Here in the terminal we can see that all information of database table is fetched. so, As long as i reload the page, python script will fetch all data from database table and displays in html. Here in fig 5.8 it can be seen that if data is fetched succesfully i got the response of ”:200” which is succesfully fetched through ajax web service othewise i could got a response of ”:404” which is forbidden error, while response of ”:303” indicates failure of response.

Chapter 7

Conclusion

By developing this projects the goal to automate home applications have been achieved. Devices can be operated automatically without human intervention based on his/her mobile location. Security of office and houses is extended to certain level at economical rates. Additional facilities of on board data storage feature as well to control door with Internet access and voice assistant automation is developed in this device. We can use this device as an alternative to Rf id card based door lock devices in office premises.

Chapter 8

Future Scope

More functionality like integration of Artificial Intelligence with IOT device can be done. Further speech recognition can be implemented for particular person. However for extended security concept of one time password can be introduced to make IOT home automation smart.

References

[1] "Open Home Automation Bus",[Online], Website,Dec 2016, *http* : *//www.openhab.org/*

[2] "PubNub:Realtime Apps Made Simple", [Online], Website,Dec 2016, *https* : *//www.pubnub.com/*

[3] "BYET Internet Services", [Online], Website,Dec 2016, *https* : *//byet.host/*

[4] "Jasper: Control everything with your voice", [Online], Website,Dec 2016, *https* : *//jasperproject.github.io/*

[5] "RESTful API with Python", [Online], Website,Dec 2016, *https* : *//blog.miguelgrinberg.com/post/designing−a−restful−api−with−python−and − flask*

[6] "RESTful HTTP methods", [Online], Website,Dec 2016, *http* : *//www.restapitutorial.com/lessons/httpmethods.html*

[7] "online Mockup, wireframe and UI prototyping tool:moqups", [Online], Website,Dec 2016, *https* : *//moqups.com/*