**Web Browser Music Remote For OMXPLAYER**

**RASPBERRY PI**

**System Requirements**

1. Raspberry Pi
2. Micro SD Card
3. Memory Card Reader
4. Image of the Raspbian OS
5. SSH connection enabled
6. Samba Server(for copying files from local machine to remote host)
7. Python 2+
8. Flask
9. Apache 2
10. Paramiko (and its dependencies)

**Approach**

1. **Setup Raspberry Pi with Raspbian OS for headless use via SSH.**

**Installing Raspbian Operating System**

This can be done by following the steps mentioned [here](https://www.raspberrypi.org/documentation/installation/installing-images/).

**Setting up Static IP**

This can be done by setting the raspberry pi to work on a stating IP by configuring it in the ***/etc/networks/interfaces*** file. This can be done by following the link: <https://www.modmypi.com/blog/tutorial-how-to-give-your-raspberry-pi-a-static-ip-address>

Now we can SSH into our Raspberry Pi device from any system on the same network.

* If it has a Windows OS then install SSH client **PuTTy** for Windows. Just type in the IP and then the username and password.
* If it has a Debian Based OS then we can open the terminal and write: ***ssh pi@<IP>*** and then type in the password.

1. **Deployment of a Flask Application on Apache.**

This can be done by following the steps of the following tutorial:

<https://www.digitalocean.com/community/tutorials/how-to-deploy-a-flask-application-on-an-ubuntu-vps>

Note: All the configuration files have been provided in the repository. One must only copy the codes in the right places.

1. **Writing the Flask Application.**

Developing a Flask Application requires the reading of the flask documents.

<http://flask.pocoo.org/docs/0.12/>

The basic Flask App has 2 components: The HTML Component(Templates) and the Control Component(Python Functions).

Make buttons on the musicplayer.html template and write its controls. The controls are written in the view of sending commands to the raspberry pi’s terminal.

The above is done by a SSH client python library called **Paramiko.** Its usage if clearly mentioned in its documentation.

<http://www.paramiko.org>

Its usage in this application is limited to the following:

1. Making a SSH object
2. Establishing connection
3. Sending commands using function ***exec\_command***
4. **Scripts to control OMXPLAYER using PIPES.**

The ***bin***folder in the ***Scripts***folder should be copied to the ***/home/***directory of the Raspberry Pi device. This folder contains all the scripts to execute while controlling the OMXPlayer. These scripts need to control the running instance of the OMXPlayer thus they use a custom **PIPE** named ***omfifo****.* The pipe is a fifo data structure for tasks in Debian based OS. This helps in the targeted execution of the scripts.

After copying the ***bin*** folder add this folder into the path of your Raspberry Pi(export as environment variable) by using the ***export***command.

Then make all the scripts executable by using the ***chmod******+x***command.

Reference can be taken from this link: <http://subupi.blogspot.in/2012/10/piping-across-shell-sessions-to-control.html>

1. **Designing GUI for Web Pages**

The web pages can be designed using the templates feature of Flask. The custom CSS can be placed in the static folder.

1. **Uploading Files to the Raspberry Pi for Playing**

**Setting Up Samba Server**

This can be easily done by following the tutorials in the given link: <http://raspberrypihq.com/how-to-share-a-folder-with-a-windows-computer-from-a-raspberry-pi/>

**Accessing shared folder from Windows PC**

You can access the shader folder from any Windows PC connected to the network and drop your music files there by entering ***\\your.pi.ip***  in the file explorer.

**Accessing shared folder from Debian based OS**

You can access the shader folder from any Windows PC connected to the network and drop your music files there by clicking on the “**Connect to Network**” in the file explorer and typing in the ip of the Raspberry Pi device.

**Future Modifications**

* More scripts for OMXPlayer controls can be easily added for example seeking forward.
* UI of the web app can be improved.
* File upload option from browser can be added.

**Making contributions**

1. Fork the repository form here: [***https://github.com/hemangr8/OMXPlayerWebRemoteFlask***](https://github.com/hemangr8/OMXPlayerWebRemoteFlask)
2. Create Pull Requests.
3. Raise Issues.