

ACKNOWLEDGEMENT

Presentation, inspiration, and motivation have always played a key role in the success of any venture. We would like to express our deep and sincere thanks to the staffs who supported us to implement this project.

We would like to express our sincere thanks to **Dr. J. KANAKARAJ**, Head of the Department, Department of Electrical and Electronics Engineering, to encourage us to the highest peak and to provide the opportunity to prepare this project.

We express our sincere thanks to our Program Coordinator **Dr.A.SOUNDARARAJAN**, Professor, Department of Electrical and Electronics Engineering, PSG College of Technology Engineering and our Tutor **Mr.M.SELVAGOPINATH**, Assistant Professor, Department of Electrical and Electronics Engineering, PSG College of Technology Engineering for their constant support and everlasting enthusiasm made it possible to complete the project within the time.

We would also like to thank, **Ms.G.PRADEEPA**, Assistant professor, Department of Electrical and Electronics Engineering, for her valuable guidance and support throughout this project.

We would like to thank the Innovation Lab Committee Members **Dr.C.MAHESWARI**, **Mr.M.SELVAGOPINATH**, **Mr.A.VISWANATH** Assistant Professor, Department of Electrical and Electronics Engineering, PSG College of Technology, for guiding and supporting us throughout the project work.

We would like to express our sincere thanks to all teaching and non-teaching staff members in the Department of Electrical and Electronics Engineering, for their direct and indirect support.

CHAPTER-1

INTRODUCTION

- The aim is to create an innovative application using facial mirror.
- People will be standing in front of mirror to check themselves from beginning of the day to presenting ourselves for college, workspace, job interviews etc.
- So it is planned to insert some good stuffs on mirror which includes basic information like date and time, a healthy remainder, some good thoughts and facts.

1.1 Hardware:

Smart Mirror is developed to run on a Raspberry Pi. It might run on various types of hardware, but new versions will only be tested on a Raspberry Pi. *Electron*, the app wrapper around Magic Mirror, only supports the Raspberry Pi 2, 3 & 4. The Raspberry Pi 0/1 is currently not supported. If want to run this on a Raspberry Pi 1, use the server only feature and setup a full screen browser.

1.2 Block Diagram:

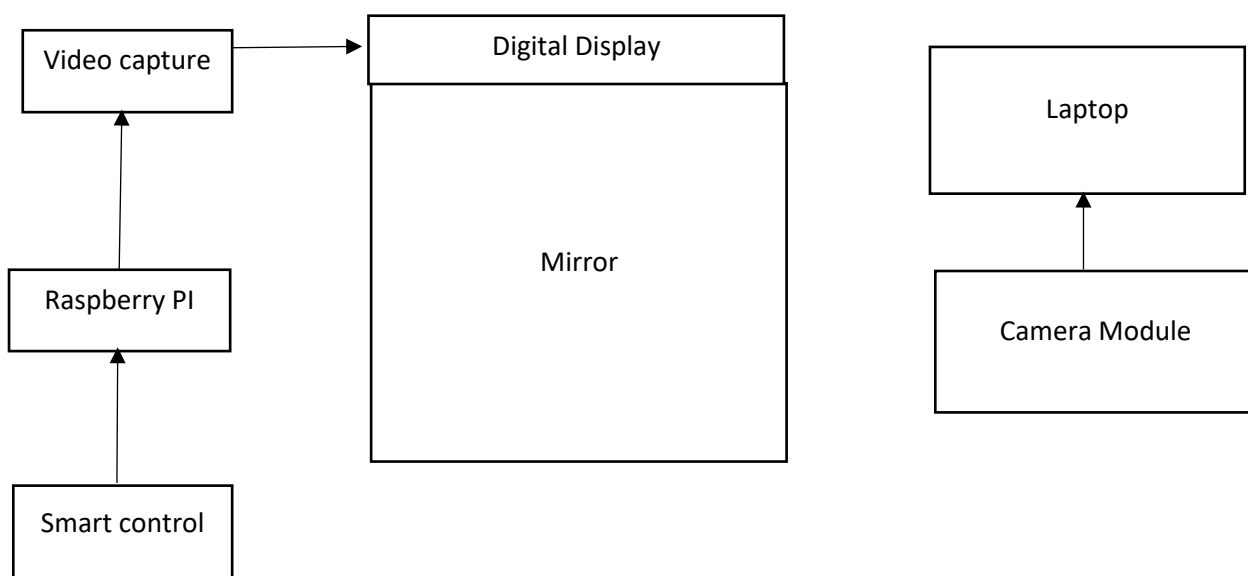


Figure no. 1.1 Block diagram

CHAPTER -2

INSTALLATION AND USAGE

Magic Mirror is an open-source modular smart mirror platform. With a growing list of installable modules, the magic mirror allows to convert the hallway or bathroom mirror into the personal assistant. Magic mirror is built by the creator of the original Magic Mirror with the incredible help of a growing community of contributors. Magic Mirror focuses on a modular plugin system and uses Electron as an application wrapper. So, no more web server or browser installs necessary.

The Magic Mirror can be installed manually or using automatic installers. At the start of 2020 the decision was made to remove the automatic installer from the Magic Mirror core repository and move it to a community maintained separate repository. Therefore, the only officially supported way of installation is by using a manual installation. Using external installation scripts is at own risk but can make the process a lot easier. Available automatic installers can be found under: alternative installation methods.

2.1 Operating System:

The Raspberry pi runs on Raspbian software. Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make the Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on Raspberry Pi. The initial build of over 35,000 Raspbian packages, optimized for best performance on the Raspberry Pi, was completed in June of 2012. However, Raspbian is still under active development with an emphasis on improving the stability and performance of as many Debian packages as possible

2.2 Node:

Node.js is an open-source server environment. Node.js is cross-platform and runs on Windows, Linux, Unix, Mac OS, etc. Node.js is a back-end JavaScript runtime

environment. Node.js runs on a JavaScript Engine (i.e. V8 engine) and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The functionality of running scripts server-side produces dynamic web page content before the page is sent to the user's web browser.

2.3 Manual Installation:

1. Download and install the latest *Node.js* version:
 - `curl -sL https://deb.nodesource.com/setup_16.x | sudo -E bash`
 - `sudo apt install -y node.js`
2. Clone the repository and check out the master branch: `git clone https://github.com/MichMich/MagicMirror`
3. Enter the repository: `cd MagicMirror/`
4. Install the application: `npm run install-mm`
5. Make a copy of the config sample file: `cp config/config.js.sample config/config.js`
6. Start the application: `npm run start`

Other Operating systems:

To get the Magic Mirror software running on Windows, two things in addition to the steps above to be done .Install dependencies in the vendor and font directories.

Power Shell:

1. `cd fonts; npm install cd`
2. `cd vendor; npm install cd`

Command Prompt:

1. `cd fonts && npm install cd`
2. `cd vendor && npm install cd`

Fixing the start script in the package.json file:

Note:

- npm start does **not** work via SSH. But use `DISPLAY=:0 nohup npm start` instead.
This starts the mirror on the remote display.
- Another method to debug on the Raspberry Pi, use `npm run start:dev` which will start MM with *Dev Tools* enabled.
- To access the toolbar menu when in mirror mode, hit ALT key.
- To toggle the (web) Developer Tools from mirror mode, use CTRL-SHIFT-I or ALT and select View.

Configuring the MagicMirror:

1. Copy `/home/pi/MagicMirror/config/config.js.sample` to `/home/pi/MagicMirror/config/config.js`.
2. `npm run config:check` in `/home/pi/MagicMirror`.

The following properties can be configured:

| Option | Description |
|-------------|--|
| port | The port on which the MagicMirror ² server will run on. The default value is 8080. |
| address | The interface ip address on which to accept connections. The default is localhost, which would prevent exposing the built-in webserver to machines on the local network. To expose it to other machines, use: 0.0.0.0. |
| ipWhitelist | The list of IPs from which allowed to access the MagicMirror ² . The default value is <code>["127.0.0.1", "::ffff:127.0.0.1", "::1"]</code> , which is from localhost only. Add yht IP when needed. Specify the IP ranges with subnet masks (<code>["127.0.0.1", "127.0.0.1/24"]</code>) or directly with |

| Option | Description |
|-------------------|---|
| | (["127.0.0.1", ["192.168.0.1", "192.168.0.100"]]). Set [] to allow all IP addresses. |
| Zoom | This allows to scale the mirror contents with a given zoom factor. The default value is 1.0 |
| language | The language of the interface. (Note: Not all elements will be localized.) Possible values are en, nl, ru, fr, etc., but the default value is en. |
| time Format | The form of time notation that will be used. Possible values are 12 or 24. The default is 24. |
| units | The units that will be used in the default weather modules. Possible values are metric or imperial. The default is metric. |
| modules | An array of active modules. The array must contain objects. See module configuration for more information. |
| electron Options | An optional array of Electron (browser) options. This allows configuration of e.g. the browser screen size and position (example: electron Options: { full screen: false, width: 800, height: 600 }). Kiosk mode can be enabled by setting kiosk: true, autoHideMenuBar: false and full screen: false. More options can be found here. |
| electron Switches | An optional array of Electron switches. This allows configuration of electron app itself. These properties will not affect the server only mode. Usually normal MM users don't need this property, if a hard-core hacker means, this is needed to handle Electron itself over Magic Mirror provides. More options can be found here (Not all available switches are described there.) |

| Option | Description |
|-----------|---|
| | example:electronSwitches:["enable-transparent-visuals", "disable-gpu"]; |
| customCss | The path of the custom.css stylesheet .The default is css/custom.css. |

Table no. 2.1 Configuration of MagicMirror

CHAPTER-3

Modules

The following modules are installed by default.

- Clock
- Calendar
- Weather
- News Feed
- Smart Touch
- Remote Control Module
- Facts Module
- Alert Module
- Word Genius Module

3.1 Clock Module:

The clock module is one of the default modules of the MagicMirror. This module displays the current date and time. The information will be updated realtime..



Figure no. 3.1 Clock module

Configuration options

The following properties can be configured:

Notification:

The clock makes use of the built-in Notification Mechanism to relay notifications to all modules.

Current notifications are:

| Notification | Description |
|--------------|--|
| CLOCK_SECOND | A second has elapsed. Parameter: second value |
| CLOCK_MINUTE | A minute has elapsed Parameter: minute value |

Table no.3.1 Clock module

3.2 Calendar Module:

The calendar module is one of the default modules of the Magicmirror. This module displays events from a public.ical calendar. It can combine multiple calendars.



Figure no. 3.2 Calendar module

3.3 Weather Module:

This module will be configurable to be used as a current weather view, or to show the forecast. This way the module can be used twice to fulfill both purposes. The biggest change is the use of weather providers. This way we are not bound to one API source. And users can choose which API they want to use as their source. The module is in a very early stage, and needs a lot of work. It's API isn't set in stone, so keep that in mind when to contribute.

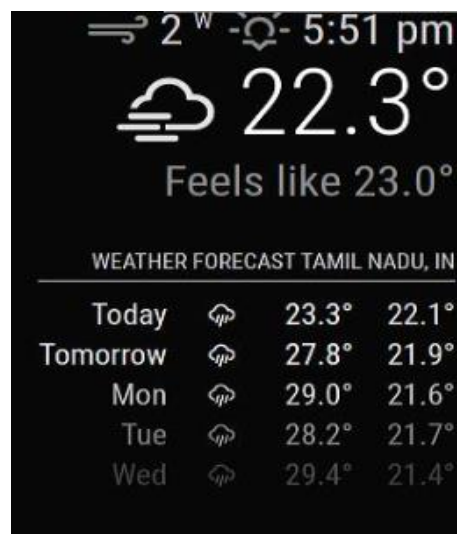


Figure no. 3.3 Weather Module

Current weather options:

| Option | Description |
|-------------|--|
| onlyTemp | Show only current Temperature and weather icon without windspeed, sunset, sunrise time and feels like. Possible values: true or false Default value: false |
| useBeaufort | Pick between using the Beaufort scale for wind speed or using the default units. Possible values: true or false Default value: true |

| Option | Description |
|--------------------------|---|
| showWindDirection | Show the wind direction next to the wind speed. Possible values: true or false Default value: true |
| showWindDirectionAsArrow | Show the wind direction as an arrow instead of abbreviation Possible values: true or false Default value: false |
| showHumidity | Show the current humidity Possible values: true or false Default value: false |
| showIndoorTemperature | If another module that emits the INDOOR_TEMPERATURE notification, the indoor temperature will be displayed Default value: false |
| showIndoorHumidity | If the another module that emits the INDOOR_HUMIDITY notification, the indoor humidity will be displayed Default value: false |
| showFeelsLike | Shows the Feels like temperature weather. Possible values: true or false Default value: true |
| showSun | Shows Sunrise and Sunset time. Possible values: true or false Default value: true |

Table no. 3.2 Weather Module (current)

3.4 News feed module:

The newsfeed module is one of the default modules of the Magic Mirror. This module displays news headlines based on an RSS feed. Scrolling through news headlines happens time-based (update Interval), but can also be controlled by sending news feed specific notifications to the module.



Figure no. 3.4 News feed module

3.5 Smart-Touch:

This module allows to transform the magic mirror into a touchscreen interactive experience. This includes:

- Standby Mode - Hide all magic mirror modules and convert back to a normal mirror by pressing the cycle button.
- Side Menu - Menu widget includes a *shutdown* and *restart* of Rpi to safely power down the mirror without unplugging and forced shutdowns (preventing SD card corruptions).



Figure no. 3.5 Smart touch 1

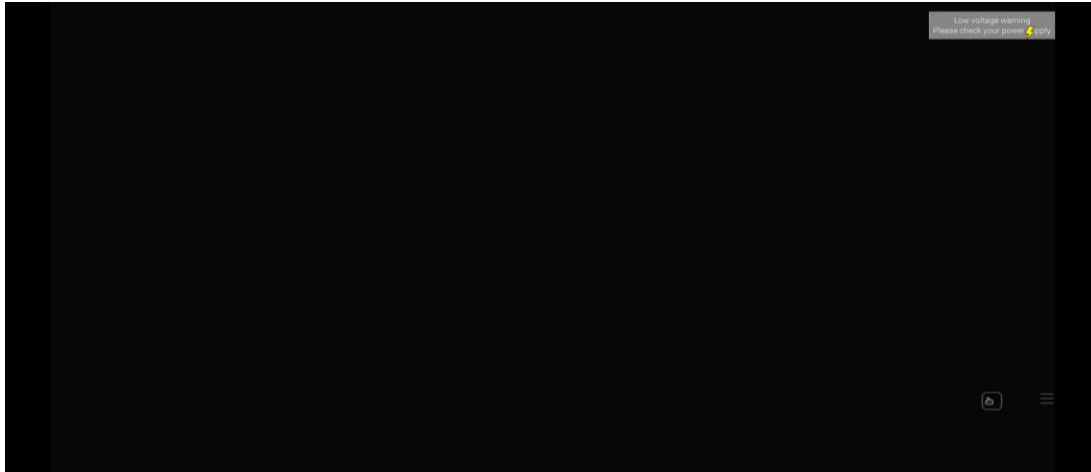


Figure no.3.6 Smart touch 2

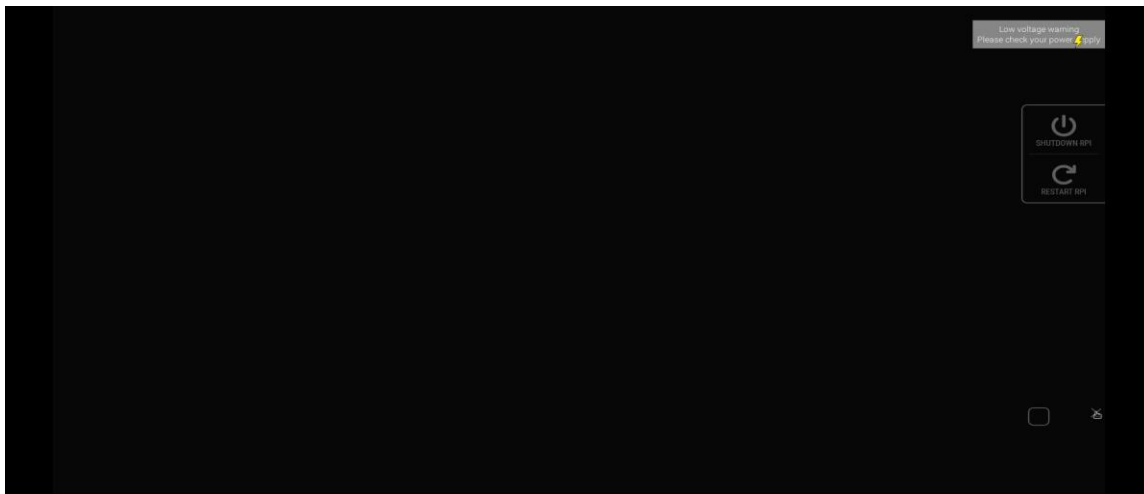


Figure no. 3.7 Smart touch 3

3.6 Remote Control Model:

Remote control allows to control the digital mirror from the smart phone which will be the very great feature and it takes the smart mirror to whole another level. There are several lists of actions that can be performed with this feature which are listed below.

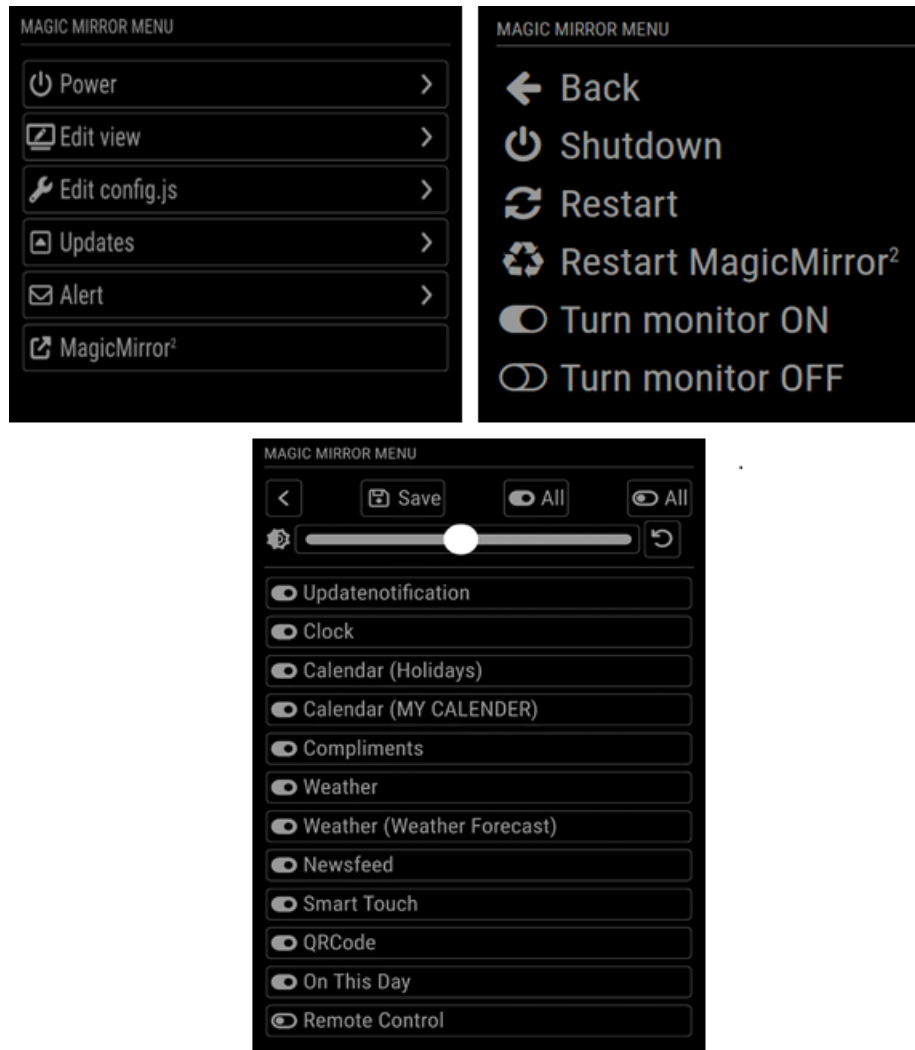


Figure no. 3.8 Remote Control Model

List of actions

System Control:

| Action | Description |
|---------------|--|
| SHUTDOWN | Shutdown your RaspberryPi |
| REBOOT | Restart your RaspberryPi |
| MONITORON | Switch your display on. Also sends a <code>"USER_PRESENCE": true</code> notification. |
| MONITOROFF | Switch your display off. Also sends a <code>"USER_PRESENCE": false</code> notification. |
| MONITORTOGGLE | Toggle the display on or off (with respective <code>"USER_PRESENCE"</code> notification. |
| MONITORSTATUS | Report back the monitor status (on or off) |

Figure no. 3.9 Remote control model

MagicMirror Control:

| Action | Description |
|------------|--|
| RESTART | Restart your MagicMirror |
| REFRESH | Refresh mirror page |
| UPDATE | Update MagicMirror and any of it's modules |
| SAVE | Save the current configuration (show and hide status of modules, and brightness), will be applied after the mirror starts |
| BRIGHTNESS | Change mirror brightness, with the new value specified by <code>value</code> . <code>100</code> equals the default, possible range is between <code>10</code> and <code>200</code> . |

Figure no. 3.10 Remote control module1

MagicMirror Electron Browser Window Control:

| Action | Description |
|------------------|------------------------------------|
| MINIMIZE | Minimize the browser window. |
| TOGGLEFULLSCREEN | Toggle fullscreen mode on and off. |
| DEVTOOLS | Open the DevTools console window. |

Figure no. 3.11 Remote control module 2

Module Control:

| Action | Description |
|-------------|--|
| HIDE | Hide a module, with the name (or identifier--see <code>MODULE_DATA</code> action) specified by <code>module</code> in the payload. You can also send <code>module: "all"</code> to hide all modules. |
| SHOW | Show a module (see above for how to specify which one). |
| TOGGLE | Toggle a module's visibility (see above for how to specify which one). |
| FORCE | Force a module to show (see above for how to specify which one). |
| MODULE_DATA | Returns a JSON format of the data displayed in the UI, including all valid identifiers for the <code>HIDE</code> and <code>SHOW</code> action. |

Figure no. 3.12 Remote control module 3

Alerts and Notifications:

| Action | Description |
|---------------|---|
| SHOW_ALERT | Show Default Alert/Notification |
| HIDE_ALERT | Hide Default Alert/Notification |
| USER_PRESENCE | Will send a notification "USER_PRESENCE" = true or false (according to "value" to all other modules. See examples above |
| NOTIFICATION | To send a notification to all modules, see the example in the API README |
| DELAYED | Send any of the above nested inside a "DELAYED" call to delay the action. Default is 10s. See Delayed Actions below. |

Figure no: 3.13 Remote control module 4

3.7 Facts Module:

This module provides day to day facts which are available in various types. The types are listed in the table and it will be displayed in smart mirror as shown in fig no: 3.9

Configuration options

| Options | Description |
|-----------------------------|--|
| <code>updateInterval</code> | Time the fact is changed in seconds |
| <code>fadeSpeed</code> | Fact text fadeout in seconds. |
| <code>category</code> | Default is <code>random</code> , but you can choose specific categories such as <code>art</code> , <code>food</code> , <code>health</code> , <code>history</code> , <code>language</code> , <code>nature</code> , <code>nerd</code> , <code>space</code> or <code>tips</code> . Feel free to add your own as well! |

Figure no. 3.14 Facts Module

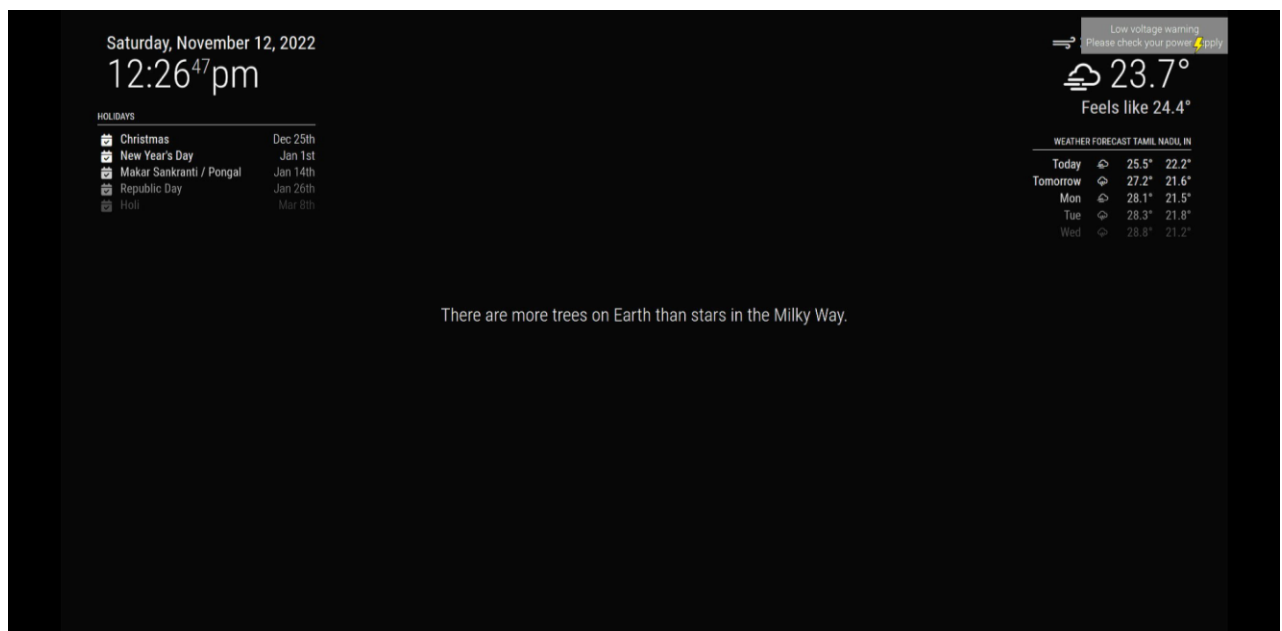


Figure no. 3.15 Facts Module

3.8 Alert:

The alert module is one of the default modules of the MagicMirror. This module displays notifications from other modules.

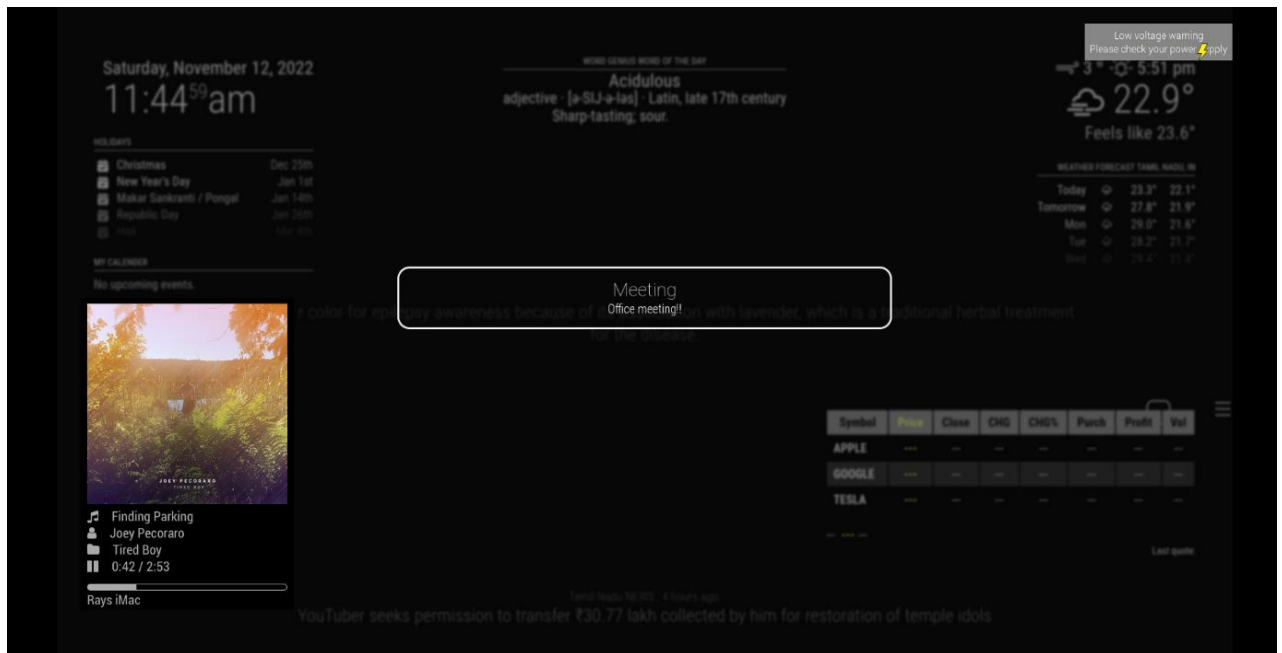


Figure no.3.16 Alert 1

MAGIC MIRROR MENU

<

Send

Hide

Type:

Alert

Title:

Meeting

Message:

Office meeting!!

Seconds:

4

Figure no.3.17 Alert 2

3.9 Word genius module:



Figure no. 3.18 Word genius module

3.10 Final Output:

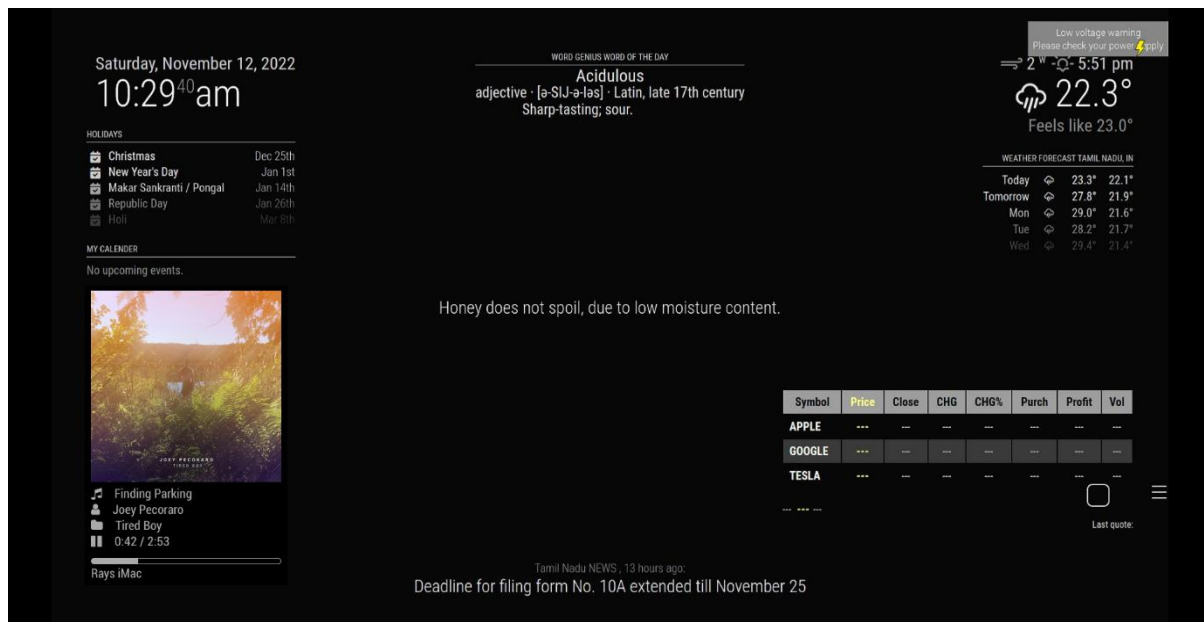


Figure no. 3.19 Final Output

Chapter 4

Conclusion and future works:

Thus, in this innovation practices designed a smart mirror which is a mirror with a display in it which shows various facts, messages, news, meetings, date, time, weather forecast with smart control which will be advanced in a way in which mood of the person will be recognized and display will change according to the mood. Thus, it will be very much useful in maintaining good mental health. Smart mirror will be using mood recognition using web cam to capture the live feed and recognize the mood and alternating display according to it which will mark a great significance regards to mental health. Now a days lot of people are under stress and depressions. This will help to analyse the mood and making display to look in the way that it will make him better if he is in a bad state. The mood recognition is on progress and it can recognize the mood in a good scale.

REFERENCES:

- 1] "Internet Of Things Based Smart Mirror: A Literature Review" by Dabiah .A.Albonaeneen and Dalia Alsaffar,Alyah Alateeq ,Computer Science Department ,Iman Abdulrahman Bin Faisal University.
- 2] "A computational model for mood recognition" by Christina Katsimerou, Judith A Redi, Ingrid Heynderickx, International Conference on User Modeling, Adaptation, and Personalization, 122-133, 2014.
- 3] "Raspberry Pi Cookbook Software and Hardware Problems & Solutions, Third Edition" by Simon monk.
- 4] "Design of smart mirror based on Raspberry Pi" by Yong Sun, Liqing Geng, Ke Dan 2018, International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 77-80, 2018.
- 5] Getting started with raspberry Pi by Matt Richardson, Shawn Wallace" O'Reilly Media, Inc.", 2012.
- 6] "Smart mirror for smart life" by Muhammad Mu'izzudeen Yusri, Shahreen Kasim, Rohayanti Hassan, Zubaile Abdullah, Husni Ruslai, Kamaruzzaman Jahidin, Mohammad Syafwan Arshad, 2017 6th ICT International Student Project Conference (ICT-ISPC), 1-5, 2017.
- 7] "Mood recognition" by Christina Katsimerou, Judith A Redi, Ingrid Heynderickx, International Conference on User Modeling, Adaptation, and Personalization, 122-133, 2014.
- 8] "Mood recognition across the ages" by Leonard Brosgole, Jaye Weisman, International Journal of Neuroscience 82 (3-4), 169-189, 1995

