

Design of Smart Mirror as a College Notice Board using IoT

¹Y.V.Sneha Prabha, ²Asha Namdhari, ³Snigdha Pannir

^{1, 2, 3}Student

^{1, 2, 3}Department of Electronics and communication Engineering

^{1, 2, 3}G. Naryanamma Institute of Technology & Science, Hyderabad, India

Abstract—

A smart mirror is a device that functions both as a reflective mirror and a computer display. It is designed with the help of Raspberry Pi v4 as its primary controller. It is currently programmed to display weather forecast, calendar, date, day, a motivational compliment and news updates. We plan on updating the current design to display college department notifications, updates and general notices. The hardware used will remain the same; the major changes will be done in the software components. We plan on using the college website to obtain the required data for the mirror.

Designing a ‘Smart Mirror’ in this fashion is something we can emulate to a real-world industry scenario as in industries most programs are not made from scratch. As a team, we chose to continue on an already working code of a smart mirror made for our mini project. In addition, we are also thinking of this project as something that would be required by the consumer. Using this mind set, we sought to create something easy and convenient for a user that has little to no computer experience. We plan on using raspberry pi modules which are fairly common to run personal and specific tasks as well as leveraging our use of Python, personal security with databases, local servers, and JavaScript to create a fluid simple system for people to use. This will enable us to create a personal, internet based, simple, secure and on-board program for maximal user enjoyment, convenience and information.

IndexTerms—Raspberry Pi, Web Application, Internet of Things, Display, Modules.

I. INTRODUCTION

Modern living has evolved era after era leading to an increased preference for comfort by mankind. Most technological advancements nowadays can be attributed to making one’s life easier. This leads to the notion of “Automated living” which goes hand in hand with Internet of Things, Machine Learning, Artificial Intelligence and several other emerging technologies in the previous decade. Artificial Intelligence, in particular, has seen new developments in the early 21st century.

Living in a globalized society has led to a pressing need for faster communication and connectivity across people. Whether it is through the television or internet, people need to be informed and in touch with the current affairs happening around the world.

The concept of Internet of Things means interconnection via the internet leading to computing devices embedded in everyday objects, enabling them to send and receive data. IOT with its enormous growth widens its applications to the living environment of the people by changing a home to a smarter home. A “Smart home” is a connected home that connects all type of digital devices to communicate each other through the internet.

This project stems from the idea that we all look at the mirror when we go out, so why can’t the mirror be a smarter version of itself. And the common approach to building a smart mirror is to use a high-quality one-way glass, an LCD display, a frame to hold the glass and display and a web browser programmed to provide the software features and drive the display.

II. RELATED WORK

"A Mobile-Programmable Smart mirror for Ambient IoT Environment"[1] published at 5th International Conference on Future Internet of Things and Cloud Workshops in 2017 describes the design and development of Interactive Smart mirror that offers simplified and customizable services to the home environment. The Smart mirror can also be used to control home appliances with very less human intervention using a mobile application. This paper helped in understanding the logistics better though its discussed features were not implemented in our project. For controlling home appliances, the mobile needs to be paired with the smart mirror successfully.

"Smart Mirror for Smart Life"[2] published at IEEE Conference publication also describes about the monitoring and controlling of home-based devices with the mirror. To ease the human tasks and develop interaction between people and system, the mirror system uses Sonus technology as a medium. The Smart mirror takes voice commands as input to give response and Sonus is a speech to text library that can quickly and easily add VUI (Voice User Interface) to any hardware or software.

The "Implementation and Customization of a Smart Mirror through a Facial recognition Authentication and a Personalized News Recommendation Algorithm"[3] published at 13th International Conference on Signal- Image Technology & Internet- Based Systems (SITIS) in 2017 includes the usage of smart mirror as a full-time security camera which opened us to newer possibilities. The daily news recommendation predictive model is implemented through the facial recognition algorithm.

The "SmiWork: An Interactive Smart Mirror Platform for Workplace Health Promotion"[4] describes about a multi-user Smart mirror that promotes wellness and healthier lifestyle.

Microsoft’s Magic Mirror: Microsoft demoed their first Magic Mirror prototype in 2016, which is powered by a Hosted Web App on Windows 10 IoT Core installed on a Raspberry Pie 3. This project was built with the concept of personalized experience that recognizes users using Windows Hello technology powered by Microsoft’s Cognitive Services. Microsoft has open – sourced the web app in their GitHub repository which can be used by anyone to build their own smart mirror. The mirror is built to be useful to

a person getting ready in the morning who is likely to be on a time crunch and wants to be well prepared for the day and is interested in weather and traffic updates.

CMUcam5 Pixy: This tiny image sensor is a partnership between Carnegie Mellon Robotics Institute and Charmed Labs that started as a Kickstarter campaign. It is small, easy to use and programmable sensor that outputs what it detects 50 times every second and supports programming languages such as C++ and Python. Running on NXP processor that clocks at 204 MHz and has dual core has 264K bytes of RAM and memory of 1M bytes. Output data can be captured using several buses such as UART serial, SPI, I2C, USB, analog, and digital.

Ekko Smart Mirror: This project is created by a company that produces fashionable furniture. Other than displaying weather and news feeds in real time this smart mirror can also play music and videos. The mirror communicated to the user's smartphone using an app that has to be installed to give the user a personalized experience. Mirror control is achieved using gestures which are read by sensors on each side.

Apple Mirror – Rafael Dymek: A fully functional smart touchscreen mirror based on Apple's iOS 10 that mimics iPhone display and can launch apps that are preferred by the user. Apps cannot be launched in full screen due to rescaling issues as for now. The Apple mirror goes to sleep after 45 seconds of inactivity and can be turned on by tapping the screen. This product has not been commercialized yet.

III. SYSTEM OVERVIEW

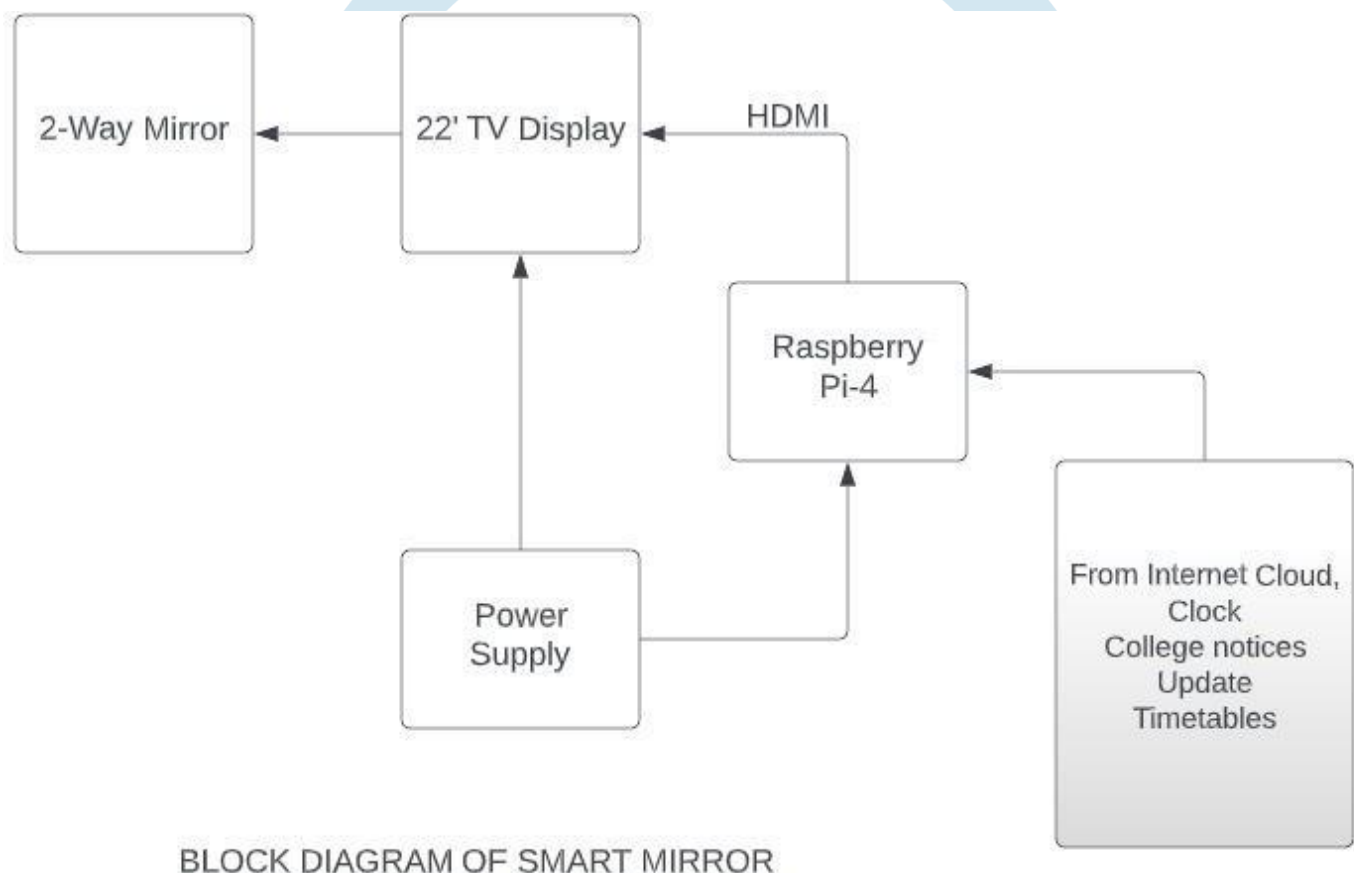


Fig 1. Block Diagram for Proposed System

Figure 1 depicts a block diagram of the proposed system. The Monitor is the primary display that the device uses and it is the only end of interaction for the user. The user remains unaware of the rest of the functionality and therefore interacts primarily with the monitor only. A see-through mirror is placed on top of the monitor to turn it into a reflective surface. Apart from being inexpensive, the mirror may also be replaced in the event of any damage. This can be substituted with a reflective glass or a suitable acrylic sheet as well.

Raspberry Pi: All of the mirror's functionality, its connection to a wireless network, and the programming of its sensors are coded onto the Pi Board. The Pi acts as the brain of the architecture and is the primary interface between the user and the rest of the architecture. The addition of newer modules onto the screen as well as the positioning and other formatting is done by programming using the Pi.

IV. COMPONENTS

A. Raspberry Pi 4

For our project we bought a 4GB Raspberry Pi accompanied with a 64GB SD card for downloading the OS onto it. The Raspberry can be treated as a microcomputer board and has a myriad of applications in real life. Its multi peripherals support building networks with all sort of devices and its OS is also comfortable to use. The OS, being primarily coded in Python is easier to learn than complex OS's like Linux or Unix. The technical features of Raspberry pi are as follows:

- A high-performance 64-bit quad-core processor
- Dual display support with resolutions up to 4K via a pair of micro-HDMI ports
- Hardware video decoding up to 4Kp60
- 4 GB of RAM
- A connection to the dual-band wireless local area network 2.4/5.0 GHz
- Bluetooth 5.0 / Gigabit Ethernet / USB 3.0 / PoE features
- SoC Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit at 1.5GHz
- SDRAM 4 GB LPDDR4-2400
- Wireless LAN 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac, Bluetooth 5.0, BLE
- True Gigabit Ethernet
- 2 USB 3.0 ports, 2 USB 2.0 ports
- Fully backward compatible 40-pin GPIO connector
- 2 HDMI micro ports supporting video resolution up to 4K 60Hz
- 2-way MIPI DSI DSI/CSI ports for camera and display
- Stereo audio output and composite video port, 4-pole
- Slot for Micro SD card, for operating system and data storage
- Requires 5.1V, 3A power supply via USB-C or GPIO
- PoE (Power over Ethernet) enabled (requires PoE HAT)

Keeping this particular project in mind, it should be stated that Raspberry is used almost as a CPU with the SD card being the memory. So, for input and output, we require the need of a display/monitor and a keyboard and mouse.

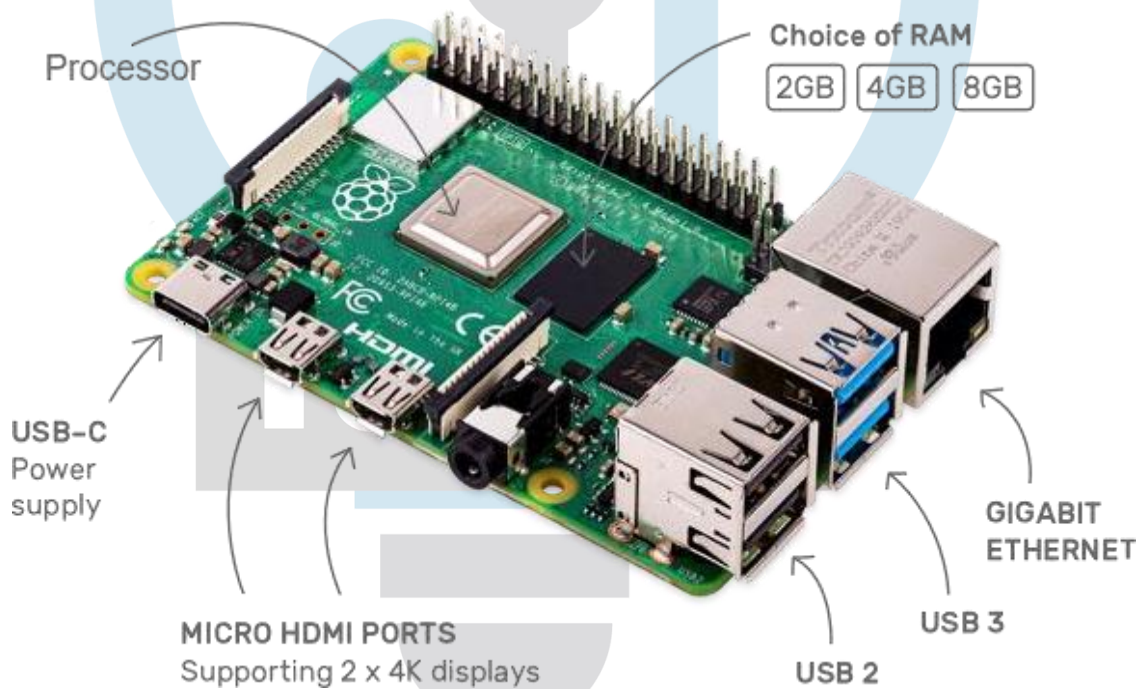


Fig 2. Raspberry Pi 4

B. Raspberry Pi adapter

A Raspberry Pi adapter is used for establishing the power connection to the Raspberry Pi. The Raspberry Pi's power port is USB-C. We can also use a USB to Micro USB cord for supplying power from the display itself.



Fig 3. Raspberry Pi adapter

C. Keyboard and Mouse

Keyboard and mouse are used for writing code to the OS. They can be wired or wireless but must be connected to the Raspberry by means of the USB ports supplied for this very reason. We used a wireless keyboard and mouse with its dongle attached to the Raspberry's USB.

D. HDMI to Micro-HDMI cable

This cable connects the display to the Raspberry. We specifically use the HDMI port indicating Output on the Raspberry which is also accompanied by HDMI for Input.



Fig 4. HDMI to Micro HDMI cable

E. Display screen

This is an ambiguous decision as this project can be done using any display including but not limited to monitors, TVs, Tablets or even displays specifically designed for Raspberry Pi. It depends on the number and availability of suitable ports on the device and budget constraints. We decided to work with a TV which has a HDMI ports and of sufficient size, thus working suitably for our purposes. It goes without saying that we need Wi-Fi connection along with a power source for the display.

V. HARDWARE IMPLEMENTATION

Discounting the aforementioned components and their connections, we still require a suitable mirror and a frame to hold this entire structure together. For the mirror, what we need to understand is that all mirrors are partially see through and partially reflective. However, this see-through nature is blocked by painting the back of the mirrors. So, our requirement is a see-through mirror, a reflective glass or a two-way mirror with a capacity of 70% reflection and 30% transparency. The two-way mirror is what gives the mirror its real identity. It's really a magic mirror as it has a reflective surface at one side and also, it's transparent for light with good intensity. The mirror stays at the front where the user can watch him/her in the mirror at the same time that allows the light from the monitor to pass through it and make available the UI. A wooden frame structure is also needed to hold the mirror and display in place without moving.



Fig 5. Hardware implementation

VI. INITIAL SETUP

A. Downloading the Raspbian OS

Raspbian OS (formerly Raspbian) is a Debian-based operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry Pi family of compact single-board computers. Writing the Raspbian OS onto the SD card is done by either using a specific etcher or the imager provided in the Raspberry Pi website.

We flashed the Raspbian OS latest version onto a SD card, directly downloaded from the Raspberry imager provided in their website. After that, we establish the hardware connections by connecting Raspberry Pi to a keyboard, mouse and our display screen. We used a wireless keyboard and mouse connected using USB while the screen is connected using HDMI to micro HDMI cable.

B. Booting the Raspberry Pi

Raspberry Pi doesn't have any specific power switch. It directly boots as soon as we open our HDMI display. On first use, we specify the user details like location, language and time zone. This is followed by changing the default username and password and connecting to a Wi-Fi network. We open the command terminal provided on the top of Raspberry home screen.



Fig 6. Raspberry Pi welcome screen

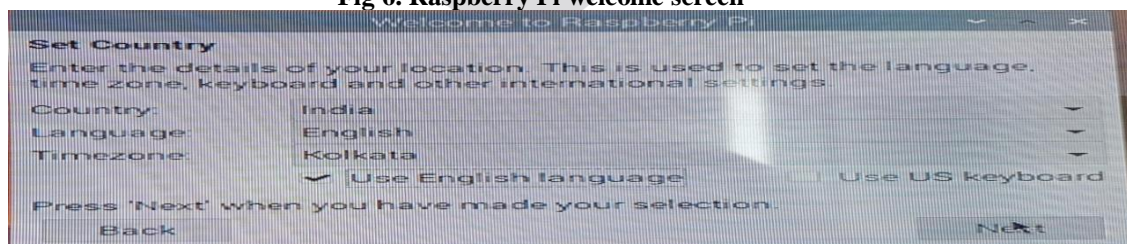


Fig7. Configuring Raspberry Pi

This is followed by installing latest updates and upgrades of the Raspberry. It is imperative that the latest updates are done to prevent non-functional dependencies. Non-functional dependencies occur when software can't find a suitable version of another module in the Pi. We then installed Node.js and MagicMirror2 github repository with necessary fonts and vendors. Node.js is an open source, cross-platform, back end JavaScript runtime environment that executes JavaScript code outside a web browser. This entire process of installing different packages and software takes approximately 4 to 5 hours.

As we plan on viewing the mirror in portrait mode, it is necessary that the default setting of the display be portrait as well. The 'BIOS' settings of the Raspberry Pi are stored in the boot partition of the system. This partition contains a `configure.txt` file with all the settings. The following `custom.css` file is created for enabling 90 degrees rotation.

C. Module modifications using JS

The base display which is seen on the first run of the software gives us a dark screen with incompatible data. We modify the module commands to customize the magic mirror. We can update the calendar by using a suitable digital calendar in iCal format. Our aim is to design a display which can be used as a college notice board. Thus, we wanted to include the following information. The seating arrangement of ECE department, the timetables of all different years in ECE, college notices and notifications and a welcome screen.

As we don't want the mirror to show any transitions, arrangements were done such that the mirror directly opens on to the browser in Kiosk mode. The following lines of code help us achieve that. We also disable the screensaver and mouse pointer by using the software `unclutter` and `xscreensaver` respectively.

VII. MODULE DESIGN

A. Using MMM-HTMLSnippet

We managed to obtain a module that displays data using HTML5 format. This was very convenient as HTML5 has several options for styling. The HTML snippet is a module in which the module accepts part of a HTML code which is assigned to default variables in JavaScript and is executed accordingly. We wrote the code for the seating arrangement in HTML using a `<table>` tag. The module converts the given HTML data into an "iframe" which is then tethered to the main DOM of the MagicMirror webpage. Following that, we styled the table with suitable positioning, size, formatting and alignment. This approach turned out to be more promising with improved results. The class name, room number and floor number are now displayed in an elegant table.

B. Using MMM-Pages

To address the issue of limited screen space, we used the module "MMM-Pages". This module separates each module into array elements and displays it as an array element one after the another. This code is implemented in JavaScript. It works on the principle of modulo, the total number of screens is considered to be "N" and a counter is incremented of value "I". The value of I is used to calculate the remainder with the value N. This gives us recurring values from 0 to N-1. This tells the module which screen or array element is to be displayed.

C. Using MMM-Airtable

We use a platform called Airtable for displaying weekly timetables of 2nd years and M.Tech students of the ECE department. Airtable provides data from its tables using API service which can be linked to the MagicMirror repository using the module aptly titled "MMM-Airtable". Airtable is a spreadsheet-database hybrid, with the features of a database but applied to a spreadsheet. The fields in an Airtable table are similar to cells in a spreadsheet, but have types such as 'checkbox', 'phone number', and 'drop-down list', and can reference file attachments like images. Users can create a database, set up column types, add records, link tables to one another, collaborate, sort records and publish views to external websites.

D. Using MMM-WeeklySchedule

Due to the working of the MMM-Pages module, there is a limitation on how we can use modules. For instance, a single module can't be called twice as the `node_helper.js` will not be able to perceive which module is being mentioned the second time. Another constraint which is formed is the fact that the pages module can display data one module at a time. As this would mean displaying the weekly timetables of 10 sections on one screen, we decided to instead to split up the data. For the remaining 6 sections, the 3rd and 4th years, we went with the idea of displaying daily timetables. This is achieved using the module "MMM-WeeklySchedule". A daily timetable displays the classes the students have on that day and changes them according to the day of the week. We have programmed this module to change the day at 1500 hours or 3pm. This means, after 3pm the timetable of the next day will be displayed. The timetable data is given in JSON data format and is locally established on the device without any cloud platform intervention.

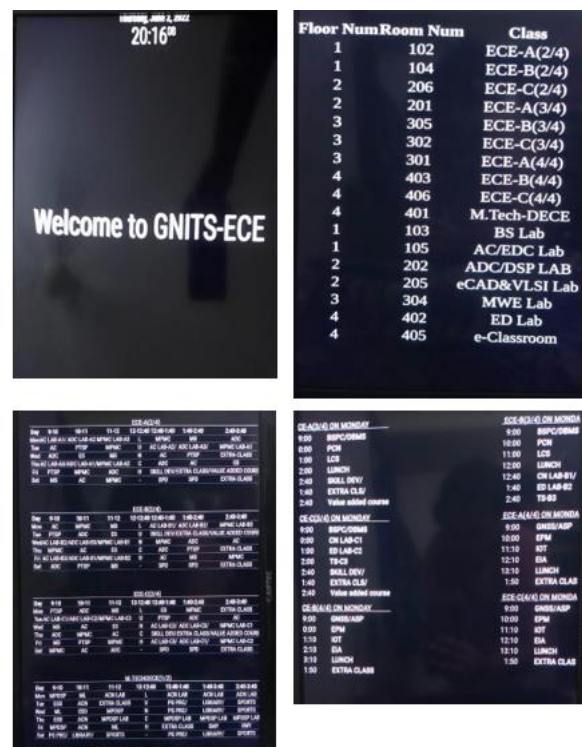


Fig 8. Output

VIII. CONCLUSION

A smart mirror is a device that functions both as a reflective mirror and a display. We programmed the Raspberry Pi v4 to currently display the seating arrangement of ECE department and we are working on the college notice board display. Web scraping is used extensively for obtaining data needed from the internet in this project.

This project also taught us many topics and introduced us to fields and studies we have never heard of before. Working with already written code to design our specifications is a new experience, one which will be very useful to have after joining the industry as most projects are typically coded across different teams. We were introduced to several unknown software, techniques and practices as we researched for the project which itself is a gift beyond measure.

IX. FUTURE SCOPE

On a personal level, we plan on including the Google assistant as a feature of the mirror. As the data to be displayed increases, it is imperative that we find a way of displaying multiple screens and controlling said screens as well. The highest target we plan on achieving is to enable notifications from the mirror to a phone. This project can be further improvised in several ways, by incorporating various digital assistants like Alexa or Siri, or by adding different streaming platforms. We can also try to use IR screens to explore the possibility of building a Smart mirror with touchscreen capabilities. The world is an oyster for this project whose possibilities are endless.

REFERENCES

- [1] M. Ghazal, T. Al Hadithy, Y. Al Khalil, M. Akmal and H. Hajjdiab, "A Mobile-Programmable Smart Mirror for Ambient IoTEnvironments," 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2017, pp. 240-245, doi: 10.1109/FiCloudW.2017.106.
- [2] M. M. Yusri et al., "Smart mirror for smart life," 2017 6th ICT International Student Project Conference (ICT-ISPC), 2017, pp. 1-5, doi: 10.1109/ICT-ISPC.2017.8075339.
- [3] I. C. A. García, E. R. L. Salmón, R. V. Riega and A. B. Padilla, "Implementation and Customization of a Smart Mirror through a Facial Recognition Authentication and a Personalized News Recommendation Algorithm," 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2017, pp. 35-39, doi: 10.1109/SITIS.2017.17.
- [4] O. Gomez-Carmona and D. Casado-Mansilla, "SmiWork: An interactive smart mirror platform for workplace health promotion," 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), 2017, pp. 1-6.
- [5] B. Cvetkoska, N. Marina, D. C. Bogatinoska and Z. Mitreski, "Smart mirror E-health assistant Posture analyze algorithmproposed model for upright posture," IEEE EUROCON 2017 -17th International Conference on Smart Technologies, Ohrid, 2017, pp. 507-512
- [6] D. Gold, D. Sollinger and Indratmo, "SmartReflect: A modular smart mirror application platform," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2016, pp. 1-7

- [7] Mohammed Ghazal, Tara al Hadithy, Yyasmina al Khalil, Muhammad Akmal and Hassan Hajjdiab, " a Mobile-programmable smart mirror for ambient IoT environments", in 5th international conference on future internet of things and cloud workshops, 2017.
- [8]. Muhammed Mu'izzudeen, YusriShahreem Kasim, Rohayanti Hassan, Zubaile Abdullah HusniRuslai, KamaruzzamanJahi din, Mohammad Syafwan Arshad, " Smart Mirror for Smart Life", in IEEE Conference publication, 2017.
- [9]. Oihane Gomez-Carmona, Diego Casado-Mansilla, "SmiWork: An Interactive Smart Mirror Platform for Workplace Health Promotion", 2017.
- [10]. Piyush Maheshwari, Maninder Jeet Kaur, Sarthak Anand, "Smart Mirror: A Reflective Interface to Maximize Productivity", International Journal of Computer Applications (0975 – 8887), Year: May-2017.

