International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 5, Sept-Oct-2025, ISSN (Online): 2395-566X

Smart Mirror

Ms. Sonia[1], Mr. Aditya Goel[2], Mr. Aneek Kumar[3], Mr. Kushagra[4], Ms. Chandni Kumari[5]
[1]Assitant Professor, [2]studnet (Ai-Ds), [3]student (Ai-Ml),[4]student (Ai-Ml),[5]student (Ai-Ds)
HMR Institute Of Technology And Management, GGSIPU, Delhi

Abstract- This paper presented the development of a Smart Mirror, a device that integrates real- time information display with everyday utility. Designed by college students, the Smart Mirror was built entirely using Python libraries and provided functionalities such as news feeds, weather updates, calendar events, reminders, and basic time and date display. The project emphasized software development using Python and various APIs to create an interactive and user- friendly interface. The Smart Mirror aims to enhance daily routines by reducing the need for multiple devices while maintaining simplicity and efficiency.

Index Terms–– Smart Mirror, Python libraries, weather updates, calendar events.

I. INTRODUCTION

The Smart Mirror is an innovative application of Internet of Things (IoT) technology, combining a traditional mirror with digital functionality. It uses a two-way mirror with an LCD screen behind it to display information while retaining its reflective properties. This project addresses the need for a seamless integration of technology into daily life, offering users quick access to essential information without disrupting their routines.

The primary focus of this project was the software architecture, leveraging Python's extensive library ecosystem to build an efficient and modular system. The Smart Mirror was developed as part of a college project, providing students with hands-on experience in software development, API integration, and Python libraries.

II. SYSTEM OVERVIEW

1. **Hardware Components**

- Two-Way Mirror: Reflective and transparent glass for dual functionality.
- LCD Screen: Displays digital content behind the mirror.
- Raspberry Pi: Acts as the primary computing unit.
- Webcam: For user recognition or gesture control.

2. Software Components

- Time and Date Display: A basic widget showing current time and date.
- Weather Updates: Real-time weather information fetched using APIs like OpenWeatherMap.
- News Feeds: Latest headlines displayed using RSS feeds or NewsAPI.
- CalendarEvents and Reminders: Integration with Google Calendar for event synchronization.

III. SOFTWARE DEVELOPMENT

1. Programming Language

The entire project was developed in Python due to its robust library support and ease of use of API integration.

2. Key Python Libraries

- Tkinter or Py QT: Tkinter is a Python standard GUI (Graphical User Interface) toolkit. It provides a way to create windows, buttons, labels, text boxes, and other visual elements that allow users to interact with Python programs. Tkinter is a object-oriented layer on top of the Tcl/Tk widget toolkit. It is included in most standard Python distributions, making it ready to use at any time.
- Requests: Sending HTTP requests is way simpler because of request library. It makes the task easier by abstracting away complexities like manual URL query string encoding, allowing developers to interact with web services and APIs more intuitively. With Requests, tasks like retrieving data, submitting forms, and handling authentication become easier. It supports various HTTP methods (GET, POST, PUT, DELETE, etc.) and handles responses efficiently, making it a fundamental tool for web development and automation in Python.
- Feedparser: Feedparser is a Python library used for parsing syndicated web feeds, such as RSS and Atom. It allows developers to extract information from these feeds in a structured format, making it easier to work with. Feedparser can handle various feed formats and versions. It can be used to monitor published feeds, gather content for web scraping, or build applications that collect and display information from different sources.

- Icalendar: iCalendar in Python refers to the use of libraries like ics.py to parse, generate, and manipulate iCalendar files. iCalendar is a file format (.ics) that allows users to store and share calendar data, such as events, tasks, and free/busy time. It's a standard format supported by various calendar applications like Google Calendar, Apple Calendar, and Microsoft Outlook.
- Python Imaging Library: The Python Imaging Library (PIL), also known as Pillow, is a powerful and versatile open-source library for image processing in Python, offering functionalities like opening, manipulating, and saving various image formats.
- Imagetk: Imagetk provides support for displaying images in Tkinter applications. Tkinter, Python's standard GUI library, has limited built-in image format support. ImageTk overcomes this problem by allowing Tkinter to display a wide range of image formats, such as JPEG and PNG, through the PhotoImage and BitmapImage classes.
- Imageenhancer: Image enhancement in python refers to the process of improving the quality or interpretability of an image. It aims to modify the image's attributes to make it more suitable for specific applications like computer vision tasks. Image enhancement techniques fall into two broad categories: spatial domain methods and frequency domain methods.

3. Api Integration

APIs were critical in enabling real-time updates:

- OpenWeatherMap API: For weather data.
- NewsAPI or RSS Feeds: For news headlines.
- Google Calendar API: For event synchronization.

4. User Interface Design

The interface was designed to be minimalistic yet functional:

- A black background ensures only essential information is visible through the mirror.
- Widgets are arranged to leave ample space for reflection.

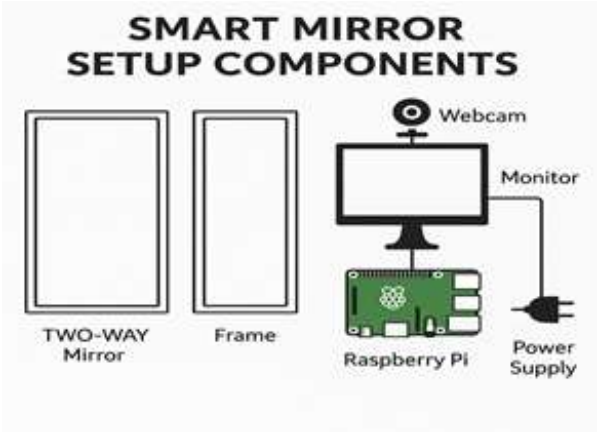

Fig1: Smart Mirror Setup Components: Two-Way Mirror, Frame, Webcam, Raspberry Pi Power Supply

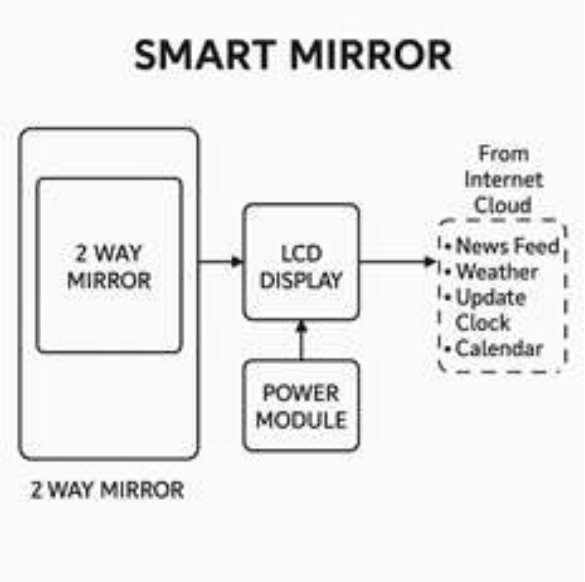

Fig2: Blocked diagram of the proposed system

IV. RESULTS

The Smart Mirror successfully integrated all planned functionalities:

- Displayed real-time weather updates, news headlines, time, and date.
- Synchronized calendar events seamlessly with Google Calendar.
- Provided an intuitive interface that required minimal user interaction.

Performance tests showed that the system operated efficiently on a Raspberry Pi without significant lag. The modular architecture allowed easy addition of new features, such as voice commands or gesture controls.

Challenges

- **API Limitations:** Rate limits on free APIs required optimization in data fetching intervals.
- **Hardware Constraints:** The Raspberry Pi's limited processing power necessitated efficient code optimization.
- User **Authentication:** Implementing OAuth2 for calendar synchronization was complex but essential for security.

VI. CONCLUSION

The Smart Mirror project demonstrated how Python can be used to create practical IoT applications with minimal hardware requirements. By focusing on software development, students gained valuable experience in

modular programming, API integration, and user-centric design.

Future Work

- Adding facial recognition for personalized content display.
- Expanding voice command capabilities using advanced NLP libraries like spaCy or NLTK.
- Integrating additional APIs for fitness tracking or smart home control.
- This project highlights the potential of smart devices to simplify daily routines while offering a platform for further innovation in IoT applications.

REFERENCES

1 Martens, B., et al. (n.d.). *Interactive Smart Mirror*. Academia.edu. https://www.academia.edu/113667207/Interactive_Smart_Mirror

2 Hardiyanto, D., Wicaksono, G., Pramudyo, A. S., Fahrizal, R., & Wiryadinata, R. (n.d.). *Interactive Smart Mirror* [PDF]. Semantic Scholar. https://pdfs.semanticscholar.org/8027/6044d1243eabb7d31fa57040898a0268ba40.pdf

3 Instructables. (n.d.). *Smart Mirror*. Instructables. https://www.instructables.com/Smart-Mirror-8/

4 Grantukas. (n.d.). *SmartMirror*. GitHub. https://github.com/grantukas/SmartMirror

5 Durgam, S., & Vanga, L. (2020). Design & development of smart mirror displaying real time sensor data. *International Journal of Engineering Research & Technology (IJERT)*, 9(7). https://www.ijert.org/design-development-of-smart-mirror-displaying-real-time-sensor-data

6 Instructables. (n.d.). *How to Build a Raspberry Pi Smart Mirror*. Instructables. https://www.instructables.com/How-to-Build-a-Raspberry-Pi-Smart-Mirror/

7 Freepik. (n.d.). *Smart Mirror Displaying Vectors*. Freepik. https://www.freepik.com/vectors/smart-mirror-displaying

8 Instructables. (n.d.). *CQ: A DIY Smart Mirror*. Instructables. https://www.instructables.com/CQ-a-DIY-Smart-Mirror/

9 Maheshwari, M., & Vohra, R. (2017). Smart mirror: A reflective interface to maximize productivity. *International Journal of Computer Applications (IJCA)*, 166(9), 1–5. https://www.ijcaonline.org/archives/volume166/number9/maheshwari-2017-ijca-914114.pdf

10 Wikipedia contributors. (n.d.). *Python (programming language)*. Wikipedia. Retrieved April 27, 2025, from https://en.wikipedia.org/wiki/Python_(programming_language)