Content is available at: CRDEEP Journals
Journal homepage: http://www.crdeepjournal.org/category/journals/global-journal-of-current-reseach-gjcr/

Global Journal of Current Research

(ISSN: 2320-2920) (Scientific Journal Impact Factor: 6.122)

UGC Approved-A Peer Reviewed Quarterly Journal

Research Paper

Built Attendance System with the help of Face Recognition for Utilizing the Modern Web Technologies

Monika Kumari[1], Jyoti Priya[1], Richa Kumari[1], Deepak Kumar Mahto[1] and Aizaz Ahmad[21]
[1-]PG Scholar, Department of Computer Applications, Tula's Institute, Dehradun
[2-]Assistant Professor ,Department of Computer Applications Tula's Institute Dehradun

ARTICLE DETAILS

Corresponding Author:
Aizaz Ahmad

Key words:
Face Recognition, Attendance System, AI/ML

ABSTRACT

This paper shows the complete attendance system we built utilizing modern web technologies. We integrated **React** and **Material-UI** for the UI, **Node.js** and **Express** for the backend, which handles administration as well as **MongoDB** for the datastore, and **OpenCV** and the **face-recognition** library for accurate face recognition. We deliver a contactless, remarkably reliable, and efficient attendance tracking system. The contactless attendance capturing and verification methodologies we developed smarten the process in educational and industrial setups. It enhances security and reduces the potential of human error.

1. Introduction

Conventional attendance methods, like RFID card scanning or manual roll calls, are frequently laborious, prone to mistakes, and susceptible to fraudulent attendance. The development of computer vision and artificial intelligence has made face recognition technology a viable option for automating attendance systems. Without requiring physical contact or manual data entry, face recognition attendance systems recognize people based on distinctive facial features captured by a camera. To accurately detect and encode faces, we use **OpenCV** and the **face-recognition** Python library to implement a full-stack face recognition attendance system. We oversee API requests and communicate with a **MongoDB** database that houses user and attendance data, with the backend created using **Node.js** and **Express**. **React** is used to build the frontend.

1.1 Problem statement

Manual attendance systems are ineffective and vulnerable to manipulation and human error. Biometric devices or RFID cards may malfunction, be misplaced, or be counterfeited. A contactless, precise, scalable attendance system that reduces human intervention and boosts data reliability is required. Problems with scalability and susceptibility to errors and manipulation make systems such as biometric and RFID tracking challenging. Issues regarding such systems include:
Fraudulent Entries: Proxy attendance is common in educational institutions.

- **Device Dependency**: Physical cards or devices may also be lost, damaged, or fabricated.
- **Human Error**: Data may be manually input incorrectly, or some fields may be omitted entirely.

2. Methodology

Face Registration: User's facial characteristics are captured and transformed into numerical vectors through facial recognition algorithms available in Python. These vectors are pickled and saved for later reference.

- **Recognition of the Face**: Webcam feeds are accessed and processed using **OpenCV**. The obtained facial encodings are compared with the kept vectors to see if they match. A successful match triggers the logging of attendance.
- **Management of Attendance**: Upon recognition, the user's details are automatically retrieved from **MongoDB** and saved along with the current date and time. The backend interacts with the recognition service to change the data and

[1]Author can be contacted at: *Professor, Department of Computer Applications, Tula's Institute, Dehradun.*
Received: 15-05-2025; Sent for Review on: *22-05*-2025; Draft sent to Author for corrections: 10-06-2025; Accepted on*: 18-06*-2025; Online Available from 22-06-2025
DOI: 10.13140/RG.2.2.19011.92968
GJCR: -8819/

synchronize it with what is already in place. To identify users, live webcam frames are transmitted to the backend, where stored known encodings are compared with captured face encodings.

Technology Stack

- **Frontend**: React + Material-UI (MUI)
- **Backend**: Node.js + Express
- **Database**: MongoDB
- **Recognition Engine**: Open-Source Computer Vision Library Face-recognition (Python)

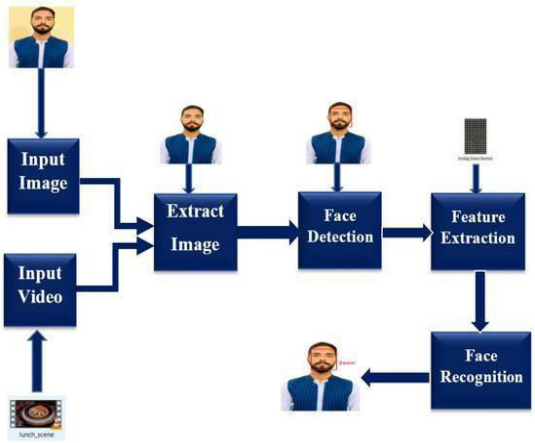

Fig1: Generic framework for face recognition

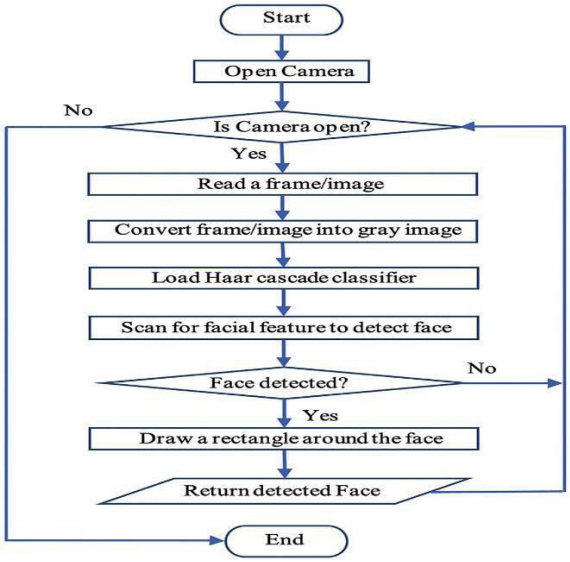

Fig 2: Flow

To manage API endpoints, send image data to the Python recognition service, and store attendance logs, the backend is built with Node.js and Express. The frontend React application uses MUI components to display attendance records in real time, send snapshots on a regular basis, and access the webcam.

Fig 3: An Attendance System Using Facial Recognition |VersionX

Fig 4: Face Recognition Attendance System in various Industry.

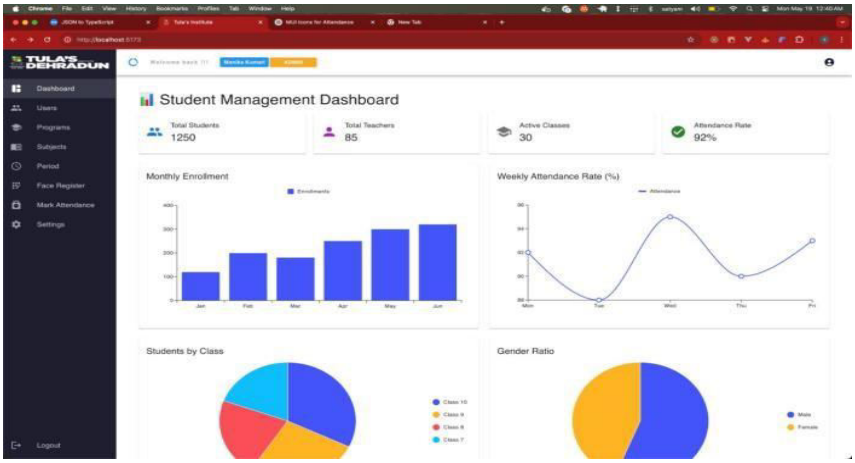

Fig 5:Dashboard

3. Result

- Real-Time Dashboard: A React based interface which presents live attendance updates with the option to filter by date, time, and user ID.
- **Notifications**: Alerts that pop up upon successful attendance record.
- Data Visualization: Trends in attendance over the course of days or weeks presented via charts which we use from libraries like Chart.js or Recharts.

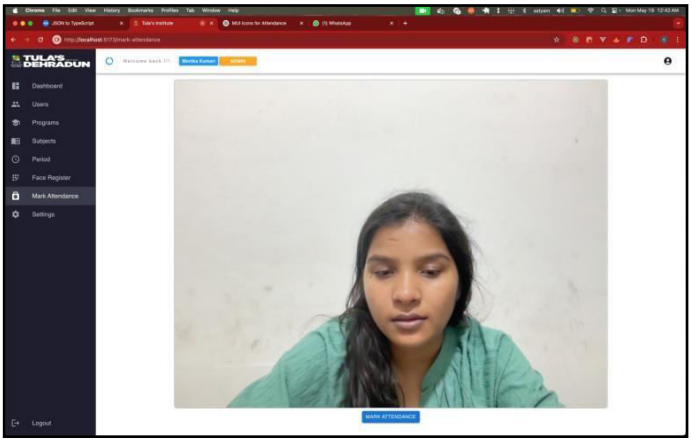

Fig 6: Marked Attendance

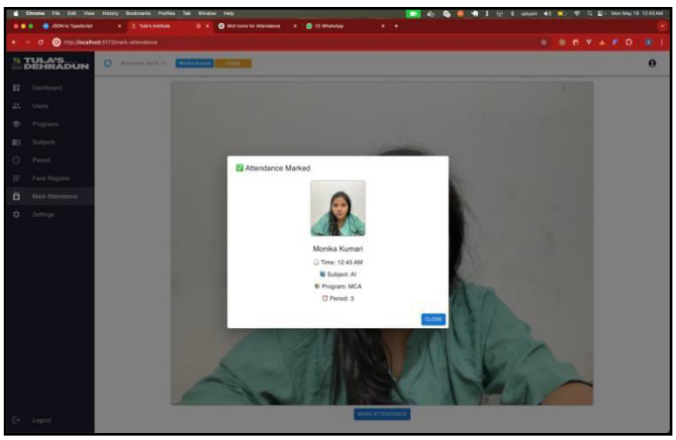

Fig 7: Attendance done

Comparative Analysis:

Feature	Manual System	ID Scanner	Face Recognition System
Contactless	No	Partial	Yes
Fraud Resistance	Low	Medium	High
Real-Time Monitoring	No	Yes	Yes
Scalability	Low	Medium	High
Hardware Requirement	Medium	High	Low

For instance:

- **Screenshot** of a dashboard built with **React** that shows attendance records in real time.
- An example of a **timestamp-based recognition success notification**.
- A **graph** that displays attendance patterns over a period of one week or one month.

4. Conclusion

An automated, precise, and contactless approach to attendance management is offered by the developed **face recognition attendance system**. Compared to conventional systems, it greatly increases efficiency and security by utilizing cutting-edge web technologies and potent computer vision algorithms.

- **Antenna**: A crucial element for both the transmitter and receiver. It facilitates the transmission and reception of radio signals, ensuring efficient communication between devices.
- **Frequency Oscillator**: The frequency oscillator generates a stable carrier frequency for the RF module. This frequency is a key parameter in establishing communication between devices.
- **Microcontroller Interface**: Many modern RF modules include interfaces with microcontrollers, enabling seamless integration into electronic systems. This interface allows for control and coordination with other components.

5. Future and scope

The face recognition-based attendance system has significant potential for future enhancements and broader deployment. One important area is scalability and remote accessibility, allowing institutions to manage and monitor attendance across multiple locations seamlessly. To strengthen security, liveness detection can be integrated to prevent spoofing attacks using photos or videos. Support for mobile applications would also be beneficial, enabling users to mark attendance using their smartphone cameras. Additionally, implementing multi-face recognition capabilities would allow the system to handle group attendance simultaneously, which is particularly useful in classrooms or large gatherings. Deploying the system on the cloud would facilitate centralized data storage, backup, and access from any location.

References

ExpressJS(2023).*ExpressWebFramework*.https://expressjs.com/
Flask(2023).*FlaskWeb FrameworkDocumentation*.https://flask.palletsprojects.com/
Kaehler,A.,&Bradski,G.(2008).*LearningOpenCV:ComputerVisionwiththeOpenCVLibrary*.O'Reilly Media.
King,D.E.(2009).*Dlib-ml:AMachineLearningToolkit*.JournalofMachineLearningResearch,10,1755– 1758.
Material-UI(2023). *MUIComponentLibrary*.https://mui.com/
MongoDBInc.(2023).*MongoDBManual*. https://docs.mongodb.com/
Node.jsFoundation(2023).*Node.jsDocumentation*. https://nodejs.org/en/docs/
Parkhi,O.M.,Vedaldi,A.,&Zisserman,A.(2015).*DeepFaceRecognition*.BritishMachineVision Conference.
PythonSoftwareFoundation(2023).*Python3Documentation*. https://docs.python.org/3/
ReactJS(2023).*ReactDocumentation*.https://reactjs.org/docs/getting-started.html
Schroff, F., Kalenichenko, D., & Philbin, J. (2015). *FaceNet: A Unified Embedding for Face Recognition and Clustering*. IEEE CVPR.
Simonyan,K.,&Zisserman,A.(2014).*VeryDeepConvolutionalNetworksforLarge-ScaleImageRecognition*. arXiv preprint arXiv:1409.1556.
Vite(2023).*Vite BuildToolDocumentation*. https://vitejs.dev/
Zhang, K. et al. (2016). *Joint Face Detection and Alignment Using Multi-Task Cascaded ConvolutionalNetworks*. IEEE Signal Processing Letters.