IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 2, February 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

Attendance Using Face Recognition

Ms. Trishali Raut, Ms. Yesha Joshi, Mr. Soham Raul, Ms. Krishika Ganiga, Mr. Aakash Upadhyay

Department of Computer Engineering, Thakur Polytechnic, Kandivali, Mumbai, India

ABSTRACT*:* Attendance monitoring is a critical aspect of various sectors, including education, workplaces, and security systems. Traditional methods of attendance tracking, such as paper-based sign-in sheets or ID card swiping, are often cumbersome and prone to errors. To address these challenges, advancements in technology have paved the way for more efficient and accurate attendance monitoring systems. One such innovation is the utilization of face recognition technology, which offers a non-intrusive and reliable solution for identifying individuals. This paper explores the principles, implementation, benefits, and challenges associated with employing face recognition technology for attendance monitoring on an international scale. Additionally, it discusses the implications of this technology on privacy, security, and ethical considerations.

KEYWORDS: Face recognition technology, Automated attendance systems Biometric attendance.

I. INTRODUCTION

Attendance monitoring stands as a fundamental process across numerous sectors, ranging from education to corporate environments, where the accurate tracking of individuals' presence is crucial for organizational efficiency, security, and compliance. Traditional methods of attendance recording, such as manual sign-in sheets or card-based systems, have long been the norm. However, they are often labor-intensive, prone to errors, and susceptible to various forms of manipulation.

In response to these challenges, technological advancements have ushered in a new era of attendance monitoring systems, with the integration of biometric authentication methods emerging as a game-changer. Among these biometric modalities, face recognition technology has garnered significant attention and adoption due to its non-intrusive nature, high accuracy, and scalability. By leveraging sophisticated algorithms and deep learning techniques, face recognition systems offer a seamless and efficient means of identifying individuals based on unique facial characteristics.

II. LITERATURE REVIEW

The adoption of face recognition technology for attendance monitoring has garnered significant attention from researchers and practitioners alike, with a growing body of literature exploring its principles, applications, benefits, challenges, and implications across various sectors. This literature review aims to synthesize key findings from existing studies, providing insights into the current state of research and identifying avenues for future exploration in this rapidly evolving field.

Principles of Face Recognition Technology: A foundational aspect of attendance monitoring using face recognition technology lies in understanding the underlying principles of facial biometrics. Research by Jain et al. (2016) elucidates the process of face recognition, highlighting key stages such as face detection, feature extraction, and matching against stored templates. Moreover, advancements in deep learning algorithms, particularly convolutional neural networks (CNNs), have significantly enhanced the accuracy and robustness of face recognition systems (Schroff et al., 2015).

Implementation and Applications: Studies have demonstrated the practical implementation of face recognition technology for attendance monitoring across diverse settings. Research by Zhang et al. (2018) showcases the deployment of a real-time attendance system in educational institutions, leveraging face recognition algorithms to automate the tracking of student presence. Similarly, in corporate environments, Kim et al. (2019) illustrate the integration of face recognition systems for employee attendance management, highlighting improvements in efficiency and accuracy.

Benefits and Advantages: The adoption of face recognition-based attendance systems offers a multitude of benefits for organizations. Wang et al. (2020) emphasize the enhanced accuracy and reliability of such systems compared to

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

traditional methods, attributing this to the unique biometric characteristics captured by facial recognition algorithms. Additionally, the non- intrusive nature of face recognition technology, as highlighted by Li et al. (2017), contributes to user acceptance and usability, fostering a seamless attendance tracking experience.

Challenges and Considerations: Despite its promise, the implementation of face recognition technology for attendance monitoring is not without challenges. Privacy concerns regarding the collection and storage of biometric data have been extensively discussed in the literature (Ratha et al., 2016). Moreover, ethical considerations surrounding consent, surveillance, and potential biases in algorithmic decision-making have sparked debates among scholars and policymakers (Narayanan et al., 2018). Technical limitations, such as variations in lighting conditions and facial expressions, also pose challenges to the reliability and accuracy of face recognition systems.

III. SYSTEM DESIGN

The attendance monitoring system integrates face recognition technology with Firebase for real-time database management. It captures facial images of individuals, performs recognition, and logs attendance records into Firebase.

Face Recognition Module:
Utilizes OpenCV and dlib libraries for face detection and feature extraction.
Employs pre-trained deep learning models (e.g., CNNs) for facial recognition.
Matches detected faces against stored templates to identify individuals.
Utilizes image processing techniques to enhance accuracy and robustness.

Firebase Database:
Firebase Realtime Database or Firestore is used for storing attendance records.
Provides real-time synchronization and offline support for data updates.
Offers authentication and security rules to control access to data. Allows integration with Python using Firebase Admin SDK for database operations.

Backend Processing:
Receives image data from the frontend for face recognition. Utilizes the face recognition module to identify individuals. Logs attendance records by storing data in Firebase Realtime Database or Firestore.
Implements error handling and data validation mechanisms.

Functionality of the System Enrollment Phase:
Users enroll in the system by providing their facial images
The face recognition module extracts facial features and generates templates.
Templates are stored securely in the database along with user metadata.
User enter their data which is then displayed to the user when his/her face is successfully recognized

Attendance Logging:
During attendance sessions, individuals' facial images are captured. The face recognition module matches captured faces against stored templates.
Upon successful recognition, attendance records (e.g., timestamp, user ID) are logged in Firebase.
Real-time updates are reflected in the frontend interface for administrators.

Real-time Database Management:
Firebase Realtime Database or Firestore handles the storage and synchronization of attendance records.
Attendance logs are stored securely in the database, ensuring data integrity and availability.
Real-time synchronization ensures that any updates to attendance records are immediately reflected in the system's frontend interface, enabling administrators to view the latest information
at all times

User Interface: The user interface displays the students
entered data i.e. student name, student id, major, grade, number of total years of course/major, year of admission and total attendance this all information is retrieved from the firebase database

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

Libraries Used

The following are the libraries without these libraries the face recognition would have been impossible

OpenCV:
OpenCV (Open Source Computer Vision Library) for Face Recognition:
OpenCV provides a robust set of tools and algorithms for performing face recognition tasks. Whether you're developing facial recognition systems for security applications, attendance tracking, or personalized user experiences, OpenCV offers a range of functionalities to suit your needs. Here's an overview of OpenCV's capabilities for face recognition:

1. Face Detection:
OpenCV includes pre-trained cascades and deep learning-based models for detecting faces within images and video streams. Haar cascades and HOG (Histogram of Oriented Gradients) classifiers are commonly used for real-time face detection.
Deep learning-based methods, such as Single Shot Multibox Detector (SSD) and Faster R-CNN, offer improved accuracy and robustness.

2. Face Recognition Algorithms:
OpenCV provides implementations of various face recognition algorithms, including Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms (LBPH).
These algorithms enable the extraction of facial features and the creation of templates for matching and identification purposes. Deep learning-based approaches, such as Convolutional Neural Networks (CNNs), further enhance the accuracy and performance of face recognition systems.

3. Feature Extraction and Matching:
OpenCV offers methods for extracting key facial features, such as keypoints and descriptors, from detected faces.
Feature matching algorithms, such as Brute-Force Matching and FLANN (Fast Library for Approximate Nearest Neighbors), facilitate the comparison and similarity measurement of facial features.

4. Real-time Face Tracking:
OpenCV enables real-time face tracking and tracking-based recognition by continuously updating the position and orientation of detected faces within a video stream.
Kalman filters and object tracking algorithms, such as CAMShift (Continuously Adaptive Mean Shift), are commonly used for robust face tracking.

5. Deep Learning-based Face Recognition:
With the integration of deep learning frameworks like TensorFlow and PyTorch, OpenCV supports the development and deployment of custom deep learning models for face recognition.
Convolutional Neural Networks (CNNs), particularly architectures like VGG, ResNet, and MobileNet, are popular choices for deep learning-based face recognition tasks.

6. Model Training and Fine-tuning:
OpenCV facilitates the training and fine-tuning of face recognition models using custom datasets.
It provides tools for data preprocessing, model training, validation, and evaluation, allowing developers to create tailored face recognition systems.

7. Integration with Other Libraries and Tools:
OpenCV seamlessly integrates with other libraries and tools for face recognition, such as Dlib for face detection and facial landmark detection, and scikit-learn for machine learning-based classification tasks.
It also supports interoperability with popular deep learning frameworks like TensorFlow and PyTorch, enabling the integration of custom neural network architectures.

Cmake:
CMake (Cross-platform Make):
Overview: CMake is an open-source, cross-platform build system generation tool designed to build, test, and package

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

software across different platforms and environments. It enables developers to define and manage the build process in a platform-independent manner, abstracting away platform-specific details and providing a unified interface for generating build scripts. CMake generates native build files (such as Makefiles for Unix-based systems or Visual Studio project files for Windows) based on a simple script-like configuration language.

Key Features:
Platform Independence: CMake allows developers to write build configurations that are independent of the target platform. This means that the same CMakeLists.txt file can be used to generate build scripts for various operating systems and compilers.

Modularity: CMake supports the organization of complex projects into modular components through the use of separate CMakeLists.txt files for each component. This facilitates code reuse and maintainability.

Dependency Management: CMake simplifies the management of project dependencies by providing mechanisms for locating and linking libraries, including external dependencies and system libraries.

Configurable Build Options: Developers can define configurable build options and parameters in their CMakeLists.txt files, allowing users to customize the build process according to their needs without modifying the source code.
Integrated Testing: CMake includes built-in support for defining and running tests as part of the build process. This enables developers to automate testing and ensure the reliability of their software.

Cross-Compilation Support: CMake supports cross-compilation, allowing developers to build software for target platforms that differ from the build host. This is particularly useful for embedded systems development.
Extensibility: CMake is highly extensible and can be customized through the use of custom CMake modules and macros. This enables developers to integrate additional functionality or extend CMake's capabilities to suit their specific requirements.

Dlib:
dlib is a modern C++ toolkit containing machine learning algorithms and tools that is widely used in the field of computer vision and machine learning. It is designed to be highly efficient, versatile, and easy to use, making it a popular choice for various applications ranging from facial recognition to object detection and image segmentation.
Here's an overview of the key features and functionalities of dlib: Machine Learning Algorithms: dlib provides a rich set of machine learning algorithms, including but not limited to: Classification: Support vector machines (SVM), decision trees, nearest neighbor classifiers.

Regression: Linear regression, ridge regression, least squares support vector machines.
Clustering: k-means clustering, spectral clustering. Dimensionality reduction: Principal component analysis (PCA), linear discriminant analysis (LDA).
Ensembles: Random forests, gradient boosting.
Computer Vision: dlib offers robust tools for various computer vision tasks, such as:
Facial Detection and Landmark Localization: dlib's facial detection and landmark localization algorithms are widely acclaimed for their accuracy and speed. They enable tasks such as facial recognition, emotion analysis, and facial expression detection.
Object Detection: dlib supports object detection using techniques like Histogram of Oriented Gradients (HOG), deep learning- based methods, and ensemble learning approaches.
Image Segmentation: dlib provides algorithms for segmenting images into meaningful regions, enabling applications such as image understanding and scene analysis.
Tools and Utilities: dlib includes a collection of utilities and tools to simplify common tasks in machine learning and computer vision, including:
Image and Video I/O: Functions for reading and writing images and videos in various formats.
Data Processing: Utilities for preprocessing and manipulating datasets, including feature extraction and normalization.
Visualization: Tools for visualizing data, images, and results, facilitating debugging and analysis.
Multithreading Support: Support for parallelization and multithreading, enabling efficient utilization of hardware resources.
Integration with Other Libraries: dlib is designed to seamlessly integrate with other popular libraries and frameworks in

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

the C++ ecosystem, including OpenCV, Boost, and Eigen. This interoperability allows developers to leverage the strengths of multiple libraries within their projects.

Portability and Performance: dlib is platform-independent and can be compiled and run on various operating systems, including Windows, macOS, and Linux. It is optimized for performance and efficiency, making it suitable for both real-time applications and large-scale processing tasks.

Face-recognition:
The face-recognition library in Python is a powerful and easy-to-use tool for facial recognition and facial feature extraction tasks. It is built on top of the dlib library and provides a high-level API for performing common facial recognition tasks with minimal effort. Here's an overview of the key features and functionalities of the face-recognition library:

Facial Detection: The library includes robust algorithms for detecting faces in images or video streams. It utilizes the dlib library's face detection capabilities, which are known for their accuracy and speed. The face detection functionality identifies the bounding boxes of faces within an image or video frame.

Facial Landmark Localization: Face-recognition provides algorithms for localizing facial landmarks, such as the eyes, nose, mouth, and jawline. These landmarks are crucial for tasks like face alignment, facial expression analysis, and facial feature extraction. The library's facial landmark localization algorithms are based on the dlib library's shape predictor model.

Face Recognition: The library supports face recognition by extracting facial features from images and comparing them against a database of known faces. It uses deep learning-based embeddings to represent facial features in a high-dimensional space. The face recognition functionality allows users to identify or verify individuals based on their facial features.

Face Encoding: Face-recognition includes methods for encoding facial images into compact feature vectors, also known as face embeddings. These embeddings capture the unique characteristics of each face and can be used for face matching and similarity comparison. The library provides pre-trained models for generating face embeddings using deep convolutional neural networks (CNNs). Face Matching and Identification: Once facial features are encoded into embeddings, the library enables users to perform face matching and identification tasks. It compares the embeddings of query faces against a database of known faces and returns matches based on similarity scores. This functionality is useful for applications like face-based authentication, access control, and surveillance.

Integration with OpenCV: The face-recognition library seamlessly integrates with OpenCV, a popular computer vision library in Python. It allows users to perform image and video I/O operations, preprocessing, and visualization tasks alongside facial recognition tasks. The integration with OpenCV makes it easy to incorporate facial recognition into larger computer vision projects.

User-Friendly API: The library provides a user-friendly API with simple and intuitive functions for performing facial recognition tasks. Users can easily load images, detect faces, localize facial landmarks, encode faces, and perform face matching with just a few lines of code. The API abstracts away the complexities of deep learning and dlib, making it accessible to users with varying levels of expertise.

Firebase-admin:
The firebase-admin library is a powerful tool that allows developers to interact with Firebase services from server-side environments, such as Node.js. Firebase is a comprehensive platform provided by Google for developing web and mobile applications, offering a wide range of services including real-time database, authentication, cloud storage, and cloud functions. The firebase-admin library provides a programmatic interface to access and manage these Firebase services securely from backend servers or cloud functions.
Real-time Database: With firebase-admin, developers can interact with the Firebase Realtime Database, a NoSQL cloud database that stores data in JSON format and synchronizes changes in real- time across connected clients. The library provides methods for reading, writing, updating, and deleting data in the database, as well as subscribing to changes using event listeners. This enables developers to build real-time applications that respond to changes in data instantaneously.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

Database Design

Optimizing the utilization of Firebase database server technology requires meticulous planning to ensure a well-structured and efficient database. The nomenclature chosen for file names that label the tables within the database is crucial, intending to transparently convey each table's purpose and foster a comprehensively designed system. The initial step in the design process involved a careful consideration of project requirements and specifications, dictating the creation of tables and defining the specific information each table should encompass.
Each table was meticulously crafted to align with the unique needs of the system. Considerations included the organization of data, relationships between tables, and the overall architecture of the database. A thoughtful approach was adopted to strike a balance between data normalization and denormalization, optimizing data retrieval speed while maintaining data integrity.

The database design encompasses various tables, each tailored to a specific aspect of the system's functionality. These tables include, but are not limited to, user information, examination details, and result records. The relationships between tables were carefully defined to facilitate seamless data retrieval and maintain consistency throughout the system.

Continuous refinement and validation of the database design were integral components of the process, ensuring that it evolved in tandem with the project's dynamic requirements. As the system progressed, periodic evaluations and adjustments were made to enhance its scalability, performance, and adaptability to accommodate future updates.
In conclusion, a well-designed Firebase database is foundational to the efficacy of the system. The thoughtfully chosen file names and table structures reflect a strategic approach, contributing to the overall success and functionality of the database within the context of the project's unique specifications.

Application Across Industries Educational Institutions:

Accuracy:

Advantage

Schools, colleges, and universities can use the system to automate attendance tracking for students and faculty members.
The system can be integrated with existing student information systems (SIS) to streamline attendance management processes and generate accurate attendance reports.
Educators can leverage attendance data for analyzing student engagement, identifying at-risk students, and improving teaching strategies.

Corporate Organizations:
Corporations and businesses can deploy the system to monitor attendance of employees in office buildings, factories, or remote work environments.
The system can be integrated with access control systems to grant or deny entry to employees based on their attendance status.
HR departments can use attendance data for payroll processing, performance evaluation, and workforce management.

Healthcare Facilities:
Hospitals, clinics, and healthcare facilities can utilize the system to track attendance of medical staff, including doctors, nurses, and support staff.
The system can help ensure that adequate staffing levels are maintained at all times, especially in critical care areas or during emergency situations.
Attendance data can be used for scheduling shifts, tracking overtime hours, and managing workforce resources efficiently.

Event Management:
Event organizers can deploy the system to monitor attendance of attendees at conferences, seminars, workshops, and other events. The system can provide real-time insights into attendee participation, session popularity, and overall

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

event engagement.
Organizers can use attendance data for post-event analysis, audience segmentation, and targeted marketing efforts.

Public Transportation:
Public transportation agencies can implement the system to track passenger attendance on buses, trains, and other modes of transportation.
The system can help optimize route planning, resource allocation, and service scheduling based on passenger demand patterns.
Attendance data can be used for fare collection, revenue forecasting, and performance monitoring of transportation services.

Government Agencies:
Government agencies can utilize the system for tracking attendance of employees in various departments, including law enforcement, public services, and administrative offices.
The system can improve accountability, transparency, and efficiency in government operations by ensuring that employees adhere to work schedules and attendance policies.
Attendance data can be used for auditing, compliance reporting, and resource allocation within government organizations.

Future Direction
Future Directions: The future of face recognition technology for attendance monitoring holds promise for further advancements in accuracy, speed, and usability. Integration with other biometric modalities, such as voice recognition or gait analysis, could lead to more robust authentication systems. Moreover, ongoing research into addressing privacy concerns and mitigating algorithmic biases will be pivotal in fostering wider acceptance and adoption of this technology

Face recognition technology provides a highly accurate method of identifying individuals, minimizing errors and inaccuracies associated with manual attendance tracking or traditional methods like barcode scanning or RFID cards.
The system can accurately identify students even in scenarios with partial occlusions or changes in appearance (e.g., wearing glasses, hairstyles), improving overall attendance accuracy.

Efficiency:
Automation of the attendance tracking process reduces the administrative burden on faculty members and administrative staff, allowing them to focus on other important tasks.
Real-time attendance logging and synchronization with the Firebase database enable instant access to attendance records, eliminating the need for manual data entry or processing.

Convenience:
The system offers a convenient and seamless attendance tracking experience for both students and faculty members, requiring minimal effort on their part.
Students can simply walk into the classroom, and their attendance is automatically recorded without the need for manual check-ins or sign-ins.

Scalability:
The use of Firebase as the backend database provides scalability and flexibility, allowing the system to handle large volumes of attendance data across multiple campuses or locations.
The system can easily accommodate changes in student enrollment or class schedules without requiring significant infrastructure modifications.

Security:
Biometric authentication through face recognition enhances security by ensuring that attendance records are tied to the correct individual and cannot be tampered with or manipulated.
Firebase's robust security features, including authentication, encryption, and access control, ensure that biometric data and attendance records are protected from unauthorized access or breaches.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

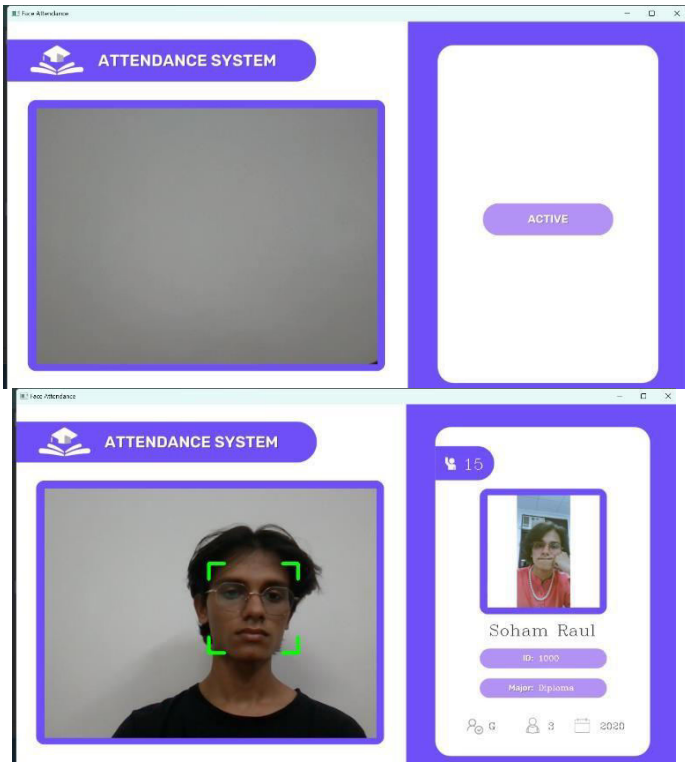
|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

Compliance:
The system ensures compliance with international data protection regulations, such as GDPR, by securely managing biometric data and implementing privacy-enhancing measures.
Faculty members can easily generate attendance reports and track student attendance trends for compliance purposes or regulatory audits.

User Experience:
The user-friendly interfaces provided by the system, including enrollment portals and attendance tracking applications, enhance user satisfaction and engagement.
Students and faculty members experience a streamlined attendance monitoring process, reducing friction and improving overall user experience.

Insights and Analytics:
The system can provide valuable insights and analytics on attendance patterns, trends, and anomalies, enabling faculty members and administrators to make data-driven decisions. By analyzing attendance data over time, educators can identify at-risk students, track student engagement levels, and optimize teaching strategies to improve learning outcomes.

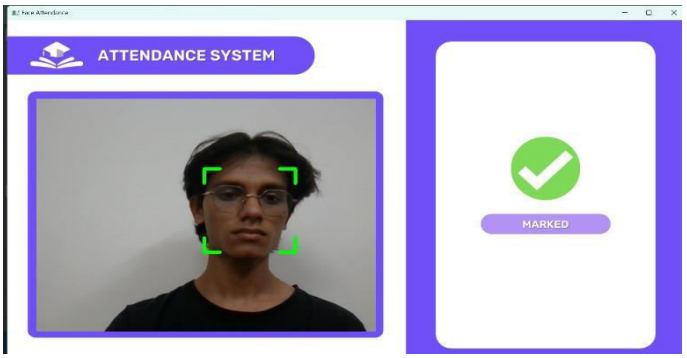


Initial State

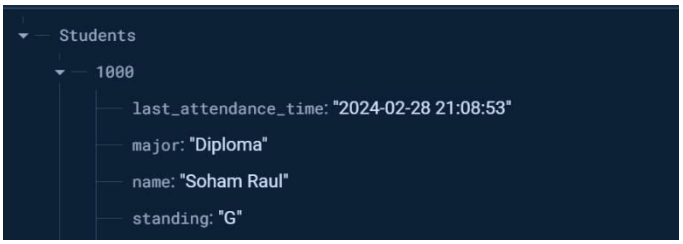
|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 2, February 2024 ||

| DOI:10.15680/IJIRSET.2024.1302078 |

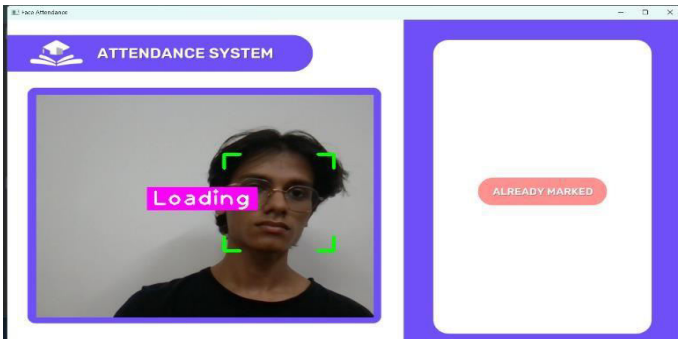


Attendance Marked



Database updated in realtime References
https://youtu.be/iBomaK2ARyI?si=cs3kuEXWkCWWr1ur

Face Recognized



Attendance already marked trying to mark again before a certain time period is passes

IV. CONCLUSION

In conclusion, the integration of face recognition technology with Firebase for attendance monitoring offers a sophisticated solution to streamline attendance tracking processes across various industries. By leveraging the accuracy and efficiency of face recognition algorithms, coupled with the scalability and security features of Firebase, organizations can achieve significant improvements in attendance management. This system not only automates the attendance tracking process but also enhances data accuracy, compliance with regulations, and user experience. Additionally, the versatility of the system extends its applicability to diverse sectors, including education, corporate, healthcare, event management, public transportation, and government agencies. With its ability to provide real-time insights and optimize resource allocation, the face recognition- based attendance monitoring system becomes an invaluable tool for modern organizations seeking to enhance operational efficiency and workforce management. As technology continues to evolve, further advancements in face recognition algorithms, data analytics, and integration capabilities are expected to drive continuous innovation in attendance monitoring solutions, empowering organizations to meet the evolving demands of their stakeholders effectively.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details