

A
Report
On

Biometric Attendance Management System Using Face Recognition

Submitted in partial fulfillment of the requirement for the award of Degree of
Bachelor of Technology in **Computer Science & Engineering**



DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

Submitted By

Priyank Saxena (1900650100044)

Vivek Sharma (1900650100061)

Under the supervision of

PROJECT GUIDE

Mrs.Ritika Kulshreshtha
(Asst. Prof.)

PROJECT COORDINATOR

Mr. Vatsya Tiwari
(Asst. Prof.)



Session: 2022-23

Department Of Computer Science & Engineering

B.S.A. COLLEGE OF ENGG. & TECHNOLOGY, MATHURA

B.S.A. College of Engineering & Technology, Mathura (U.P)

Department of Computer Science & Applications

CERTIFICATE

This is to certify that the work embodies in this Project entitled "**BIOMETRIC ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION**" being submitted by "**PRIYANK SAXENA (1900650100044)**", "**VIVEK SHARMA (1900650100061)**" in partial fulfillment of the requirement for the award of "**Bachelor of Technology in (CSE)**" to Dr. A.P.J Abdul Kalam Technical University, Lucknow (U.P.) during the academic year 2022-2023 is a record of bonafide piece of work, carried out by them under my supervision and guidance in the "**Department of Computer Science & Applications**", **B.S.A. College of Engineering & Technology, Mathura.**

Project Guide:

Signature

Mrs.Ritika Kulshreshtra

(Asst. Prof.)

Project Coordinator:

Signature

Mr.Vatsya Tiwari

(Asst. Prof.)

B.S.A. College of Engineering & Technology, Mathura (U.P)

Department of Computer Science & Applications

CERTIFICATE OF APPROVAL

The Project entitled "**BIOMETRIC ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION**" and application name entitled "**FIGMENT AUTHENTICATOR**" being submitted by "**Priyank Saxena (1900650100044) & Vivek Sharma (1900650100061)**" have been examined by us and is hereby approved for the award of degree "Bachelor of Technology in CSE", for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinions expressed or conclusion drawn therein, but approved the project only for the purpose for which it has been submitted.

PROJECT GUIDE

Mrs. Ritika kulshreshtha
(Asst. Prof.)

PROJECT COORDINATOR

Mr. Vatsya Tiwari
(Asst. Prof.)

B.S.A. College of Engineering & Technology, Mathura

Department of Computer Science & Applications

DECLARATION

We “**PRIYANK SAXENA** and **VIVEK SHARMA**” students of **Bachelor of Technology in (CSE)**, session 2022-23 **B.S.A. College of Engineering & Technology, Mathura**, hereby declare that the work presented in this Project entitled “**BIOMETRIC ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION**” is the outcome of our own work, is confide and correct to the best of our knowledge and this work has been carried out taking care of Ethics. The work presented does not infringe any patented work and has not been submitted to any other University or anywhere else for the award of any degree or any professional diploma.

PRIYANK SAXENA

(1900650100044)

VIVEK SHARMA

(1900650100061)

Date: 30/05/2023

B.S.A. College of Engineering & Technology, Mathura

Department of Computer Science & Applications

ACKNOWLEDGEMENT

It is not possible to prepare a project report without the assistance & encouragement of other people. This one is certainly no exception. On the very outset of this report, we would like to extend my sincere & heartfelt obligation towards all the personages who helped us in this endeavor. We are ineffably indebted to Mrs. Ritika Kulshreshtha, our guide for conscientious guidance and encouragement to accomplish this project. We extend our gratitude to B.S.A. College of Engg. & Tech., Mathura for giving us this opportunity. At last, but not the least, gratitude goes to all my friends who directly or indirectly helped me to complete this project report.

PRIYANK SAXENA
(1900650100044)

VIVEK SHARMA
(1900650100061)p

Date: 30/05/2023

Table of Contents

S.No	Title	Page No.
1.	Introduction 1.1. Objective 1.2. Project Scope 1.2.1. System Overview 1.2.2. Functional Requirements 1.2.3. Non-Functional Requirements 1.3. Background 1.4. Introduction to Facial Recognition 1.5. About KNN Classifier Facial Recognition Algorithm using Scikit-learn 1.6. Problem Statement	1 1 3 6 7 9 11
2.	Feasibility Study 2.1. Technical Feasibility 2.2. Economic Feasibility 2.3. Operational Feasibility 2.4. Security and Privacy Feasibility	13 13 14 15 16
3.	Requirement Specification 3.1. Hardware Requirements 3.1.1. Camra 3.1.2. Computer System 3.1.3. Graphics Processing Unit (GPU) 3.1.4. Network Equipments 3.1.5. Storage Device 3.1.6. Display and Input Devices 3.1.7. Power Backup	17 17

	3.2. Software Requirements 3.2.1. Integrated Development Environment 3.2.2. Operating System 3.2.3. Programming Language 3.2.4. Web Framework 3.2.5. Web Technologies 3.2.6. Python Modules and Libraries	18
	3.3. Data Requirements 3.3.1. User Data 3.3.2. Face Image Data 3.3.3. Attendance Records 3.3.4. System Logs 3.3.5. Configuration Data 3.3.6. Training Data	20
4.	Methodology 4.1. Requirement Analysis 4.2. System Design 4.3. Data Collection and Preprocessing 4.3.1. Data Collection 4.3.2. Data Preprocessing 4.4. Face Recognition Model 4.5. Enrollment Process 4.6. Attendance Tracking 4.7. System Integration 4.8. Testing and Evaluation	22 23 23 23 24 24 24 24 24
5.	System Design Documentation 5.1. Flow Chart 5.2. Context Diagram 5.3. Component Diagram 5.4. Data Flow Diagram	26 26 27 27 29
6.	System Implementation and Coding 6.1. Flask Code(Python): app.py 6.2. Html File: index.html 6.3. Html File: home.html	30 30 35 51

7.	Screenshots	57
	7.1. Front Web Page	57
	7.1.1. Introduction Section	
	7.1.2. Features Section	
	7.1.3. Our-Team Section	
	7.1.4. Footer Section	
	7.2. Attendance Management System Web Page	59
	7.2.1. Attendance Section	
	7.2.2. Add User Section	
	7.2.3. Delete User Section	
	7.2.4. View Database Section	
	7.3. Working of Application	61
	7.3.1. Attendance Marking	
	7.3.2. Adding New User	
	7.3.3. Deleting User	
	7.4. Face Folder	65
8.	Future Scope	66
	8.1. Admin Authentication	66
	8.2. Pricing Transaction Gateway	66
	8.3. Application Performance Improvement	66
	8.4. Secure and Scalable Database Connectivity	67
	8.5. Machine Learning Model Updates	67
9.	Conclusion	69
10.	References	71

List of Figures

Fig No. 1 : Facial Recognition Process.....	8
Fig No. 2 : KNN Classification using Scikit-learn.....	11
Fig No. 3 : Flow Chart.....	26
Fig No. 4 : Context Diagram.....	27
Fig No. 5 : Component Diagram.....	28
Fig No. 6 : DFD Face Recognition.....	29
Fig No. 7 : DFD Admin Portal.....	29
Fig No. 8 : Introduction Section.....	57
Fig No. 9 : Features Section.....	57
Fig No. 10 : Our-Team Section.....	58
Fig No. 11 : Footer Section.....	58
Fig No. 12 : Face Recognition Based Attendance System.....	59
Fig No. 13 : Attendance Section.....	59
Fig No. 14 : Add User Section.....	60
Fig No. 15 : Delete User Section.....	60
Fig No. 16 :View Database Section.....	61
Fig No. 17 :Facial Recognition.....	61
Fig No. 18 : Attendance Marked.....	62
Fig No. 19 : Attendance Report (CSV File).....	62
Fig No. 20 : Enter User Data.....	63
Fig No. 21 : Capturing User Face.....	63
Fig No. 22 : Total 50 Image Captured.....	63
Fig No. 23 : Console Message: Training Model.....	64
Fig No. 24 : Enter User Data.....	64
Fig No. 25 : Console Message: Deleted User.....	64

Fig No. 26 : Face Folder.....65

List of Tables

Table no.1 : Methodology Table.....22

Table no.2 : Testing Application Performance.....25

1. INTRODUCTION

1.1. Objective

An attendance management system using face recognition has gained significant attention and adoption in various sectors, including education, corporate environments, and public institutions. This system utilizes advanced facial recognition technology to automate and enhance the process of tracking and managing attendance. By eliminating the need for manual attendance recording methods, such as paper registers or swipe cards, this system aims to achieve a range of objectives. In this article, we will elaborate on the objectives and benefits of implementing an attendance management system using face recognition.

- Accuracy**

One of the primary objectives of an attendance management system using face recognition is to ensure accuracy in attendance tracking. Facial recognition algorithms are designed to identify individuals with a high degree of precision. By comparing facial features captured in real-time with pre-registered images, the system can accurately determine an individual's identity. This reduces the possibility of errors and eliminates the potential for proxy attendance, significantly enhancing the reliability of attendance records.

- Efficiency**

The system aims to improve efficiency by automating the attendance recording process. Traditional methods require manual data entry or card swiping, which can be time-consuming and prone to errors. With face recognition, individuals can be quickly and seamlessly identified as they enter a specific location. The system captures and records attendance data in real-time, eliminating the need for manual paperwork and reducing administrative tasks. This automated approach not only

saves time for both administrators and individuals but also ensures a more streamlined and efficient attendance management process.

- **Time-saving**

Implementing an attendance management system using face recognition results in significant time savings. The system eliminates the need for manual attendance tracking, reducing administrative burdens and freeing up valuable time for other important tasks. Attendance records are automatically generated and updated in real-time, enabling instant access to accurate attendance data. Administrators can easily retrieve attendance information, generate reports, and perform necessary analysis, thereby saving time and enabling prompt decision-making.

- **Security**

Security is a crucial objective of any attendance management system. Facial recognition technology enhances the security of the attendance tracking process. By accurately identifying individuals, the system ensures that only authorized personnel are granted access. This significantly reduces the risk of identity fraud, unauthorized entry, and attendance manipulation. The system maintains a secure database of registered facial images, employing advanced encryption techniques to protect sensitive information, thereby safeguarding the overall integrity of the attendance management system.

- **Streamlined Data Management**

Another key objective of an attendance management system using face recognition is to streamline data management. The system centralizes attendance data in a secure and easily accessible platform. Administrators can efficiently manage and organize attendance records, eliminating the need for physical storage of paper

registers. Additionally, the system enables comprehensive reporting and analytics, facilitating data-driven decision-making processes. Integration with other systems, such as payroll or student information systems, further enhances data management capabilities, enabling seamless data sharing and reducing manual data transfer.

1.2. Project Scope

1.2.1. System Overview:

The attendance management system using face recognition is an innovative solution designed to revolutionize the process of tracking and managing attendance in educational institutions, corporate environments, and other organizations. The system leverages advanced facial recognition technology to automate attendance recording and eliminate the need for traditional manual methods.

With its primary purpose being to streamline attendance management, the system offers a range of powerful functionalities. Administrators can register individuals into the system by capturing their facial images and associating them with relevant identification details. These registered images serve as a reference for subsequent identification. The system utilizes sophisticated facial recognition algorithms to compare real-time facial images with the registered images, ensuring accurate identification and verification of individuals.

Once an individual is successfully identified, the system records their attendance, capturing the date and time of entry. This information is stored in a centralized database, creating a comprehensive and reliable record of attendance data. Administrators have real-time access to this data, allowing them to monitor attendance as it happens, view attendance trends, and quickly identify any anomalies or patterns.

To facilitate effective management and analysis of attendance data, the system provides a user-friendly interface. Administrators can navigate through the system effortlessly, performing tasks such as registering new individuals, monitoring attendance, generating reports, and analyzing attendance patterns. The interface is designed to be intuitive, ensuring that administrators can easily navigate and utilize the system's functionalities without the need for extensive training.

In terms of system components, the attendance management system comprises a robust facial recognition engine that employs state-of-the-art algorithms to accurately identify individuals. The system relies on a secure and centralized database to store registered facial images and attendance records. These components work harmoniously to ensure the system's efficiency, accuracy, and reliability.

Moreover, the attendance management system has the potential for integration with other systems. This allows for seamless data sharing and synchronization with other organizational systems such as payroll or student information systems. Integration enhances overall operational efficiency and facilitates more streamlined and automated processes.

Overall, the attendance management system using face recognition offers a modern and efficient approach to attendance tracking. By automating the process, it minimizes errors, saves time, and improves overall accuracy and reliability. With its user-friendly interface and powerful functionalities, the system empowers administrators to effectively manage attendance data, generate insightful reports, and make data-driven decisions to optimize resource allocation and enhance organizational efficiency.

1.2.2. Functional Requirements:

- **Facial Registration:** Administrators can register individuals by capturing their facial images and entering their identification details.
- **Facial Recognition:** The system will compare real-time facial images with registered images to accurately identify individuals.
- **Attendance Recording:** Upon successful identification, the system will record the attendance with date and time stamp.
- **Real-time Monitoring:** Administrators can view and track attendance data in real-time.
- **Reporting:** The system will generate basic attendance reports, such as daily or monthly summaries.
- **Integration:** The system can be integrated with other systems to synchronize attendance data if needed.

1.2.3. Non-functional Requirements:

- **Performance:** The system should process facial recognition requests quickly and efficiently.
- **Accuracy:** The facial recognition algorithms should have a high level of accuracy to minimize errors.
- **User Interface:** The system should have a simple and intuitive interface for easy management.
- **Reliability:** The system should reliably capture attendance data, even during system failures.
- **Privacy:** The system should adhere to privacy regulations and protect sensitive data.
- **Compatibility:** The system should work on commonly used hardware and operating systems.

1.3. Background

Face recognition is vital in quotidian life in order to identify family, buddies or someone we are familiar with. We might not perceive that several way have actually been taken in order to identify mortal faces. mortal intelligence allows us to admit information and interpret the information in the recognition process. We admit information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic swells which are radiated from a source onto an object and projected to mortal vision. Robinson- Riegler,G., & Robinson- Riegler,B.(2008) mentioned that after visual processing done by the mortal visual system, we actually classify shape, size, figure and the texture of the object in order to anatomize the information. The anatomized information will be compared to other representations of objects or faces that live in our memory to fete. In fact, it's a hard challenge to make an automated system to have the same capability as a mortal to recognize faces. still, we need a large memory to fete different faces, for illustration, in the Universities, there are a lot of scholars with different races and genders, it's impossible to flash back every face of the existent without making misapprehensions. In order to overcome mortal limitations, computers with nearly bottomless memory, high processing speed and power are used in face recognition systems. The mortal face is a unique representation of individual identity. thus, face recognition is defined as a biometric system in which identification of an existent is performed by comparing real- time internee images with stored images in the database of that person(Margaret Rouse, 2012).

Presently, face recognition systems are current due to their simplicity and miraculous performance. For case, field protection systems and FBI use face recognition for lawless examinations by tracking suspects, missing children and drug exertion(Robert Silk, 2017). piecemeal from that, Facebook, which is a popular social networking website, tools face recognition to allow the stoners to tag their buddies in the print for entertainment purposes(Sidney Fussell, 2018). likewise, Intel Company allows the stoners to use face recognition to get access to their online account(Reichert,C., 2017).

Apple allows the stoners to unleash their mobile phone, iPhone X by using face recognition(deAgonia,M., 2017). Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which demanded the director to descry eyes, cognizance, nose and mouth from images. The distance and rates between the located features and the common reference points are also calculated and compared. The studies were further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features analogous as hair color and lip viscosity to automate the recognition. In 1988, Kirby and Sirovich first suggested top element analysis(PCA) to break face recognition problems. multitudinous studies on face recognition were also conducted continuously until moment(Ashley DuVal, 2012).

1.4. Introduction to Facial Recognition

Clearly! Facial recognition is a technology that uses artificial intelligence(AI) and computer vision to identify and corroborate individualities grounded on their unique facial features. It has come an integral part of colorful operations, including security, access control, authentication, and more.

The process of facial recognition involves several ways.

Face Discovery, The first step is to detect and descry faces within an image or videotape sluice. This is generally done using algorithms that dissect patterns and features associated with mortal faces, similar as the arrangement of eyes, nose, and mouth.

Face Alignment and Normalization, Once a face is detected, the system aligns and normalizes it by relating facial milestones. These milestones are specific points on the face, similar as the corners of the eyes and the tip of the nose. By aligning the face grounded on these milestones, the system ensures thickness for farther analysis.

Point birth The coming step is to prize applicable features from the face. This involves landing the unique characteristics that distinguish one face from another. Common features include the shape of the face, the distance between the eyes, the size of the nose, and so on. These features are converted into a fine representation called a face template or face print.

Matching and Recognition: The facial recognition system applies algorithms to compare the captured face template with the face templates in the database. It calculates a similarity score or distance between the templates to determine if there's a match. However, the system recognizes the individual and provides an identification or verification result. If the similarity score exceeds a destined threshold.

Decision and Affair Grounded on the corresponding results, the system can make opinions similar as granting access, flagging a implicit match for farther scrutiny, or simply furnishing an identification result. The affair can be in the form of a double response(match or no match) or a ranked list of possible matches, depending on the specific operation.

It's worth noting that ultramodern facial recognition systems frequently employ machine literacy ways, similar as deep neural networks, to enhance delicacy and performance. These models are trained on large datasets of labeled faces to learn patterns and optimize the recognition process.

While facial recognition offers multitudinous benefits, it also raises enterprises about sequestration, surveillance, and implicit impulses. The responsible and ethical use of facial recognition technology is pivotal to address these enterprises and insure proper safeguards are in place.

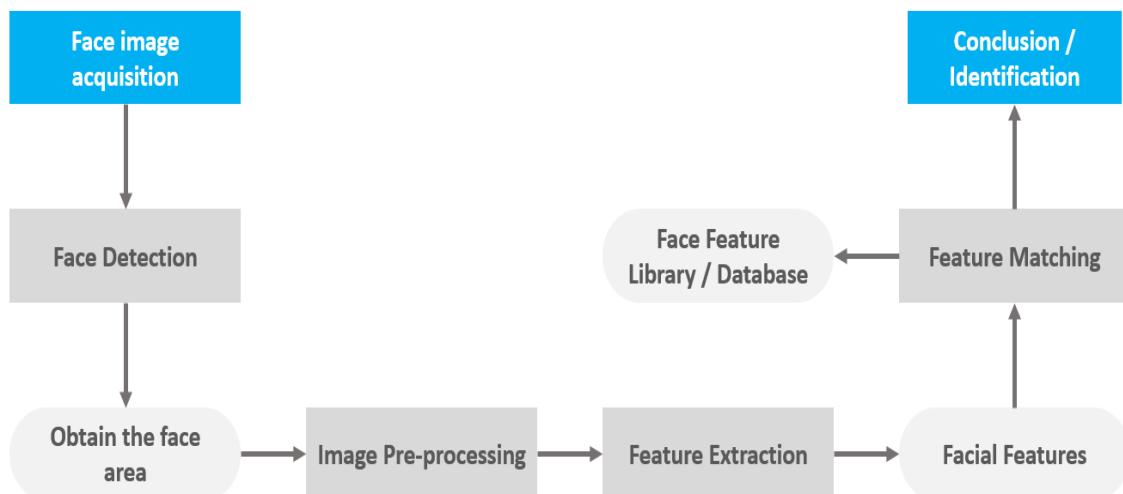


Fig No. 1 : Facial Recognition Process

1.5. About KNN Classifier Facial Recognition Algorithm using Scikit-learn

The k-Nearest Neighbors (k-NN) classifier algorithm from scikit-learn is often used in facial recognition for several reasons. Firstly, the k-NN algorithm is relatively simple and easy to implement. It does not require complex training procedures or assumptions about the data distribution, making it accessible for both beginners and experienced practitioners. Secondly, facial recognition often involves working with small to medium-sized datasets, and the k-NN algorithm performs well in such scenarios. It can effectively classify new facial images by comparing them to the nearest neighbors in the training set, which is particularly useful when the data exhibits complex or non-linear patterns. Additionally, the k-NN algorithm can handle multi-class classification, allowing it to recognize multiple individuals simultaneously. Finally, scikit-learn provides a well-documented and user-friendly implementation of the k-NN algorithm, along with various tools for data preprocessing, feature extraction, and model evaluation. This makes it convenient to integrate the k-NN classifier into a facial recognition pipeline and experiment with different settings and hyperparameters. Overall, the simplicity, effectiveness on small to medium-sized datasets, and the support provided by scikit-learn make the k-NN classifier a valuable choice for facial recognition applications.

Certainly! To provide a mathematical expression for the k-Nearest Neighbors (k-NN) classifier algorithm in the context of facial recognition, we can describe its basic steps:

1.5.1. Training Phase:

- Let X_{train} be the training dataset consisting of n facial images, each represented as a feature vector x_{train_i} of length d , where i ranges from 1 to n .
- Let y_{train} be the corresponding labels or identities associated with each facial image in X_{train} .
- The k-NN algorithm stores the training dataset ($X_{\text{train}}, y_{\text{train}}$) for future classification.

1.5.2. Classification Phase:

- Given a new facial image represented as a feature vector x_{test} , the k-NN algorithm computes the distance or similarity between x_{test} and all feature vectors in X_{train} using a distance metric.
- The k-NN algorithm selects the k nearest neighbors from X_{train} based on the computed distances or similarities.
- Let NN_k be the set of indices corresponding to the k nearest neighbors.
- The k-NN classifier assigns a label or identity to the new facial image by considering the class labels $y_{\text{train}}[NN_k]$ of the k nearest neighbors. The label can be determined using a majority voting scheme or weighted voting, where each neighbor's vote is weighted based on its proximity to x_{test} .

Mathematically, we can represent the classification decision for a new facial image x_{test} as:

$$y_{\text{test}} = \text{argmax}(C_k)$$

where C_k is a vector of size K , and $C_k[i]$ represents the count or weighted sum of occurrences of each unique class label $y_{\text{train}}[NN_k[i]]$ among the k nearest neighbors.

The distance/similarity computation and the choice of k depend on the specific implementation and problem requirements. It's important to note that feature extraction techniques, such as Principal Component Analysis (PCA) or deep learning-based methods, can be applied to obtain the feature vectors x_{train}_i and x_{test} .

This mathematical formulation demonstrates the basic principles of the k-NN algorithm for facial recognition, where the decision is based on the nearest neighbors in the feature space.

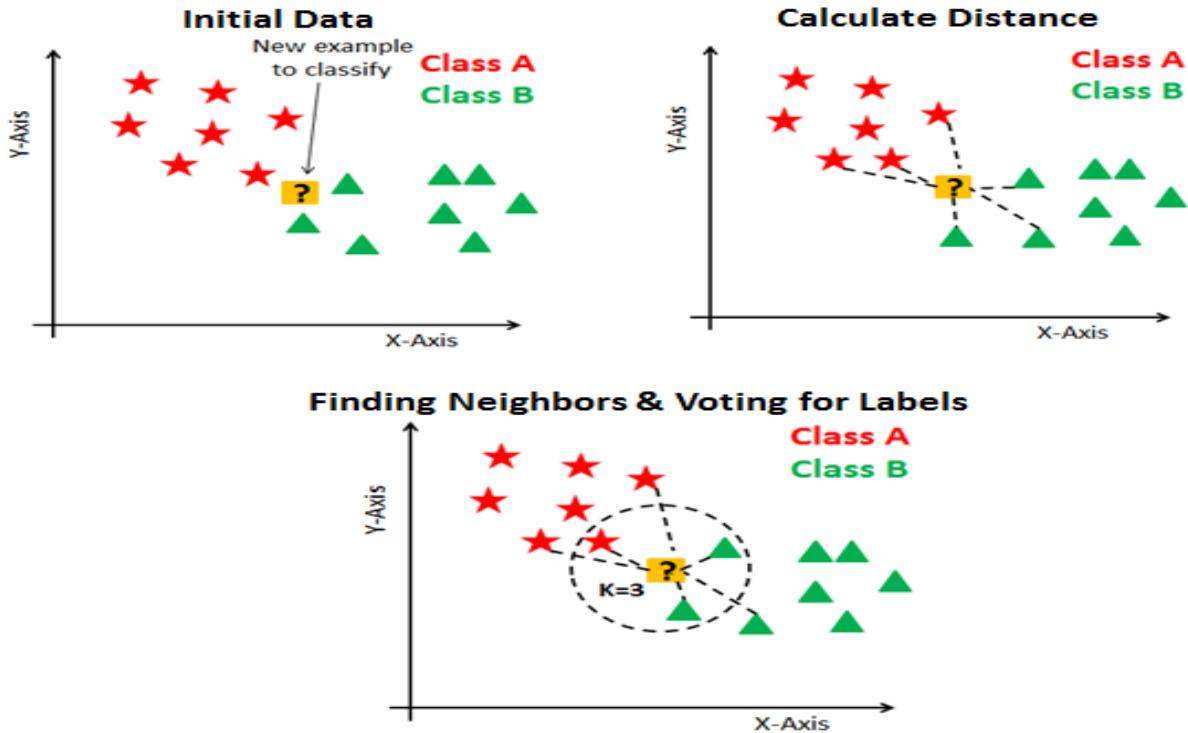


Fig No. 2 : KNN Classification using Scikit-learn

1.6. Problem Statement

The existing attendance management systems in many organizations and educational institutions rely on manual methods such as paper-based registers, swipe cards, or manual entry of attendance data. These traditional methods are prone to errors, time-consuming, and lack accuracy, leading to inefficient and ineffective attendance management. Moreover, they are susceptible to issues like proxy attendance and buddy punching, which compromise the integrity of attendance records.

To address these challenges and improve attendance management, there is a need for an automated system that utilizes face recognition technology. The objective of this project report is to design and develop an attendance management system using face recognition that accurately and efficiently records attendance for individuals in real-time.

The specific problems to be addressed by the project are as follows:

- **Inaccurate Attendance Recording:** The manual methods employed for attendance recording are prone to errors, including incorrect data entry, misplacement of records, and human oversight. These inaccuracies can lead to incorrect attendance records and hinder effective management of attendance data.
- **Time-Consuming Process:** Manual attendance management processes require significant time and effort from administrators, who have to manually collect, compile, and process attendance data. This results in administrative inefficiencies and reduces productivity.
- **Proxy Attendance and Buddy Punching:** Traditional attendance systems are vulnerable to proxy attendance, where someone marks attendance on behalf of another person, and buddy punching, where individuals fraudulently mark attendance for absent colleagues. These practices undermine the reliability and fairness of attendance records.
- **Lack of Real-time Data:** Manual attendance systems often suffer from delays in updating and accessing attendance data. This lack of real-time information hampers the ability to monitor attendance patterns, identify trends, and make timely decisions based on attendance data.
- **Limited Scalability:** Traditional attendance methods face limitations when it comes to handling large-scale attendance management, especially in organizations or institutions with a large number of employees or students. These systems struggle to efficiently process and manage attendance data for a significant number of individuals.

The proposed attendance management system using face recognition aims to overcome these challenges by automating the attendance recording process, ensuring accuracy, efficiency, and real-time data access. By leveraging the unique biometric features of individuals' faces, the system aims to provide reliable and secure attendance management, eliminate proxy attendance and buddy punching, reduce administrative burden, and improve overall efficiency in attendance monitoring and analysis.

2. FEASIBILITY STUDY

The feasibility study aims to assess the viability and practicality of implementing an Attendance Management System using Face Recognition. The system utilizes face recognition technology to accurately record and manage attendance in organizations and educational institutions. This study examines various aspects, including technical feasibility, economic feasibility, operational feasibility, and security and privacy feasibility, to determine the potential success and benefits of the project.

2.1. Technical Feasibility:

Technical feasibility evaluates the technical requirements and capabilities of implementing the Face Recognition Attendance Management System. Key considerations include:

- Face Recognition Algorithms and Technologies:**

Assess the availability and suitability of face recognition algorithms and technologies. Determine whether reliable and accurate face recognition algorithms are readily accessible for integration into the system.

- Hardware Resources:**

Evaluate the hardware requirements, such as cameras and processing power, necessary for capturing and analyzing facial images. Ensure that the organization has the required infrastructure or can acquire it within a reasonable budget.

- Compatibility and Integration:**

Examine the compatibility of the system with existing hardware, software, and databases. Determine if the system can be seamlessly integrated with the organization's infrastructure and systems.

- **Robustness and Reliability:**

Evaluate the robustness and reliability of the face recognition system. Consider factors such as accuracy, speed, and the ability to handle varying lighting conditions, pose variations, and occlusions.

2.2. Economic Feasibility:

Economic feasibility assesses the financial aspects of implementing the Face Recognition Attendance Management System. Key considerations include:

- **Acquisition Costs:**

Calculate the costs associated with acquiring the necessary hardware, including cameras, servers, networking equipment, and face recognition software licenses.

- **Development and Implementation Costs:**

This includes hiring skilled professionals or outsourcing development, customization, and integration with existing systems.

- **Operational Costs:**

Assess the ongoing operational costs, such as maintenance, updates, and support. Consider factors like software updates, hardware maintenance, and training costs for administrators and users.

- **Cost Savings:**

Identify potential cost savings resulting from the implementation of the system. This may include reducing manual attendance management processes, minimizing errors, and optimizing workforce allocation.

2.3. Operational Feasibility:

Operational feasibility assesses the practicality and effectiveness of implementing the Face Recognition Attendance Management System within the organization. Key considerations include:

- **User Acceptance:**

Evaluate the acceptance and willingness of employees, students, or users to adopt the new system. Conduct surveys or interviews to gather feedback and address any concerns or resistance.

- **Training Requirements:**

Identify the training needs for administrators and users to effectively use the system. Determine the resources and time required to train individuals on system operation, face registration, and troubleshooting.

- **Scalability:**

Assess the system's scalability to accommodate the organization's size and growth. Determine if the system can handle a large number of individuals' faces and attendance records without compromising performance.

- **Integration with Existing Processes:**

Evaluate the ease of integrating the Face Recognition Attendance Management System with existing attendance management processes. Consider the impact on workflows, data transfer, and reporting.

2.4. Security and Privacy Feasibility:

Security and privacy considerations are crucial when implementing a face recognition system. Evaluate the feasibility of implementing robust security and privacy measures. Key factors to consider include:

- Data Protection:**

Assess the measures in place to ensure the confidentiality and integrity of biometric data. Consider encryption, access controls, and secure storage practices to protect sensitive information.

- Legal and Regulatory Compliance:**

Ensure compliance with relevant privacy regulations, data protection laws, and industry standards. Identify any legal or ethical considerations associated with collecting, storing, and using biometric data.

- Liveness Detection:**

Assess the system's ability to detect liveness to prevent spoofing or fraudulent attempts.

3. REQUIREMENT SPECIFICATION

We are setting up to design a system comprising two modules. The first module (face detector) is a mobile component, which is basically, a camera application that captures faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is a desktop application that does face recognition of the captured images (faces) in the file and then stores the results in a database for future analysis.

3.1. Hardware Requirements:

3.1.1. Camera:

- Resolution: Minimum 2 megapixels or higher for clear image capture.
- Features: Consider low-light sensitivity or infrared capabilities for improved performance in different lighting conditions.

3.1.2. Computer System:

- Processor: Dual-core processor or higher.
- RAM: Minimum 4 GB RAM, but higher capacity is recommended for better performance.
- Storage: Adequate storage space for face recognition database and related data.
- Operating System: Compatible with the chosen face recognition software and organization's IT infrastructure.

3.1.3. Graphics Processing Unit (GPU):

- Consider a dedicated graphics card or GPU with CUDA cores or similar parallel processing capabilities for accelerated face recognition algorithms.

3.1.4. Network Equipment:

- Network connectivity: Ethernet switches, routers, and cabling for reliable and high-speed communication between system components.

3.1.5. Storage Device:

- Sufficient storage capacity: Choose appropriate hard disk drives (HDDs) or solid-state drives (SSDs) based on storage requirements and system performance needs.

3.1.6. Display and Input Devices:

- Monitor or display screen: Adequate size and resolution for administrators to view attendance data, reports, and system status.
- Keyboard and mouse or other input devices for system administration and configuration.

3.1.7. Power Backup:

- Uninterruptible Power Supply (UPS) or backup power sources to ensure continuous system operation during power outages or fluctuations.

3.2. Software Requirements:

3.2.1. Integrated Development Environment (IDE):

- Visual Studio Code (VS Code): Version 1.78 or higher (or any preferred IDE that supports Python development).

3.2.2. Operating System:

- All operating systems are capable to run application but recommended Windows Operating system XP or higher(36-bit or 64-bit)

3.2.3. Programming Language:

- Python: Version 3.10.11 or higher (used for implementing the core functionalities of the attendance management system).

3.2.4. Web Framework:

- Flask: Version 2.3.2 or higher (used for building the web-based interface and handling HTTP requests and responses).

3.2.5. Web Technologies:

- HTML: Version 5 (used for creating the structure and layout of web pages).
- CSS: Version 3 (used for styling and enhancing the visual appearance of web pages).
- JavaScript: ES6 or higher (used for client-side interactions and dynamic functionalities).
- Bootstrap: Version 5.3 or higher (CSS framework for responsive web design).
- Font Awesome: Version 5.1.0 or higher (icon toolkit for web design).

3.2.6. Python Modules and Libraries:

- OpenCV (cv2): Version 4.7.0 or higher (used for face detection and image processing).
- os: Built-in module (used for interacting with the operating system, file handling, etc.).
- datetime: Built-in module (used for date and time operations).

- numpy: Version 1.24.3 or higher (used for numerical computations and array operations).
- scikit-learn: Version 1.2.2 or higher (used for implementing machine learning algorithms like k-Nearest Neighbors).
- pandas: Version 1.2.5 or higher (used for data manipulation and analysis).
- joblib: Version 2.7 or higher (used for saving and loading trained machine learning models).

3.3. Data Requirements:

3.3.1. User Data:

- Name: The full name of the individual.
- Unique ID: A unique identifier for each individual.

3.3.2. Face Image Data:

- Face Images: A collection of high-quality images for each individual enrolled in the system. These images should capture variations in lighting, pose, and expressions to ensure accurate recognition.
- Image Format: Specify the supported image format, such as JPEG or PNG.

3.3.3. Attendance Records:

- Date and Time: Capture the date and time when an individual's attendance is recorded.
- Attendance Status: Indicate whether the individual is present or absent.
- Unique ID: Associate the attendance record with the corresponding individual.

3.3.4. System Logs:

- Log Entries: Capture system activities, errors, warnings, and any other relevant information.
- Timestamp: Record the date and time of each log entry.
- Log Level: Specify different log levels (e.g., info, warning, error) to categorize log entries.

3.3.5. Configuration Data:

- Face Recognition Algorithm: Specify the algorithm used for face recognition.
- Thresholds: Define the similarity threshold for face matching to determine a positive match.
- Camera Settings: Specify camera resolution, frame rate, and other relevant settings.

3.3.6. Training Data:

- Training Dataset: Specify the dataset used to train and fine-tune the face recognition algorithm.
- Data Size: Indicate the number of images and individuals in the training dataset.

4. METHODOLOGY

To achieve this work, the methodology to be the objectives of adopted is as under:

S.No.	DATE	TASK	STATUS
1.	18-Sept-2022	Requirement Analysis	Done
2.	20-Dec-2022	System Design <ul style="list-style-type: none">● UX/UI Design Prototype● Flow of data● Flowchart and diagram	Done Done Done Done
3.	14-Jan-2023	Data Collection and Preprocessing <ul style="list-style-type: none">● Data Collection● Data Preprocessing● Data Store	Done Done Done Done
4.	22-Feb-2023	Face Recognition Model <ul style="list-style-type: none">● ML Algorithm● Training process	Done Done Done
5.	10-Mar-2023	Enrollment Process	Done
6.	14-Mar-2023	Attendance Tracking	Done
7.	15-Apr-2023	System Integration <ul style="list-style-type: none">● User Interface Integration● Camra and hardware Integration● Real-Time Face Detection and Recognition Integration	Done Done Done Done
8.	8-May-2023	Testing and Evaluation	Done

Table no.1 : Methodology Table

\

4.1. Requirement analysis:

- Requirement analysis is a crucial step in developing an Attendance Management System using Face Recognition. It involves identifying and documenting the needs and expectations to ensure that the system meets its requirements effectively.

4.2. System Design:

- Discussed the overall system architecture, including the components, their interactions, and the flow of data.
- The design decisions made regarding the choice of face recognition algorithms, camera setup, database integration, and user interface design.
- Including relevant diagrams or flowcharts to illustrate the system design.

4.3. Data Collection and Preprocessing:

4.3.1. Data Collection:

- Set up a camera or cameras in an appropriate location to capture face images.
- Ensure proper lighting conditions and consider capturing images under different lighting scenarios.
- Instruct individuals to face the camera directly, with minimal head tilts or angles.
- Capture multiple images of each individual to account for variations in expressions and poses.
- Store the captured face images in a suitable format, such as JPEG or PNG, for further processing.

4.3.2. Data Preprocessing:

- Resize and Crop: Resizing the captured face images to a standard size for consistency. Cropped the images to focus on the facial region, removing any unnecessary background.

- Normalization: Applying normalization techniques to standardize the intensity or color distribution across the face images. This helps to mitigate variations caused by lighting conditions.
- Data Labeling: Associating relevant information with each face image, such as the individual's name or ID, to establish a connection between the face and the corresponding identity.

4.4. Face Recognition Model:

- Using K-NN with scikit-learn and train a face recognition algorithm using the collected face image dataset.

4.5. Enrollment Process:

- Enrolling individuals into the system, including capturing face images, face detection, and face representation extraction.
- Enrolled data, including the associated user information, is stored in the database.

4.6. Attendance Tracking:

- The process of capturing images in real-time using the system's camera.
- Face detection, face recognition, and attendance recording, including date and time stamping.

4.7. System integration:

- Integrating the Attendance Management System with the database for storing user data, attendance records, and system logs.
- Integrating the user interface for seamless access to attendance-related functions.

4.8. Testing and Evaluation:

- Conducted comprehensive testing to ensure system functionality, accuracy, and performance.

- Evaluating the system's performance.

S.No.	TASK	STATUS
1.	Running Flask server	Pass
2.	Loading templates on localhost:5000	Pass
3.	Adding user data successfully	Pass
4.	Video frame web-camra on	Pass
5.	50 face snapshots	Pass
6.	Creating user face folder successfully	Pass
7.	Attendance marking web-camra on	Pass
8.	Automatic user face recognition & detectioin	Pass
9.	Marking attendance in real-time	Pass
10.	Correctly showing date and time in application	Pass
11.	Taking attendance one by one user	Pass
12.	Taking attendance more than one user in single frame and time (min 2 user face)	Pass
13.	Generating Attendance CSV file	Pass
14.	View database working	Pass
15.	Delete user action working	Pass
16.	Operating manually database & application	Pass

Table no.2 : Testing Application Performance

5. SYSTEM DESIGN DOCUMENTATION

5.1. Flow Chart:

This is the flow chart of the application working process. Firstly when application runs on the server the application takes image as input for the server and Face recognition and detection working start its process and store in the database. Then at timestamp marking attendance recognizes face from database and mark real-time attendance.

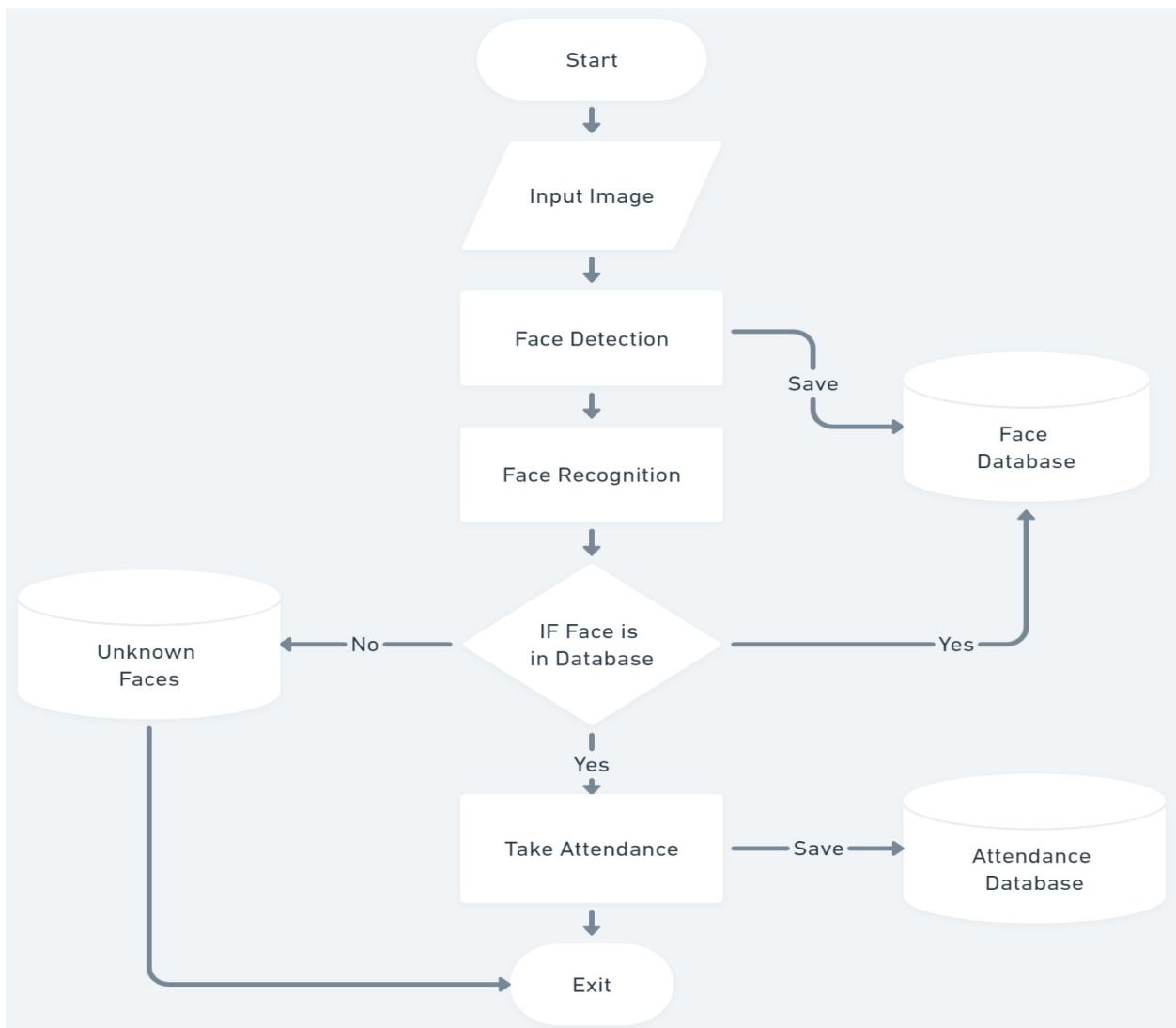


Fig No. 3 : Flow Chart

5.2. Context Diagram:

In the context of an Attendance Management System using Face Recognition, a context diagram would illustrate the system's interaction with external entities and the flow of data between them. The context diagram illustrates the boundaries of the Attendance Management System and its interactions with the external entities. It shows that the system interacts with users for enrollment, attendance tracking, and report generation. It also demonstrates the system's dependence on the camera system for capturing face images and the database for storing and retrieving data. The reporting system receives data from the Attendance Management System to generate attendance-related reports and analytics.

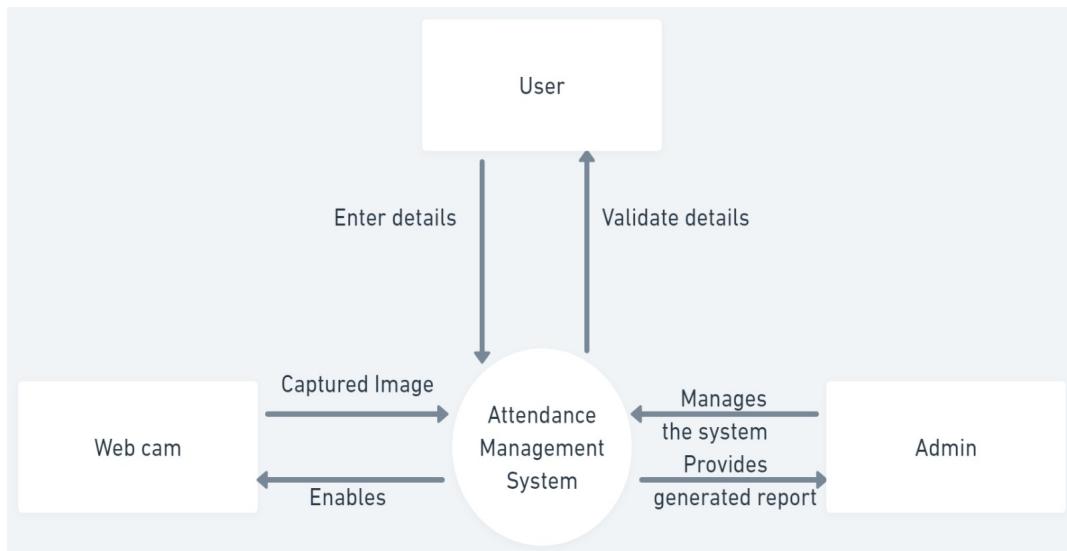


Fig No. 4 : Context Diagram

5.2. Component Diagram:

In a component diagram for a Face Recognition application, the components involved in the face recognition process and their relationships can be represented.

- **Image Capture Component:** This component is responsible for capturing face images either from a camera or from pre-existing images. It acquires the raw image data needed for further processing.

- **Data Augmentation Component:** This component applies various transformations and modifications to the captured face images to augment the training dataset. It performs operations like rotation, translation, flipping, cropping, and noise addition to create additional training examples with variations.
- **Database Component:** The database component represents the storage and retrieval of data within the face recognition system. It stores the captured and augmented face images, along with associated metadata and labels.
- **Trained Model Component:** This component represents the trained face recognition model, which is a result of the training process. It contains learned parameters and representations that enable accurate face recognition.

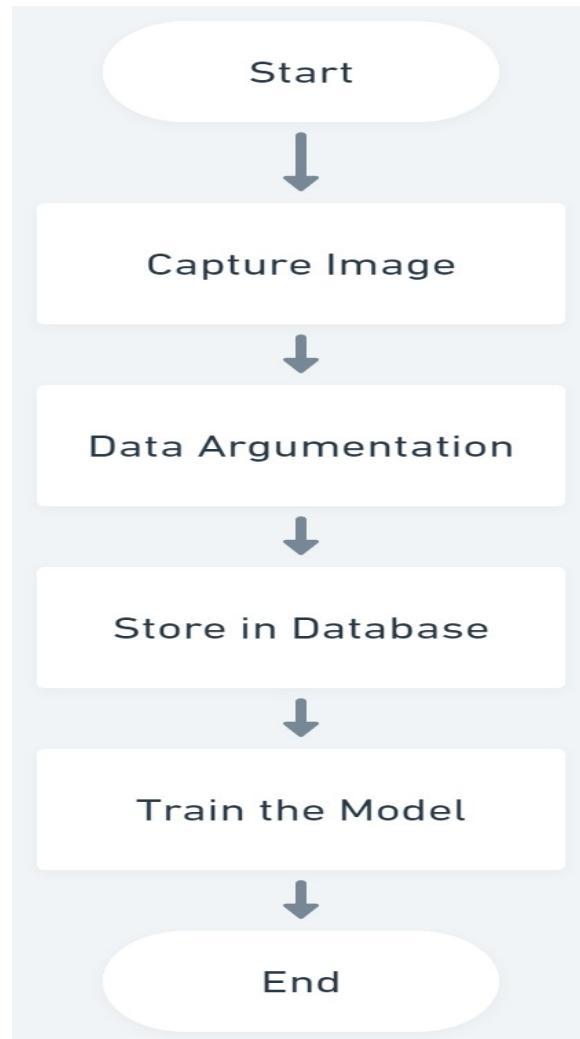


Fig No. 5 : Component Diagram

5.2. Data Flow Diagram:

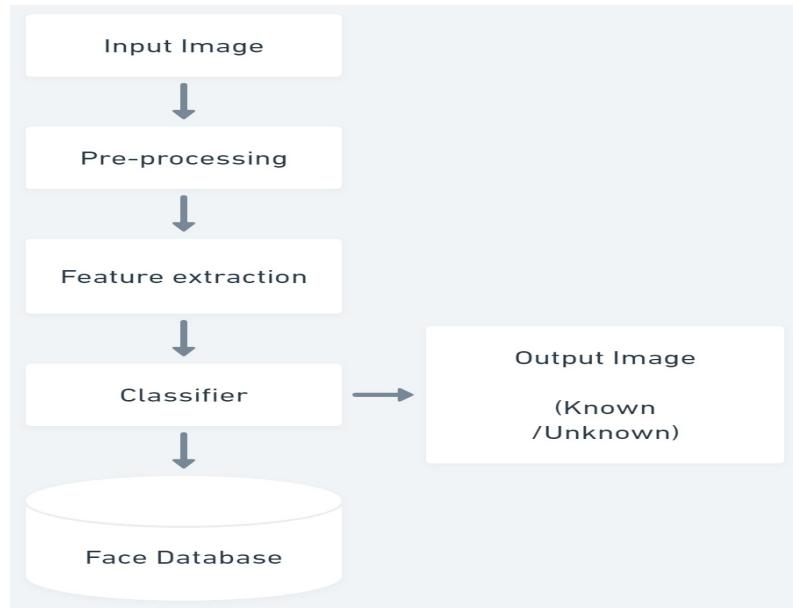


Fig No. 6 : DFD Face Recognition

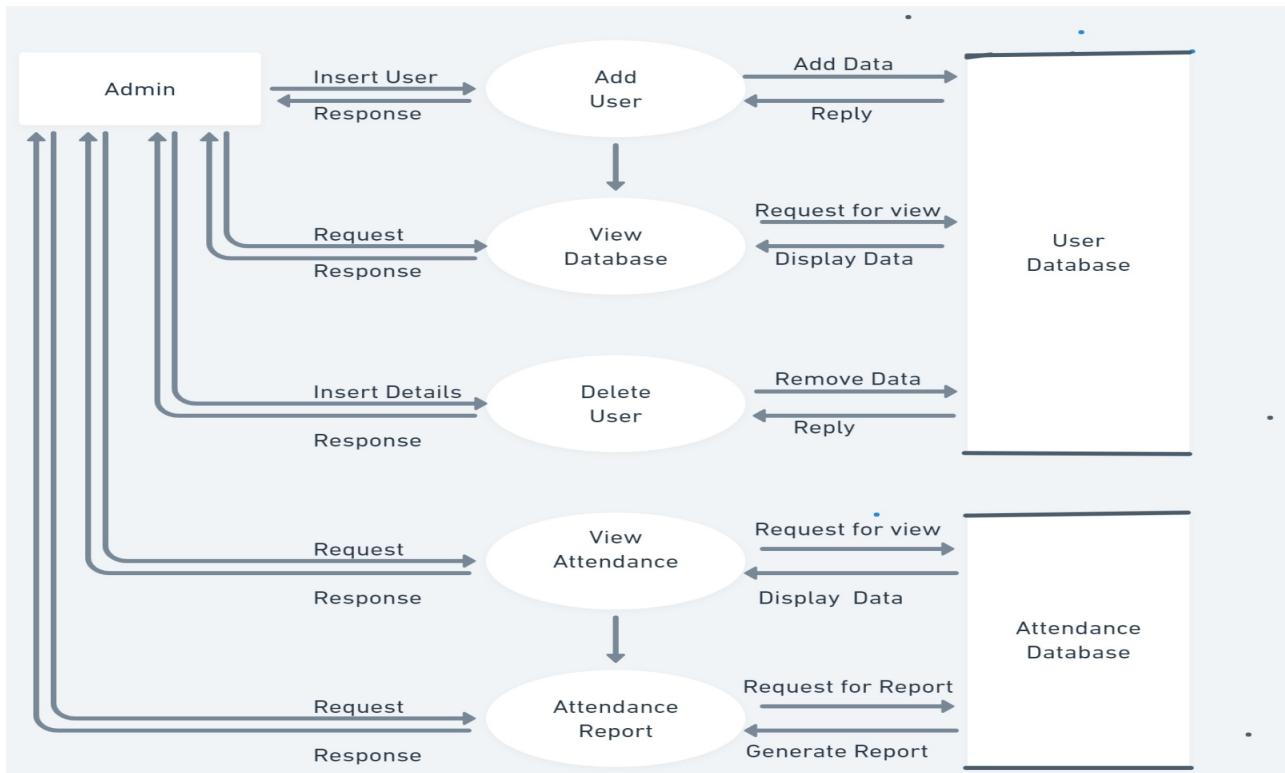


Fig No. 7 : DFD Admin Portal

6. SYSTEM IMPLEMENTATION AND CODING

6.1. Flask Code(Python): app.py

```
import cv2
import os
from flask import Flask,request,render_template, redirect,url_for,send_file
from datetime import date
from datetime import datetime
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import joblib
import shutil
##### Defining Flask App
app = Flask(__name__)

##### Saving Date today in 2 different formats
datetoday = date.today().strftime("%m_%d_%y")
datetoday2 = date.today().strftime("%d-%B-%Y")

##### Initializing VideoCapture object to access WebCam
face_detector =
cv2.CascadeClassifier('static/haarcascade_frontalface_default.xml')
cap = cv2.VideoCapture(0)

##### If these directories don't exist, create them
if not os.path.isdir('Attendance'):
    os.makedirs('Attendance')
if not os.path.isdir('static'):
    os.makedirs('static')
if not os.path.isdir('static/faces'):
    os.makedirs('static/faces')
if f'Attendance-{datetoday}.csv' not in os.listdir('Attendance'):
    with open(f'Attendance/Attendance-{datetoday}.csv','w') as f:
        f.write('Name,Roll,Time')
```

```

##### get a number of total registered users
def totalreg():
    return len(os.listdir('static/faces'))


##### extract the face from an image
def extract_faces(img):
    if img!=[]:
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        face_points = face_detector.detectMultiScale(gray, 1.3, 5)
        return face_points
    else:
        return []

##### Identify face using ML model
def identify_face(facearray):
    model = joblib.load('static/face_recognition_model.pkl')
    return model.predict(facearray)


##### A function which trains the model on all the faces available in faces
##### folder
def train_model():
    faces = []
    labels = []
    userlist = os.listdir('static/faces')
    for user in userlist:
        for imgname in os.listdir(f'static/faces/{user}'):
            img = cv2.imread(f'static/faces/{user}/{imgname}')
            resized_face = cv2.resize(img, (50, 50))
            faces.append(resized_face.ravel())
            labels.append(user)
    faces = np.array(faces)
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(faces,labels)
    joblib.dump(knn,'static/face_recognition_model.pkl')


##### Extract info from today's attendance file in attendance folder
def extract_attendance():
    df = pd.read_csv(f'Attendance/Attendance-{datetoday}.csv')
    names = df['Name']
    rolls = df['Roll']
    times = df['Time']
    l = len(df)

```

```

    return names,rolls,times,l

#### Add Attendance of a specific user
def add_attendance(name):
    username = name.split('_')[0]
    userid = name.split('_')[1]
    current_time = datetime.now().strftime("%H:%M:%S")

    df = pd.read_csv(f'Attendance/Attendance-{datetoday}.csv')
    if int(userid) not in list(df['Roll']):
        with open(f'Attendance/Attendance-{datetoday}.csv','a') as f:
            f.write(f'\n{username},{userid},{current_time}')

#####
##### ROUTING FUNCTIONS #####
#####

##### Our main page
@app.route('/')
def home():
    return render_template('index.html')
# names,rolls,times,l = extract_attendance()
# return
render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=datetoday2)

@app.route('/application')
def application():
    names,rolls,times,l = extract_attendance()
    return
render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=datetoday2)

#####
##### This function will run when we click on Take Attendance Button
@app.route('/start',methods=['GET'])
def start():
    if 'face_recognition_model.pkl' not in os.listdir('static'):
        return
render_template('home.html',totalreg=totalreg(),datetoday2=datetoday2,mess='There is no trained model in the static folder. Please add a new face to continue.')

    cap = cv2.VideoCapture(0)
    ret = True
    while ret:

```

```

ret,frame = cap.read()
if extract_faces(frame)!=():
    (x,y,w,h) = extract_faces(frame)[0]
    cv2.rectangle(frame,(x, y), (x+w, y+h), (255, 0, 20), 2)
    face = cv2.resize(frame[y:y+h,x:x+w], (50, 50))
    identified_person = identify_face(face.reshape(1,-1))[0]
    add_attendance(identified_person)

cv2.putText(frame,f'{identified_person}',(30,30),cv2.FONT_HERSHEY_SIMPLEX,1,(25
5, 0, 20),2, cv2.LINE_AA)
cv2.imshow('Attendance',frame)
if cv2.waitKey(1)==27:
    break
cap.release()
cv2.destroyAllWindows()
names,rolls,times,l = extract_attendance()
return
render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=datetoday2)

```

```

##### This function will run when we add a new user
@app.route('/add',methods=['GET','POST'])
def add():
    newusername = request.form['newusername']
    newuserid = request.form['newuserid']
    userimagefolder = 'static/faces/'+newusername+'_'+str(newuserid)
    if not os.path.isdir(userimagefolder):
        os.makedirs(userimagefolder)
    cap = cv2.VideoCapture(0)
    i,j = 0,0
    while 1:
        _,frame = cap.read()
        faces = extract_faces(frame)
        for (x,y,w,h) in faces:
            cv2.rectangle(frame,(x, y), (x+w, y+h), (255, 0, 20), 2)
            cv2.putText(frame,f'Images Captured:
{i}/50',(30,30),cv2.FONT_HERSHEY_SIMPLEX,1,(255, 0, 20),2, cv2.LINE_AA)
            if j%10==0:
                name = newusername+'_'+str(i)+'.jpg'
                cv2.imwrite(userimagefolder+'/'+name,frame[y:y+h,x:x+w])
                i+=1
            j+=1
        if j==500:
            break
    cv2.imshow('Adding new User',frame)

```

```

        if cv2.waitKey(1)==27:
            break
        cap.release()
        cv2.destroyAllWindows()
        print('Training Model')
        train_model()
        names,rolls,times,l = extract_attendance()
        return
    render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=tot
talreg(),datetoday2=datetoday2)

@app.route('/delete', methods=['POST'])
def delete():
    username = request.form['username']
    userid = request.form['userid']
    userimagefolder = f'static/faces/{username}_{userid}'
    if os.path.isdir(userimagefolder):
        # Delete the user's image folder
        shutil.rmtree(userimagefolder)
        # Retrain the model after deleting the user
        train_model()
        # Redirect to the home page
        return redirect(url_for('application'))
    else:
        # Redirect to the home page with an error message
        return redirect(url_for('application', error='User does not exist.'))

# @app.route('/database')
# def view_database():
#     users = []
#     userlist = os.listdir('static/faces')
#     for user in userlist:
#         user_images = []
#         for imgname in os.listdir(f'static/faces/{user}'):
#             user_images.append(imgname)
#         users.append({'user': user, 'images': user_images})
#     return render_template('database.html', users=users)

@app.route('/database')
def view_database():
    users = []
    userlist = os.listdir('static/faces')

```

```

for user in userlist:
    username, userid = user.split('_')

    users.append({'username': username, 'userid': userid})
return render_template('home.html', users=users)

@app.route('/download')
def download():
    # Path to the generated CSV file
    csv_path = f'Attendance/Attendance-{datetoday}.csv'

    # Return the file for download
    return send_file(csv_path, as_attachment=True)

#### Our main function which runs the Flask App
if __name__ == '__main__':
    app.run(debug=True)

```

6.2. Html File : index.html

```

<!DOCTYPE html>
<html lang="">
<head>
<title>Foxclore</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no">
<link href="/static/layout/styles/layout.css" >
<style>
@charset "utf-8";
@import url("/static/layout/styles/fontawesome-free/css/all.css");
@import url("/static/layout/styles/framework.css");

/* Rows
-----
----- */
.row0, .row0 a{}
.row1, .row1 a{}
.row2, .row2 a{}
.row3, .row3 a{}
.row4, .row4 a{}
.row5, .row5 a{}

```

```

/* Header
-----
----- */
#header{padding:50px 0;}

#header #logo *{margin:0; padding:0; line-height:1;}
#header #logo h1{font-size:32px;}
#header #logo p{font-size:.8rem;}

#header ul{}
#header ul li{}
#header ul li div{position:relative; min-height:45px; padding:0 0 0 60px;
line-height:1; word-wrap:break-word;}
#header ul li div i{position:absolute; top:0; left:0; width:45px; height:45px;
line-height:45px; font-size:16px; text-align:center; border-radius:50%;}
#header ul li div span{display:block; padding:4px 0 0 0; }
#header ul li div strong{display:block; margin:0 0 8px 0; }

#header div:last-child{margin-bottom:0;}/* Used when elements stack in small
viewports */

/* Navwrapper
-----
----- */
#navwrapper{padding:0 0 0 20px; }

#navwrapper #searchform{display:block; position:relative; float:right;
height:58px; line-height:1; z-index:999; border-radius:10px 0; }
#navwrapper #searchform::after{display:block; height:58px; line-height:58px;
padding:0 15px; font-size:20px; font-family:"Font Awesome 5 Free";
font-weight:900; content:"\f002"/* fa-search */; }
#navwrapper #searchform:hover::after, #navwrapper #searchform
button{cursor:pointer; }
#navwrapper #searchform > div{position:absolute; top:58px; right:0;
width:280px; visibility:hidden; opacity:0/* Required for transition effect */; }
#navwrapper #searchform:hover > div{visibility:visible; opacity:1/* Required
for transition effect */; }
#navwrapper #searchform form{display:block; position:relative; }
#navwrapper #searchform form *{border:none; }
#navwrapper #searchform input{display:block; width:100%; height:36px;
padding:8px 40px 8px 10px; font-size:1rem; }
#navwrapper #searchform ::placeholder{text-transform:capitalize; }
#navwrapper #searchform button{display:block; position:absolute; top:3px;
right:2px; height:30px; font-size:16px; line-height:1; }

/* Page Intro
-----
----- */
#pageintro{padding:80px 0; }

```

```

#pageintro article{display:block; max-width:55%;}
#pageintro .heading{margin-bottom:20px; font-size:4rem;}
#pageintro p:first-of-type{margin:0 0 20px 0; text-transform:uppercase;
font-size:1.6rem; letter-spacing:2px;}
#pageintro p:last-of-type{font-size:1.2rem;}
#pageintro footer{margin-top:50px;}

/* Content Area
-----
*/
.container{padding:80px 0; }

/* Content */
.container .content{}

.sectiontitle{display:block; max-width:55%; margin:0 auto 80px;
text-align:center;}
.sectiontitle,.heading{margin-bottom:20px; font-size:4rem;}
.sectiontitle *{margin:0; }

/* Introblocks */
.introblocks{display:block; position:relative; margin:-150px 0 0 0;
z-index:1;}/* 180px => 100px + container padding */
.introblocks > ul > li{margin-bottom:50px;}
.introblocks > ul > li:nth-last-child(-n+3){margin-bottom:0;}/* Removes bottom
margin from the last line of items - margin is restored in the media queries
when items stack */
.introblocks > ul > li:nth-child(3n+1){margin-left:0; clear:left;}/* Removes
the need to add class="first" */
.introblocks figure{}
.introblocks figure figcaption{display:block; padding:30px 0 0 0;}
.introblocks figure figcaption > a{}
.introblocks figure figcaption .heading{font-size:1.2rem; font-weight:700;
margin-bottom:0; }

/* Stats */
#stats{}
#stats > li{display:block; float:left; width:25%; border:solid; border-width:0
1px 0 0; text-align:center;}
#stats > li:last-child{border:none;}
#stats > li *{margin:0; padding:0;}
#stats > li i{margin-bottom:30px; font-size:5rem;}
#stats > li p:first-of-type{margin-bottom:15px; font-size:3rem;
font-weight:700; line-height:1;}
#stats > li p:last-of-type{text-transform:uppercase; }

/* Latest */
.latest{margin-bottom:80px;}
.latest > li:last-child{margin-bottom:0;}/* Used when elements stack in small
viewports */
.latest article{position:relative; padding:0 0 0 80px; }

```

```

.latest article time{display:block; position:absolute; top:0; left:0;
width:60px;}
.latest article time *{display:block; margin:0; padding:0; text-align:center;
text-transform:uppercase; font-style:normal; line-height:1;}
.latest article time strong{padding:12px 0; font-size:18px; border-bottom:1px
solid; opacity:.5;}
.latest article time em{padding:8px 0;}
.latest article .excerpt .meta li{display:inline-block; font-size:.8rem;}
.latest article .excerpt .meta li::after{margin-left:5px; content:"|";}
.latest article .excerpt .meta li:last-child::after{margin:0; content:"";}
.latest article .excerpt .heading{margin:0; font-size:1.2rem;}
.latest article .excerpt p{margin-bottom:30px;}

/* Comments */
#comments ul{margin:0 0 40px 0; padding:0; list-style:none;}
#comments li{margin:0 0 10px 0; padding:15px;}
#comments .avatar{float:right; margin:0 0 10px 10px; padding:3px; border:1px
solid;}
#comments address{font-weight:bold;}
#comments time{font-size:smaller;}
#comments .comcont{display:block; margin:0; padding:0;}
#comments .comcont p{margin:10px 5px 10px 0; padding:0; }

#comments form{display:block; width:100%;}
#comments input, #comments textarea{width:100%; padding:10px; border:1px
solid;}
#comments textarea{overflow:auto;}
#comments div{margin-bottom:15px;}
#comments input[type="submit"], #comments
input[type="reset"]{display:inline-block; width:auto; min-width:150px;
margin:0; padding:8px 5px; cursor:pointer; }

/* Sidebar */
.container .sidebar{}

.sidebar .sdb_holder{margin-bottom:50px;}
.sidebar .sdb_holder:last-child{margin-bottom:0;}

/* Testimonials
-----
*/
#testimonials{}

#testimonials article{text-align:center;}
#testimonials article *{margin:0; padding:0;}
#testimonials article img{margin-bottom:20px; border-radius:50%;}
#testimonials article blockquote{margin-bottom:20px; padding:20px;}
#testimonials article blockquote::before{top:5px; left:5px; font-size:30px;
line-height:30px;}
#testimonials article .heading{font-size:1.4rem;}
#testimonials article em{display:block; margin-top:-5px; font-size:.8rem;
font-style:normal;}

```

```

/* Footer
-----
#footer{padding:80px 0;}

#footer .heading{margin-bottom:50px; font-size:1.2rem;}

#footer > div:first-of-type .heading{margin-bottom:20px;
text-transform:uppercase; font-size:1.8rem;}
#footer > div:first-of-type .faico{margin-bottom:30px;}

#footer input, #footer button{border:1px solid;}
#footer input{display:block; width:100%; padding:8px;}
#footer button{padding:8px 18px 10px; text-transform:uppercase;
font-weight:400; cursor:pointer;}

#footer .linklist li{display:block; margin-bottom:15px; padding:0 0 15px 0;
border-bottom:1px solid;}
#footer .linklist li:last-child{margin:0; padding:0; border:none;}
#footer .linklist li::before, #footer .linklist li::after{display:table;
content:"";}
#footer .linklist li, #footer .linklist li::after{clear:both;}

#footer .contact{}
#footer .contact.linklist li, #footer .contact.linklist
li:last-child{position:relative; padding-left:40px;}
#footer .contact li *{margin:0; padding:0; line-height:1.6;}
#footer .contact li i{display:block; position:absolute; top:0; left:0;
width:30px; font-size:16px; text-align:center;}

#footer .latestimg{}
#footer .latestimg > li{display:inline-block; float:left; width:30%; margin:0 0
5% 5%;}
#footer .latestimg > li:nth-last-child(-n+3){margin-bottom:0;}/* Removes bottom
margin from the last three items - margin is restored in the media queries when
items stack */
#footer .latestimg > li:nth-child(3n+1){margin-left:0; clear:left;}/* Removes
the need to add class="first" */
#footer .latestimg > li img{width:100%;}/* Force the image to resize to take
the full space - may have to be changed for tablets, depends on personal
preference */
#footer .latestimg > li a.imgover{display:block;}

/* Copyright
-----
#copyright{padding:20px 0;}
#copyright *{margin:0; padding:0;}
```

```

/* Transition Fade
This gives a smooth transition to "ALL" elements used in the layout - other
than the navigation form used in mobile devices
If you don't want it to fade all elements, you have to list the ones you want
to be faded individually
Delete it completely to stop fading
-----
----- */
*,:before,:after{transition:all .3s ease-in-out;}
#mainnav form *{transition:none !important;}

/*
-----
----- */
/*
-----
----- */
/*
-----
----- */
/*
-----
----- */
/*
-----
----- */
/*
-----
----- */

/* Navigation
-----
----- */
nav ul, nav ol{margin:0; padding:0; list-style:none;}

#mainnav, #breadcrumb, .sidebar nav{line-height:normal;}
#mainnav .drop::after, #mainnav li li .drop::after, #breadcrumb li a::after,
.sidebar nav a::after{position:absolute; font-family:"Font Awesome 5 Free";
font-weight:900; font-size:10px; line-height:10px; }

/* Top Navigation */
#mainnav{display:block; float:left;}
#mainnav ul{text-transform:uppercase;}
#mainnav ul ul{z-index:9999; position:absolute; width:180px;
text-transform:none; text-align:left;}
#mainnav ul ul ul{left:180px; top:0;}
#mainnav li{display:block; float:left; position:relative; margin:0; padding:0;}
#mainnav li:last-child{margin-right:0;}
#mainnav li li{width:100%; margin:0;}
#mainnav li a{display:block; padding:20px;}
#mainnav li li a{border:solid; border-width:0 0 1px 0;}
#mainnav .drop{padding-left:15px;}
#mainnav li li a, #mainnav li li .drop{display:block; margin:0; padding:10px
15px;}

```

```

#mainnav .drop::after, #mainnav li li .drop::after{content:"\f0d7";}
#mainnav .drop::after{top:25px; left:5px;}
#mainnav li li .drop::after{top:15px; left:5px;}
#mainnav ul ul{visibility:hidden; opacity:0;}
#mainnav ul li:hover > ul{visibility:visible; opacity:1;}

#mainnav form{display:none; width:100%; margin:0; padding:0;}
#mainnav form select, #mainnav form select option{display:block; cursor:pointer;
outline:none;}
#mainnav form select{width:100%; padding:5px; border:none;}
#mainnav form select option{margin:5px; padding:0; border:none; }

/* Breadcrumb */
#breadcrumb{padding:150px 0 30px;}
#breadcrumb ul{margin:0; padding:0; list-style:none; text-transform:uppercase;}
#breadcrumb li{display:inline-block; margin:0 6px 0 0; padding:0;}
#breadcrumb li a{display:block; position:relative; margin:0; padding:0 12px 0
0; font-size:12px;}
#breadcrumb li a::after{top:4px; right:0; content:"\f0da";}
#breadcrumb li:last-child a{margin:0; padding:0;}
#breadcrumb li:last-child a::after{display:none;}
#breadcrumb .heading{margin:0; font-size:2rem;}

/* Sidebar Navigation */
.sidebar nav{display:block; width:100%;}
.sidebar nav li{margin:0 0 3px 0; padding:0;}
.sidebar nav a{display:block; position:relative; margin:0; padding:5px 10px 5px
15px; text-decoration:none; border:solid; border-width:0 0 1px 0;}
.sidebar nav a::after{top:10px; left:5px; content:"\f0da";}
.sidebar nav ul ul a{padding-left:35px;}
.sidebar nav ul ul a::after{left:25px;}
.sidebar nav ul ul ul a{padding-left:55px;}
.sidebar nav ul ul ul a::after{left:45px; }

/* Pagination */
.pagination{display:block; width:100%; text-align:center; clear:both;}
.pagination li{display:inline-block; margin:0 2px 0 0;}
.pagination li:last-child{margin-right:0;}
.pagination a, .pagination strong{display:block; padding:8px 11px; border:1px
solid; background-clip:padding-box; font-weight:normal; }

/* Back to Top */
#backtotop{z-index:999; display:inline-block; position:fixed;
visibility:hidden; bottom:20px; right:20px; width:36px; height:36px;
line-height:36px; font-size:16px; text-align:center; opacity:.2;}
#backtotop i{display:block; width:100%; height:100%; line-height:inherit;}
#backtotop.visible{visibility:visible; opacity:.5;}
#backtotop:hover{opacity:1; }

/* Tables
-----
----- */

```

```

table, th, td{border:1px solid; border-collapse:collapse; vertical-align:top;}
table, th{table-layout:auto;}
table{width:100%; margin-bottom:15px;}
th, td{padding:5px 8px;}
td{border-width:0 1px;}

/* Gallery
-----
----- */
#gallery{display:block; width:100%; margin-bottom:50px;}
#gallery figure figcaption{display:block; width:100%; clear:both;}
#gallery li{margin-bottom:30px;}

/* Font Awesome Social Icons
-----
----- */
.faico{margin:0; padding:0; list-style:none;}
.faico li{display:inline-block; margin:8px 5px 0 0; padding:0;
line-height:normal;}
.faico li:last-child{margin-right:0;}
.faico a{display:inline-block; width:36px; height:36px; line-height:36px;
font-size:18px; text-align:center; }

.faico a{color:#FFFFFF; background-color:#2A2D34;}
.faico a:hover{}

.faicon-dribbble:hover{background-color:#EA4C89;}
.faicon-facebook:hover{background-color:#3B5998;}
.faicon-google-plus:hover{background-color:#DB4A39;}
.faicon-linkedin:hover{background-color:#0E76A8;}
.faicon-twitter:hover{background-color:#00ACEE;}
.faicon-vk:hover{background-color:#4E658E;}

/*
-----
----- */
/*
----- */
/*
----- */
/*
----- */
/*
----- */
/*
----- */
/*
----- */

```

```

/* Colours
-----
----- */

body{color:#474747; background-color:#FFFFFF;}
a{color:#41C379;}
a:active, a:focus{background:transparent;}/* IE10 + 11 Bugfix - prevents grey
background */
hr, .borderedbox{border-color:#D7D7D7;}
label span{color:#FF0000; background-color:inherit;}
input:focus, textarea:focus, *:required:focus{border-color:#41C379;}

.overlay{color:#FFFFFF; background-color:inherit;}
.overlay::after{color:inherit; background-color:rgba(0,0,0,.55);}
.overlay.light{color:#474747;}
.overlay.light::after{background-color:rgba(255,255,255,.7);}

.btn, .btn.inverse:hover{color:#FFFFFF; background-color:#41C379;
border-color:#41C379;}
.btn:hover, .btn.inverse{color:inherit; background-color:transparent;
border-color:inherit; }

.imgover:hover::before{background-color:rgba(0,0,0,.55);}
.imgover, .imgover:hover::after{color:#41C379;}

/* Rows */
.row0, .row0 a{}
.row1, .row1 a{}
.row2{color:#474747; background-color:#F4F4F4;}
.row3{color:#474747; background-color:#FFFFFF;}
.row4{color:#C1C2C4; background-color:#323639;}
.row5, .row5 a{color:#C1C2C4; background-color:#2A2D34; }

.coloured{color:#FFFFFF; background:linear-gradient(90deg, #41C379, #37AE6A);}

/* Header */
#header #logo a{color:inherit;}
#header ul li div i{background-color:#FFFFFF;}
#header ul li div:hover i{color:#FFFFFF; background-color:#41C379; }

/* Navwrapper */
#navwrapper{color:#FFFFFF; background-color:rgba(0,0,0,.35);}
#navwrapper #searchform{color:#41C379; background:#FFFFFF;}
#navwrapper #searchform:hover{color:#41C379;}
#navwrapper #searchform input{color:#FFFFFF; background:#41C379;}
#navwrapper #searchform button{color:#FFFFFF; background:transparent; }

/* Page Intro */
#pageintro{color:#FFFFFF;}
```

```

/* Content Area */
#stats > li{border-color:#D7D7D7;}
#stats > li i{color:rgba(0,0,0,.2);}
#stats > li:hover i{color:rgba(0,0,0,.6);}

.latest article time{color:#474747; background-color:#F4F4F4;}
.latest article time strong{border-color:rgba(0,0,0,.2);}

/* Testimonials */
#testimonials article blockquote{color:#474747; background-color:#FFFFFF;}
#testimonials article blockquote::before{color:rgba(0,0,0,.1);}

/* Footer */
#footer .heading{color:#FFFFFF;}
#footer hr, #footer .borderedbox, #footer .linklist
li{border-color:rgba(255,255,255,.1);}

#footer input, #footer button{border-color:transparent;}
#footer input{color:#474747; background-color:#FFFFFF;}
#footer input:focus{border-color:#41C379;}
#footer button{color:#FFFFFF; background-color:#41C379;}
#footer button:hover{color:inherit; background-color:transparent;
border-color:inherit;}

/* Navigation */
#mainnav{}
#mainnav li a{color:inherit;}
#mainnav .active a, #mainnav a:hover, #mainnav li:hover > a{color:#41C379;
background-color:inherit;}
#mainnav li li a, #mainnav .active li a{color:#FFFFFF;
background-color:rgba(65,195,121,.5); /* #41C379 */
border-color:rgba(0,0,0,.6);}
#mainnav li li:hover > a, #mainnav .active .active > a{color:#FFFFFF;
background-color:#41C379;}
#mainnav form select{color:#474747; background-color:#FFFFFF; }

#breadcrumb a{color:inherit; background-color:inherit;}
#breadcrumb li:last-child a{color:#41C379; }

.container .sidebar nav a{color:inherit; border-color:#D7D7D7;}
.container .sidebar nav a:hover{color:#41C379; }

.pagination a, .pagination strong{border-color:#D7D7D7;}
.pagination .current *{color:#FFFFFF; background-color:#41C379; }

#backtotop{color:#FFFFFF; background-color:#41C379; }

/* Tables + Comments */

```

```

table, th, td, #comments .avatar, #comments input, #comments
textarea{border-color:#D7D7D7;}
#comments input:focus, #comments textarea:focus, #comments
*:required:focus{border-color:#41C379;}
th{color:#FFFFFF; background-color:#373737;}
tr, #comments li, #comments input[type="submit"], #comments
input[type="reset"]{color:inherit; background-color:#FBFBFB;}
tr:nth-child(even), #comments li:nth-child(even){color:inherit;
background-color:#F7F7F7;}
table a, #comments a{background-color:inherit;}

```

/* Media Queries

```

----- */
@-ms-viewport{width:device-width;}

```

/* Max Wrapper Width - Laptop, Desktop etc.

```

----- */
@media screen and (min-width:1140px){
.hoc{max-width:1140px;}
}

```

```

@media screen and (min-width:978px) and (max-width:1140px){
.hoc{max-width:95%;}
}

```

/* Mobile Devices

```

----- */
@media screen and (max-width:978px){
.hoc{max-width:90%;}
}

```

#header{}

```

.navwrapper #mainnav{width:80%; padding:15px 0 0 0;}
.navwrapper #mainnav ul{display:none;}
.navwrapper #mainnav form{display:block;}

```

#breadcrumb{}

```

.container{}
.comments input[type="reset"]{margin-top:10px;}
.pagination li{display:inline-block; margin:0 5px 5px 0;}

```

#footer{}

#copyright{}

```

#copyright p:first-of-type{margin-bottom:10px;}
}

@media screen and (max-width:750px){
  .imgl, .imgr{display:inline-block; float:none; margin:0 0 10px 0;}
  .fl_left, .fl_right{display:block; float:none;}
  .group .group > *:last-child, .clear .clear > *:last-child, .clear .group >
*:last-child, .group .clear > *:last-child{margin-bottom:0;}/* Experimental -
Needs more testing in different situations, stops double margin when stacking
*/
  .one_half, .one_third, .two_third, .one_quarter, .two_quarter,
.three_quarter{display:block; float:none; width:auto; margin:0 0 30px 0;
padding:0;}

#header{}

#navwrapper{padding:0 20px;}
#navwrapper #searchform{display:none;}/* Needs modifying if the form is
required in small viewports */
#navwrapper #mainnav{float:none; width:100%; padding:15px 0; }

#pageintro article{max-width:none;}
#pageintro .heading{font-size:2rem;}
#pageintro p:first-of-type{font-size:1rem; }

#introblocks > ul{margin:0 auto; max-width:348px;}/* Centre the items and
restrict the width to the one_third grid element */
#introblocks > ul > li:nth-last-child(-n+3){margin-bottom:50px;}
#introblocks > ul > li:last-child{margin-bottom:0; }

.sectontitle{max-width:none; }

#footer{padding-bottom:50px;}/* Not required, just looks a little better */
}

@media screen and (min-width:450px) and (max-width:750px){
#stats > li{width:50%; margin-bottom:50px;}
#stats > li:nth-child(even){border:none;}
#stats > li:nth-last-child(-n+2){margin-bottom:0;}
}

@media screen and (max-width:450px){
#stats > li{float:none; width:100%; margin-bottom:25px; padding-bottom:25px;}
#stats > li, #stats > li:nth-child(even){border-width:0 0 1px 0;}
#stats > li:last-child{margin:25px 0 0 0; padding-bottom:0;}
}

/* Other
-----
----- */
@media screen and (max-width:650px){
.scrollable{display:block; width:100%; margin:0 0 30px 0; padding:0 0 15px 0;
}

```

```

overflow:auto; overflow-x:scroll;}
.scrollable table{margin:0; padding:0; white-space:nowrap; }

.inline li{display:block; margin-bottom:10px;}
.pushright li{margin-right:0;}

.font-x2{font-size:1.4rem;}
.font-x3{font-size:1.6rem;}
}

</style>
</head>
<body id="top">

<div class="bgded overlay"
style="background-image:url('/static/images/demo/backgrounds/synthetic-data-102
4x640.jpg');">

<header id="header" class="hoc clear">

<div id="logo" class="one_quarter first">
  <h1><a href="index.html">AMS</a></h1>
  <p>Face Recognizer</p>
</div>

</header>

<section id="navwrapper" class="hoc clear">
</section>

<div id="pageintro" class="hoc clear">

<article>
  <h3 class="heading">Facial-Fusion</h3><br>
  <h4>Facial recognition. Attendance revolution</h4><br>
  <p>Welcome to our cutting-edge attendance management system powered by
face recognition technology. <br>
    Our advanced facial recognition algorithms ensure accurate
identification, providing a reliable and secure method of attendance
verification. <br> Experience the future of attendance management,
where convenience meets precision.
  </p>
  <footer><a class="btn" href="/application" target="_blank">Attendance
Management Application</a></footer>
</article>

</div>

</div>

<div class="wrapper row2">

```

```

<section class="hoc container clear">

    <div class="sectiontitle">

        <h3 class="heading">FEATURES</h3>
    </div>
    <br><br>
    <ul class="nospace group center">
        <li class="one_third first">
            <article><a href="#"><i class="fas fa-camera fa-5x btm space-50"></i></a>
                <h6>Facial Recognition</h6>
                <p class="btm space-30"> Our attendance management system utilizes advanced facial recognition technology to accurately identify and verify individuals.</p>

            </article>
        </li>
        <li class="one_third">
            <article><a href="#"><i class="fas fa-clock fa-5x btm space-50"></i></a>
                <h6>Seamless Attendance Tracking</h6>
                <p class="btm space-30">With just a simple glance, user can effortlessly clock in. The system captures facial data in real-time, ensuring accurate attendance records.</p>

            </article>
        </li>
        <li class="one_third">
            <article><a href="#"><i class="fas fa-lock fa-5x btm space-50"></i></a>
                <h6>Reliable and Secure</h6>
                <p class="btm space-30"> Our system employs robust facial recognition algorithms that provide a high level of accuracy and security. </p>

            </article>
        </li>
    </ul>
    <br>
    <ul class="nospace group center">
        <li class="one_third first">
            <article><a href="#"><i class="fas fa-chart-line fa-5x btm space-50"></i></a>
                <h6>Real-time Monitoring</h6>
                <p class="btm space-30"> Stay informed with real-time attendance tracking and monitoring. Supervisors and administrators can easily view and manage attendance data.</p>

            </article>
        </li>
        <li class="one_third">
            <article><a href="#"><i class="fas fa-file-alt fa-5x btm space-50"></i></a>
                <h6>Attendance Report</h6>
                <p class="btm space-30">Generate straightforward and

```

easy-to-understand attendance reports. Our system simplifies the reporting process, providing attendance report on daily basis.</p>

```
</article>
</li>
<li class="one_third">
    <article><a href="#"><i class="fas fa-desktop fa-5x
btmspace-50"></i></a>
        <h6>User-Friendly Interface</h6>
        <p class="btmspace-30"> Our web-based interface is intuitive and
user-friendly, making it easy for users and administrators to navigate and
access attendance-related information. </p>

    </article>
</li>
</ul>

</section>
</div>

<div class="wrapper coloured">
    <section id="testimonials" class="hoc container clear">

        <div class="sectiontitle">
            <h6 class="heading">OUR TEAM</h6><br><br>
        </div>
        <article class="one_half first">
            <!-- <p>Proin sed sapien ut convallis lectus ac magna nulla mattis purus
phasellus consectetur fermentum augue integer vulputate lectus vitae lorem
suspendisse potenti donec suscipit.</p> -->
            <h6 class="heading">Priyank Saxena</h6>
            <em>B.Tech CSE 4th Year</em></article>
        <article class="one_half">
            <!-- <blockquote>Pellentesque et ipsum mattis ipsum pellentesque pretium
proin rutrum turpis non accumsan aliquet odio magna luctus neque in adipiscing
mi odio ac felis sed mattis enim quis ipsum.</blockquote> -->
            <h6 class="heading">Vivek Sharma</h6>
            <em>B.Tech CSE 4th Year</em></article>

    </section>
</div>

<div class="bgded overlay row4" style="background-color: black;">
    <footer id="footer" class="hoc clear">

        <div class="center btmspace-50">
            <h6 class="heading">AMS : Face Recognizer</h6>
            <ul class="faico clear">
                <li><a class="faicon-dribbble" href="#"><i class="fab
fa-dribbble"></i></a></li>
            </ul>
        </div>
    </footer>
</div>
```

```

        <li><a class="faicon-facebook" href="#"><i class="fab fa-facebook"></i></a></li>
        <li><a class="faicon-google-plus" href="#"><i class="fab fa-google-plus-g"></i></a></li>
        <li><a class="faicon-linkedin" href="#"><i class="fab fa-linkedin"></i></a></li>
        <li><a class="faicon-twitter" href="#"><i class="fab fa-twitter"></i></a></li>
        <li><a class="faicon-vk" href="#"><i class="fab fa-vk"></i></a></li>
    </ul>
    <p class="nospace">Revolutionize attendance with facial precision.</p>
</div>

<hr class="btmspace-50">
<p class="nospace" style="text-align: center;">We are dedicated to
simplifying attendance management with our state-of-the-art system powered by
face recognition technology. Our solution allows users to conveniently clock in
by simply looking into the camera, eliminating the hassle of traditional
methods. With our advanced facial recognition algorithms, accuracy and security
are guaranteed. We prioritize the protection of your data, ensuring that all
facial information is securely stored and encrypted. Experience seamless
attendance tracking, real-time monitoring, and customizable reporting options.
Join us in embracing the future of attendance management with our user-friendly
and efficient system.</p>
</footer>
</div>

<div class="wrapper_row5">
    <div id="copyright" class="hoc clear">
        <p style="text-align: center;">Copyright © 2023 - All
        Rights Reserved </p>
        <p class="fl_right">Project by Priyank Saxena & Vivek Sharma</p>
    </div>
</div>

<a id="backtotop" href="#top"><i class="fas fa-chevron-up"></i></a>

<script src="/static/layout/scripts/jquery.min.js"></script>
<script src="/static/layout/scripts/jquery.backtotop.js"></script>
<script src="/static/layout/scripts/jquery.mobilemenu.js"></script>
</body>
</html>

```

6.3. Html File : home.html

```
<!doctype html>
<html lang="en">

<style type='text/css'>
  * {
    padding: 0;
    margin: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  }

  body {
    background-image:
url('https://cutewallpaper.org/21/1920-x-1080-gif/1920x1080-Wallpapercartoon-Wallpapers-Driverlayer-Search-.gif');
    background-size: cover;
    font-family: sans-serif;
    margin-top: 40px;
    height: 100vh;
    padding: 0;
    margin: 0;
  }

  table {
    border: 1px;
    font-family: arial, sans-serif;
    border-collapse: collapse;
    width: 86%;
    margin: auto;
  }

  td,
  th {
    border: 1px solid black !important;
    padding: 5px;
  }

  tr:nth-child(even) {
    background-color: #dddddd;
  }
</style>

<head>
```

```

<!-- Required meta tags -->
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-eOJMYsd53ii+sc0/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgw
ra6" crossorigin="anonymous">

<title>Face Recognition Based Attendance System</title>

<script>
    function updateDateTime() {
        var now = new Date();
        var date = now.toLocaleDateString();
        var time = now.toLocaleTimeString();
        document.getElementById('date').textContent = date;
        document.getElementById('time').textContent = time;
    }

    // Update date and time every second
    setInterval(updateDateTime, 1000);
</script>
</head>

<body>

<div class='mt-3 text-center'>
    <h1 style="width: auto; margin: auto; color: white; padding:
11px; font-size: 44px;">Face Recognition Based
        Attendance System</h1>

        <p style="width: auto; margin: auto; color: white; padding:
11px; font-size: 20px;">Date: <span id="date"></span> Time: <span
id="time"></span></p>
</div>

{% if mess%}
<p class="text-center" style="color: red; font-size: 20px;">{{ mess }}</p>
{% endif %}

```

```

<div class="row text-center" style="padding: 20px; margin: 20px;">

    <div class="col"
        style="border-radius: 20px; padding:
0px; background-color:rgb(211,211,211,0.5); margin:0px 10px 10px 10px; min-height:
400px;">
        <h2 style="border-radius: 20px 20px 0px 0px; background-color:
#0c5c2e; color: white; padding: 10px;">Today's
            Attendance <i class="material-icons">assignment</i></h2>
        <a style="text-decoration: none; max-width: 300px;" href="/start">
            <button
                style="font-size: 24px; font-weight: bold; border-radius:
10px; width:490px; padding: 10px; margin-top: 30px; margin-bottom: 30px; "
                type='submit' class='btn btn-primary'>Take Attendance <i
                    class="material-icons">beenhere</i></button>
            </a>
        <table style="background-color: white;">
            <tr>
                <td><b>S No</b></td>
                <td><b>Name</b></td>
                <td><b>ID</b></td>
                <td><b>Time</b></td>
            </tr>
            {% if l %}
            {% for i in range(l) %}
            <tr>
                <td>{{ i+1 }}</td>
                <td>{{ names[i] }}</td>
                <td>{{ rolls[i] }}</td>
                <td>{{ times[i] }}</td>
            </tr>
            {% endfor %}
            {% endif %}
        </table>

        <br><p>
        <div class="btn btn-primary"> <a href="{{ url_for('download') }}"
style="color: white; text-decoration: none" download>Download Attendance
CSV</a></div>
        </p>

    </div>

    <div class="col"

```

```

        style="border-radius: 20px;padding:
0px;background-color:rgb(211,211,211,0.5);margin:0px 10px 10px 10px;height:
400px;">
        <form action='/add' method="POST" enctype="multipart/form-data">
            <h2 style="border-radius: 20px 20px 0px 0px;background-color:
#0c5c2e;color: white;padding: 10px;">Add
                New User <i
            class="material-icons">control_point_duplicate</i></h2>
            <label style="font-size: 20px;"><b>Enter New User
Name*</b></label>
            <br>
            <input type="text" id="newusername" name='newusername'
                style="font-size: 20px;margin-top:10px;margin-bottom:10px;">
required>
            <br>
            <label style="font-size: 20px;"><b>Enter New User
Id*</b></label>
            <br>
            <input type="number" id="newusereid" name='newuserid'
                style="font-size: 20px;margin-top:10px;margin-bottom:10px;">
required>
            <br>
            <button style="width: 232px;margin-top: 20px;font-size: 20px;">
type='submit' class='btn btn-success'>Add
                New User
            </button>
            <br>
            <h5 style="padding: 25px;"><i>Total Users in Database:
{{totalreg}}</i></h5>
        </form>
    </div>

    <div class="col"
        style="border-radius: 20px;padding:
0px;background-color:rgb(211,211,211,0.5);margin:0px 10px 10px 10px;min-height:
400px;">
        <form action="/delete" method="POST">
            <h2 style="border-radius: 20px 20px 0px 0px;background-color:
#0c5c2e;color: white;padding: 10px;">Delete User <i
            class="material-icons">delete</i></h2>
            <div class="form-group">
                <label style="font-size: 20px;">
for="username"><b>Username:</b></label>
                <br>
                <input type="text" style="font-size:
20px;margin-top:10px;margin-bottom:10px;" name="username" required>

```

```

        </div>
        <div class="form-group">
            <label style="font-size: 20px;" for="userid"><b>User
ID:</b></label>
            <br>
            <input type="text" style="font-size:
20px; margin-top:10px; margin-bottom:10px;" name="userid" required>
        </div>
        <button style="width: 232px; margin-top: 20px; font-size: 20px;" type='submit' class='btn btn-danger'>Delete User</button>
    </form>

</div>
<br>

<br>
<hr>
<br>
<br>
<!-- <form action="{{ url_for('view_database') }}">
    <input type="submit" value="View Database">
</form> -->

<div class="col"
      style="border-radius: 20px; padding:
0px; background-color:rgb(211,211,211,0.5); margin:0px 10px 10px 10px; min-height:
400px;">

    <h2 style="border-radius: 20px 20px 0px 0px; background-color:
#0c5c2e; color: white; padding: 10px;">
        User Database
        <i class="material-icons">storage</i>
    </h2>
    <br>
    <form action="{{ url_for('view_database') }}">
        <input type="submit" value="View Database" class="btn
btn-primary">
    </form>
    <br>
    <table style="background-color: white;">
        <tr>
            <td><b>ID</b></td>
            <td><b>Name</b></td>

```

```

        </tr>
        {% for user in users %}
        <tr>
            <td>{{ user.userid }}</td>
            <td>{{ user.username }}</td>

        </tr>
        {% endfor %}

    </table>

</div>

<br>
<br>

<!-- &lt;form action="{{ url_for('view_database') }}"&gt;
    &lt;input type="submit" value="View Database" class='btn btn-dark'&gt;
&lt;/form&gt;

&lt;h2&gt;User Database&lt;/h2&gt;
{% if users %}
    &lt;table style="color: #dddddd;"&gt;
        &lt;tr&gt;
            &lt;th&gt;Username&lt;/th&gt;
            &lt;th&gt;ID&lt;/th&gt;
        &lt;/tr&gt;
        {% for user in users %}
        &lt;tr&gt;
            &lt;td style="color: red;"&gt;{{ user.username }}&lt;/td&gt;
            &lt;td style="color: red"&gt;{{ user.userid }}&lt;/td&gt;
        &lt;/tr&gt;
        {% endfor %}
    &lt;/table&gt;

    {% endif %}
&lt;/div&gt; --&gt;

&lt;/body&gt;

&lt;/html&gt;
</pre>

```

7. SCREENSHOTS

7.1. Front web Page:

7.1.1. Introduction Section

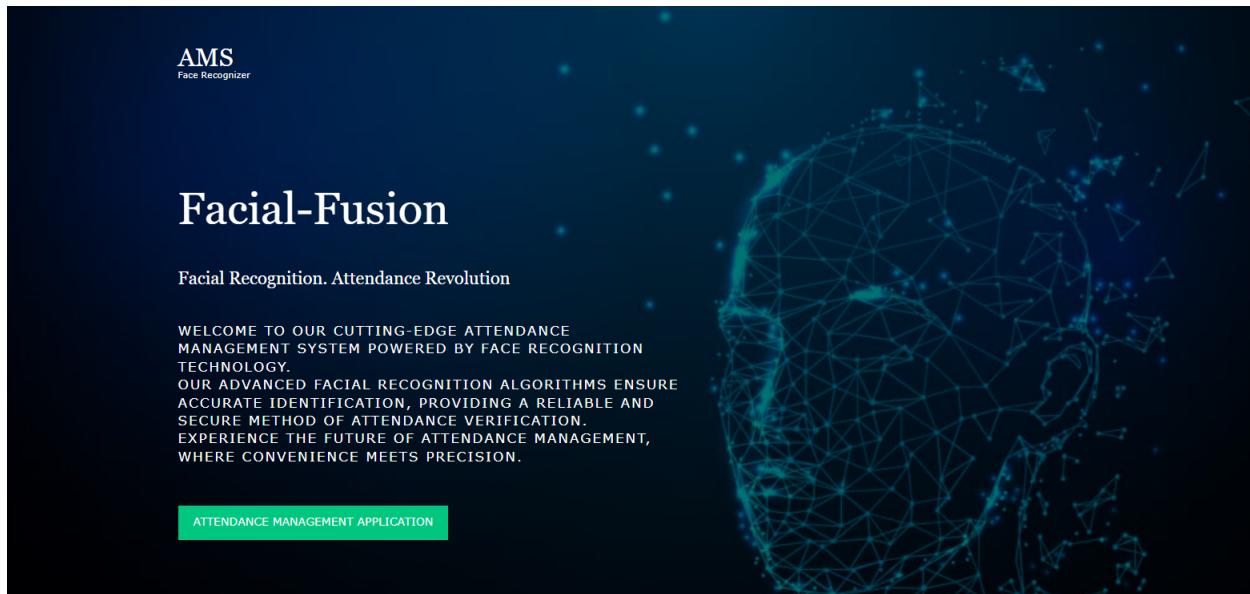


Fig No. 8 : Introduction Section

7.1.2. Features Section

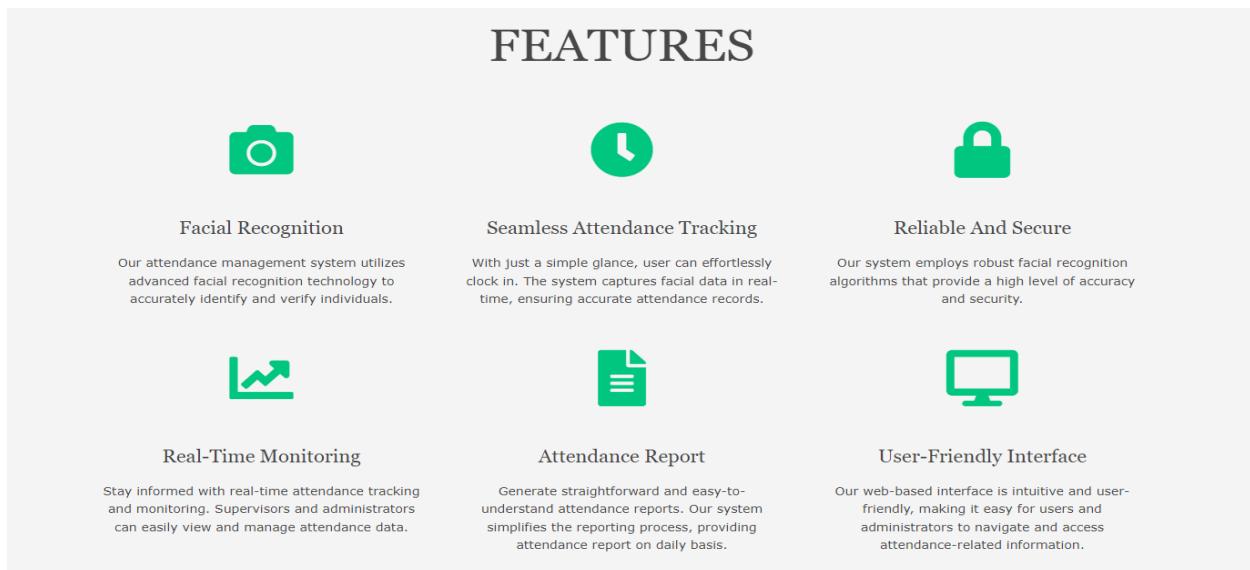


Fig No. 9 : Features Section

7.1.3. Our-Team Section

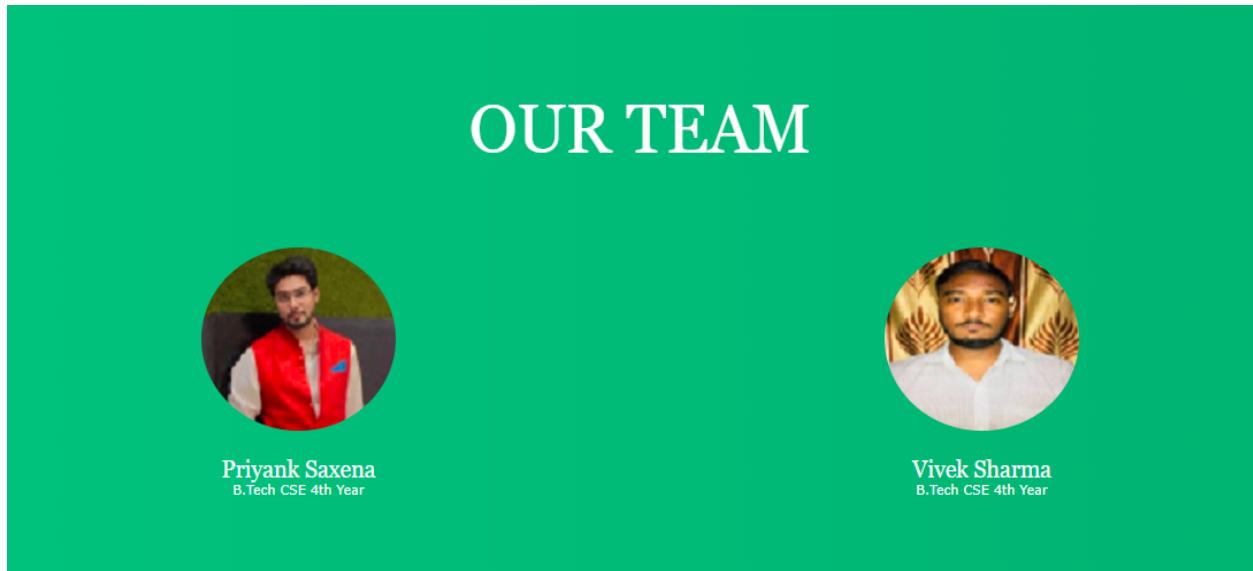


Fig No. 10 : Our-Team Section

7.1.4. Footer Section

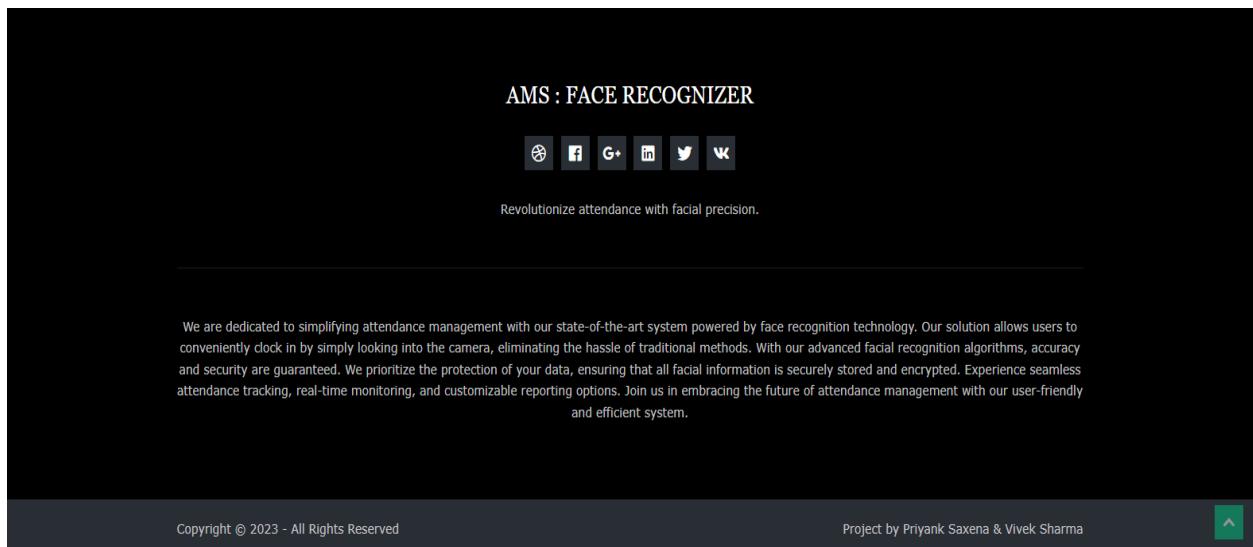


Fig No. 11 : Footer Section

7.2. Attendance Management System Web Page:

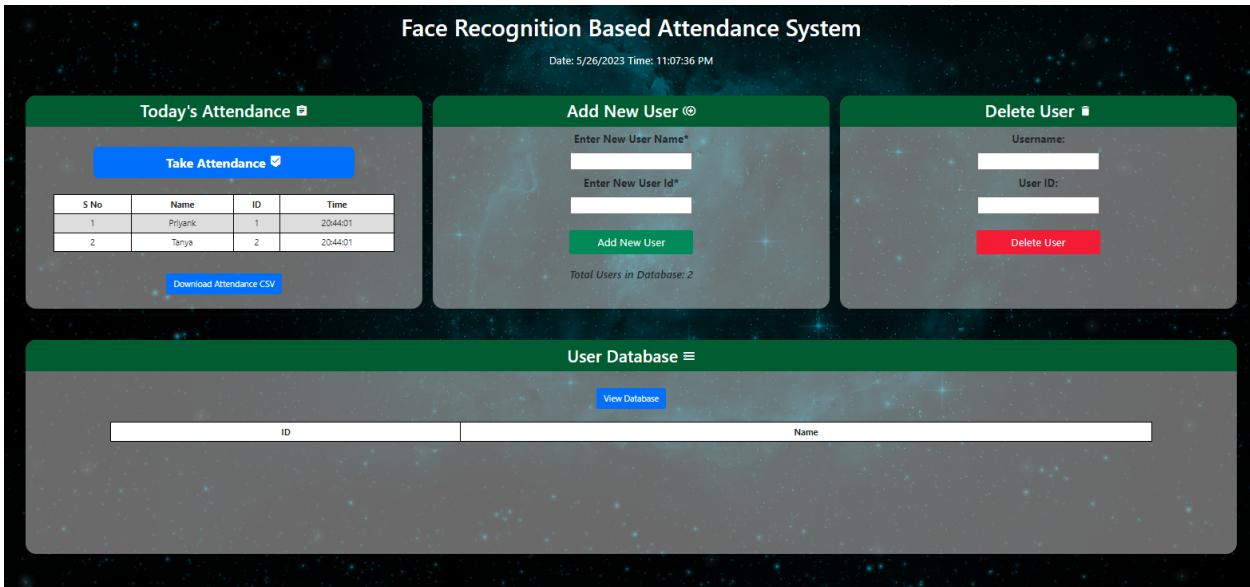


Fig No. 12 : Face Recognition Based Attendance System

7.2.1. Attendance Section:

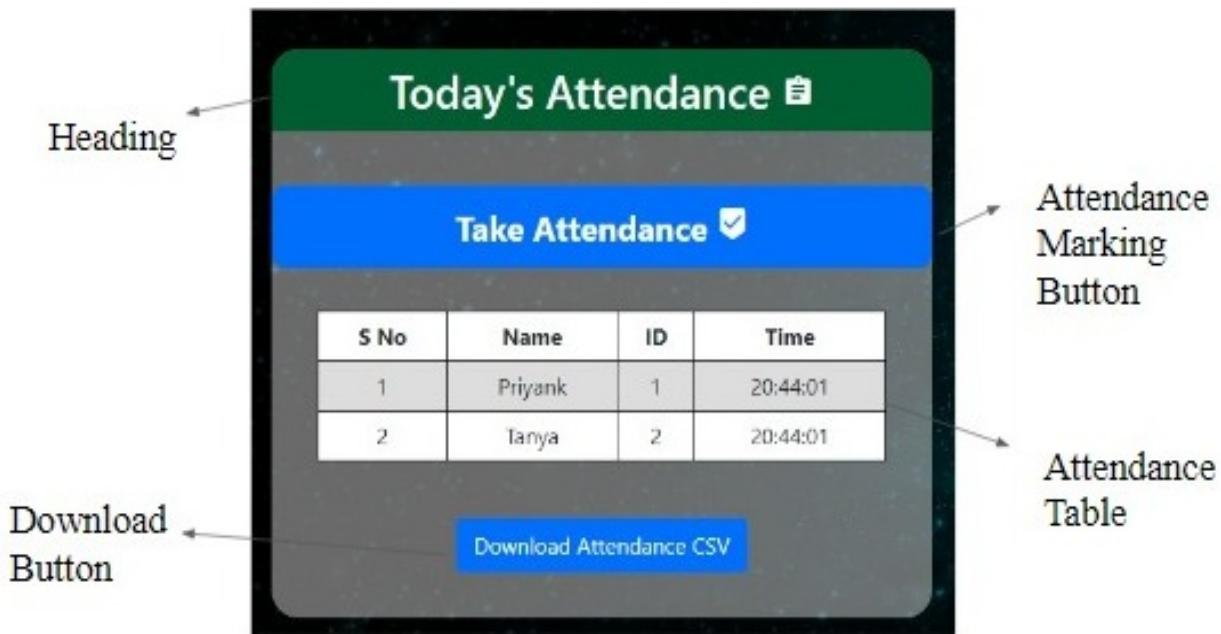


Fig No. 13 : Attendance Section

7.2.2. Add User Section:

The screenshot shows a user interface titled "Add New User" with a green header bar. Below the title is a sub-header "Enter New User Name*". There is a white input field for entering the user's name. Below it is another sub-header "Enter New User Id*" followed by a second white input field. At the bottom of the form is a green button labeled "Add New User". A small text message at the bottom right states "Total Users in Database: 2".

Input Form →

Add User Button →

Fig No. 14 : Add User Section

7.2.3. Delete User Section:

The screenshot shows a user interface titled "Delete User" with a green header bar. Below the title is a sub-header "Username:" followed by a white input field. Below it is another sub-header "User ID:" followed by a second white input field. At the bottom of the form is a red button labeled "Delete User".

Input Form →

Delete User →

Delete User Button →

Fig No. 15 : Delete User Section

7.2.4. View Database Section:

User Database	
ID	Name
1	Priyank
2	Vivek

Fig No. 16 :View Database Section

7.3.Working of Application:

7.3.1.Attendance Marking:

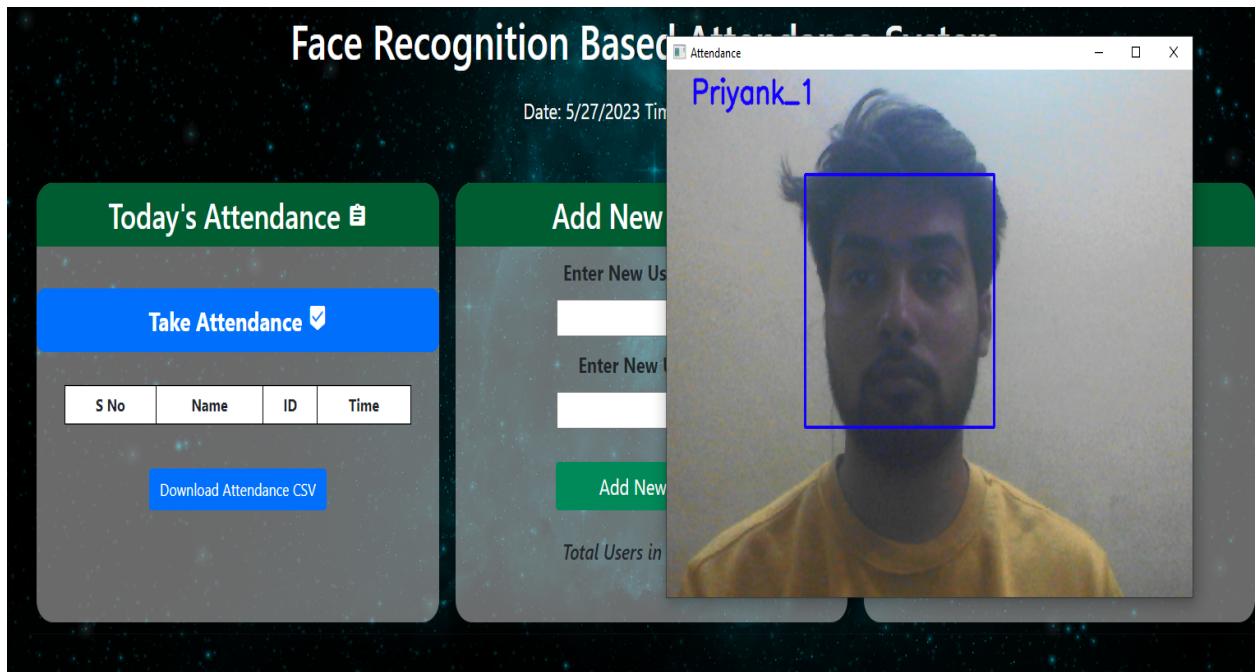


Fig No. 17 :Facial Recognition

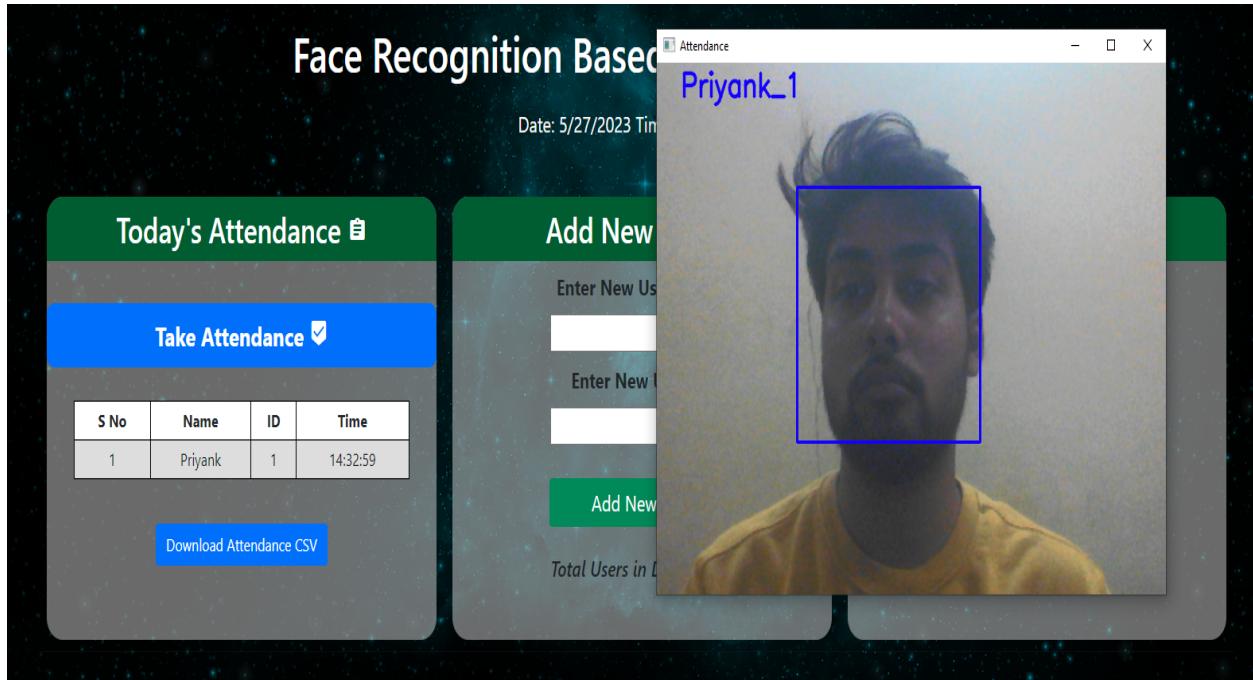


Fig No. 18 : Attendance Marked

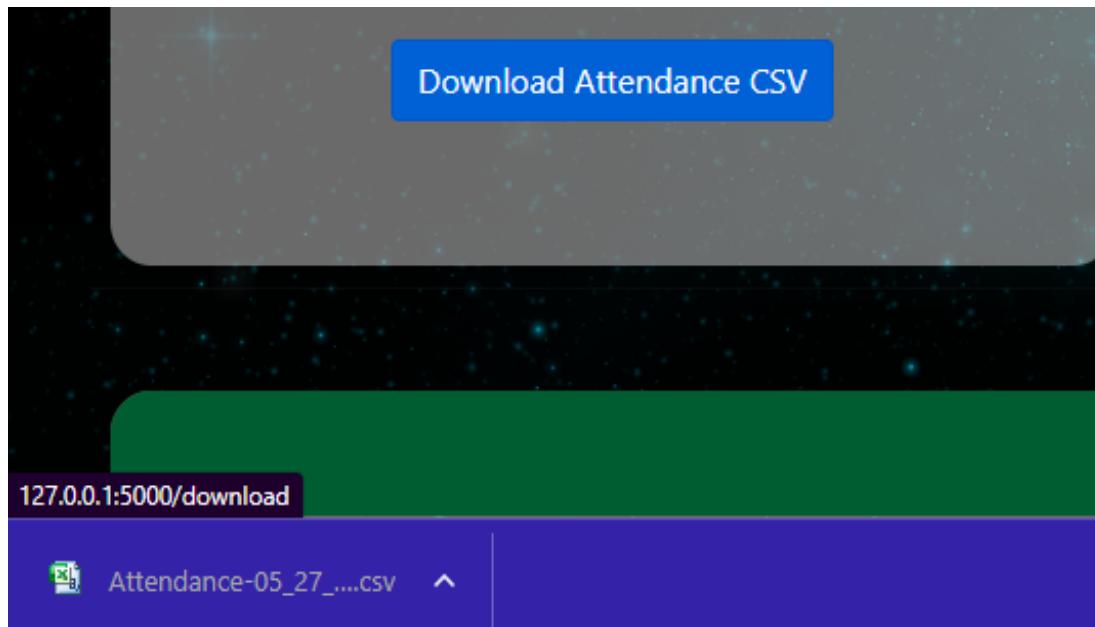


Fig No. 19 : Attendance Report (CSV File)

7.3.2.Adding New User:

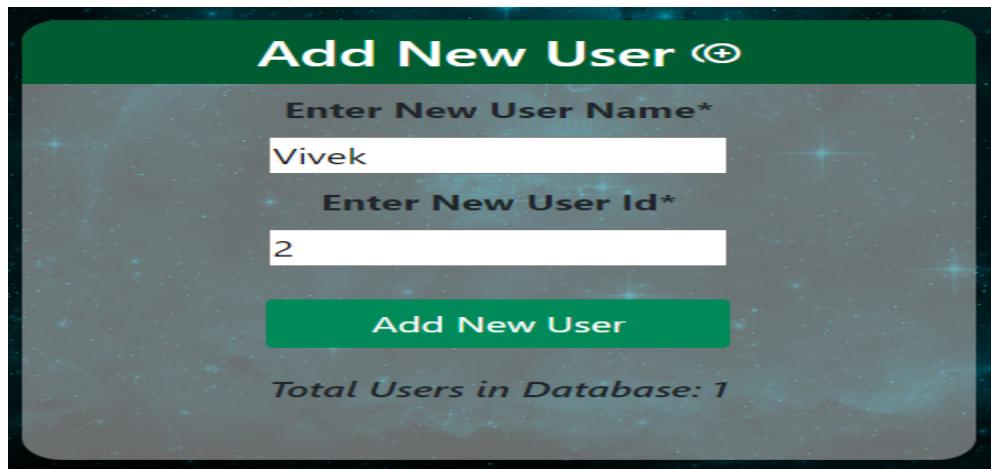


Fig No. 20 : Enter User Data



Fig No. 21 : Capturing User Face



Fig No. 22 : Total 50 Image Captured

```
127.0.0.1 - - [27/May/2023 14:33:10]
127.0.0.1 - - [27/May/2023 14:33:40]
127.0.0.1 - - [27/May/2023 14:37:52]
Training Model
127.0.0.1 - - [27/May/2023 14:42:50]
```

Fig No. 23 : Console Message: Training Model

7.3.3. Deleting User:

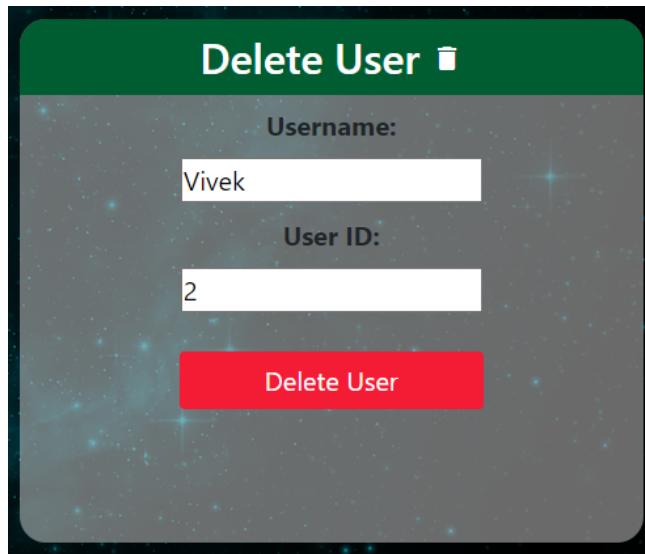


Fig No. 24 : Enter User Data

```
4:42:50] "POST /add HTTP/1.1" 200 -
4:57:32] "POST /delete HTTP/1.1" 302 -
4:57:32] "GET /application HTTP/1.1" 200
```

Fig No. 25 : Console Message: Deleted User

7.4.Face Folder:

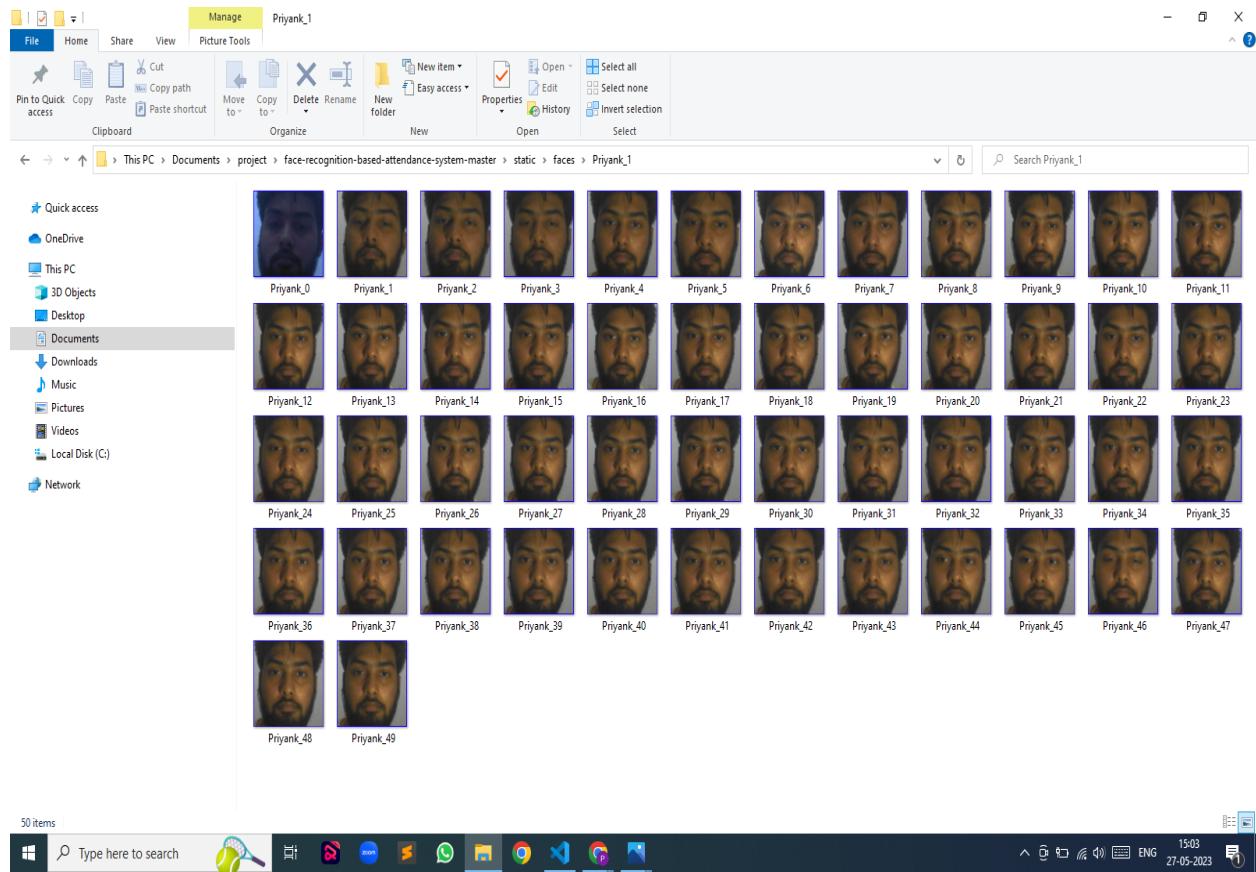


Fig No. 26 : Face Folder

8. FUTURE SCOPE

8.1. Admin Authentication:

Implementing admin authentication can enhance the security of the system. It would allow only authorized administrators to access the system and perform administrative tasks such as managing user accounts, generating reports, and configuring system settings. Admin authentication can be implemented using various methods such as username/password, two-factor authentication, or biometric authentication.

8.2. Pricing Transaction Gateway:

Integrating a pricing transaction gateway can enable the system to handle financial transactions related to services provided by the Attendance Management System. It would allow users or organizations to subscribe to different pricing plans, make payments, and manage their subscription details. This feature would be particularly useful for commercial deployments of the system.

8.3. Application Performance Improvements:

Improving the performance of the application can lead to a better user experience and increased efficiency. Some possible performance enhancement areas to consider are:

- **Optimize Face Recognition Algorithms:** Explore and implement advanced face recognition algorithms or techniques that improve accuracy and speed. This could include leveraging deep learning models, feature extraction methods, or hardware acceleration for faster processing.
- **Database Connectivity:** Connect the system to a flexible and scalable database to eliminate the reliance on local disk databases. This allows for better data management, improved data accessibility, and increased scalability. Consider

using cloud-based databases or distributed database systems to handle large volumes of data and concurrent user access.

- **Caching and Data Retrieval Optimization:** Implement caching mechanisms to store frequently accessed data in memory, reducing the need for repeated database queries. This can significantly improve application performance and response times. Utilize indexing, query optimization techniques, and data compression to enhance data retrieval efficiency.
- **Load Balancing and Scalability:** Implement load balancing techniques to distribute the workload across multiple servers or instances. This ensures high availability, improved performance, and the ability to handle increased user loads. Consider using horizontal scaling by adding more servers or utilizing containerization technologies for efficient resource allocation.
- **User Interface Optimization:** Enhance the user interface and user experience to make the application more intuitive and responsive. Optimize the client-side code, minimize network requests, and leverage modern web development frameworks to ensure a smooth and efficient user interaction.

8.4. Secure and Scalable Database Connectivity:

Connect the Attendance Management System to a dedicated and secure database system that supports scalability and high availability. Utilize industry-standard encryption protocols, access control mechanisms, and data backup strategies to ensure data integrity, confidentiality, and disaster recovery.

8.5. Machine Learning Model Updates:

Continuously update and improve the face recognition algorithms and models used by the system. Regularly retrain the models with new data to enhance accuracy and adaptability to changing scenarios.

By incorporating these future scope enhancements, the Attendance Management System using Face Recognition will become more robust, secure, scalable, and efficient. It will provide enhanced functionality, seamless integration, improved performance, and reliable data management capabilities.

In conclusion, the future scope of the Attendance Management System using Face Recognition includes implementing admin authentication, integrating a pricing transaction gateway, improving application performance, connecting to a flexible and scalable database, and enabling integration with external systems. These enhancements will enhance security, financial management, performance, scalability, and data management capabilities of the system, leading to an improved user experience and increased efficiency.

9. CONCLUSION

The Attendance Management System using Face Recognition has been successfully developed and implemented as a reliable and efficient solution for managing attendance in various settings. Through the use of advanced face recognition algorithms and technologies, the system offers accurate and automated attendance tracking, eliminating the need for manual processes and reducing human error.

Throughout the project, we have achieved the objectives set forth at the beginning, which included designing and implementing a user-friendly interface, integrating face recognition capabilities, and providing robust attendance management functionalities. The system has demonstrated its effectiveness in accurately identifying individuals and recording their attendance in real-time.

The project has also addressed several important considerations, such as data security and privacy. By implementing secure authentication mechanisms and adhering to industry best practices, the system ensures that only authorized individuals have access to sensitive data. Additionally, privacy concerns have been taken into account by incorporating data protection measures and compliance with relevant regulations.

Furthermore, the future scope of the project presents opportunities for further enhancements. Admin authentication can be implemented to enhance security and control, allowing authorized administrators to manage the system effectively. Integrating a pricing transaction gateway would enable commercial deployments of the system and facilitate financial transactions related to its services.

Improvements in application performance, including optimizing face recognition algorithms, enhancing database connectivity, and implementing caching mechanisms, can contribute to faster response times and a better user experience. Connecting the system to a flexible and scalable database eliminates the limitations of local disk storage and provides efficient data management capabilities.

Additionally, the system's integration with external systems such as student information systems or HR systems can streamline data synchronization and enhance overall efficiency. Regular updates and improvements to the face recognition models will ensure adaptability to changing scenarios and maintain high accuracy.

In conclusion, the Attendance Management System using Face Recognition offers a comprehensive solution for attendance management in various domains. It combines the power of face recognition technology with robust functionalities and security measures. With the potential for future enhancements, the system can continue to evolve and meet the ever-changing needs of attendance management in the modern era.

10. REFERENCES

URL References:

- https://en.wikipedia.org/wiki/Artificial_intelligence
- https://www.hpe.com/in/en/what-is/artificial-intelligence.html?jumpid=ps_cz_xkg6qts3_aid-520061736&ef_id=Cj0KCQjw4omaBhDqARIsADXULuVrHQ34iwers7NHPrRWUI8wUtocWr99eVdOOJOVO_ArlMcv4MZb8FQaAgSPEALw_wcB:G:s&s_kwcid=AL!13472!3!595029415342!e!!g!!learn%20about%20ai!17064341482!133355606822&
- http://ethesis.nitrkl.ac.in/2610/1/15th_may.pdf
- <https://www.nec.co.nz/market-leadership/publications-media/advantages-and-disadvantages-of-fingerprint-recognition>
- <https://www.ijarcce.com/upload/2015/march-15/IJARCCE%20149.pdf>
- <https://www.phddirection.com/fingerprint-recognition-opencv-python>
- https://www.academia.edu/12184517/Biometric_Student_Record_Management_System
- <https://www.spectra-vision.com/resources/the-future-of-biometrics-in-security-and-access-control>