1. INTRODUCTION

1.1 GENERAL INTRODUCTION:

Human face plays an important role in our day-to-day life mostly for identification of a person. Face recognition is a part of biometric identification that extracts the facial features of a face, and then stores it as a unique face print to uniquely recognize a person. Biometric face recognition technology has gained the attention of many researchers because of its wide application. Face recognition technology is better than other biometric based recognition techniques like finger-print, palm-print, iris because of its non-contact process. The face recognition techniques are currently implemented in social media websites like Facebook, at the airports, railway stations, at crime investigations. Face recognition technique can also be used in crime reports, the captured photo can be stored in a database, and can be used to identify a person. Facebook uses the facial recognition technique for automating the process of tagging people. For face recognition we require large dataset and complex features to identify a person in all conditions like change of illumination, age, pose, etc. Recent researches show there is a betterment in facial recognition systems. In the last ten years there is huge development in recognition techniques.

But currently most of the facial recognition techniques is able to work fine only if the number of people in one frame is very few and under controlled illumination, proper position of faces and clear images. During the recent few years, a good improvement has been made in facial recognition systems. In comparison to the last decade, one can observe an enormous development in the world of face recognition. Currently, most of the facial recognition systems perform well with limited faces in the frame. Moreover, these methodologies have been tested under controlled lighting conditions, proper face poses and non- blurry images. The system that is proposed for face recognition in this paper for attendance system is able to recognize multiple faces in a frame without any control on illumination, position of face.

1.2 EXISTING SYSTEM

The term multi-view face recognition, in a strict sense, only refers to situations where multiple cameras acquire the subject (or scene) simultaneously and an algorithm collaboratively utilizes the acquired images/videos. But the term has frequently been used to recognize faces across pose variations. This ambiguity does not cause any problem for recognition with (still) images; a group of images simultaneously taken with multiple cameras and those taken with a single camera but at different view angles are equivalent as far as pose variations are concerned. However, in the case of video data, the two cases diverge. While a multi-camera system guarantees the acquisition of multi-view data at any moment, the chance of obtaining the equivalent data by using a single camera is unpredictable. Nonetheless, most existing multi-view video face recognition algorithms exploit single-view videos. Given a pair of face images to verify, they look up in the collection to "align" the face part's appearance in one image to the same pose and illumination of the other image. This method will also require the poses and illumination conditions to be estimated for both face images. This "generic reference set" idea has also been used to develop the holistic matching algorithm, where the ranking of look-up results forms the basis of matching measure. Student attendance system is needed to measure student participation in a classroom. In many institutions and organization, the attendance is important. The previous approach in which manually taking attendance and maintain its records was very inconvenient task. Another Approach is Biometric device, using biometric device like fingerprint scanner. In the existing system, they used **The Eigen Face Method** for facial recognition. Since we have seen that the **Advanced Haar-Cascade Algorithm with Deep learning and CNN** methods performs better than the Eigenface method in certain conditions.

1.2.1 DRAWBACKS OF THE EXISITNG SYSTEM

- Failed in simultaneous access in face recognition
- Performance is less at the time of face recognition
- Need large number of datasets
- There is no interface for predict unauthorized access in real time

-

1.3 PROPOSED SYSTEM

Nowadays Attendance is considered as an important factor for both the student as well as the teacher of an educational organization. With the advancement of the machine learning technology the machine automatically detects the attendance performance of the students and maintains a record of those collected data. In the video scenes, human faces can have unlimited orientations and positions, so its detection is of a variety of challenges to researchers. The proposed feature is developed using the spherical harmonic representation of the face, texture mapped onto a sphere. The texture map for the whole face is constructed by back-projecting the image intensity values from each o-f the views onto the surface of the spherical model. A promising approach to handle pose variations and its inherent challenges is the use of multi-view data. In video based face recognition, great success has been made by representing videos as linear subspaces, which typically lie in a special type of non-Euclidean space known as HAAR cascade algorithm manifold. From the perspective of manifold learning, our method can be regarded as performing a geometry-aware dimensionality reduction from the original HAAR cascade algorithm manifold to a lower-dimensional, more discriminative manifold where more favorable classification can be achieved. And also provide neural network algorithm to classify faces with improved accuracy in attendance system. Finally provide SMS and Email based alert system with real time implementation. The task of the proposed system is to capture face of each student and to store it in the database for their attendance. Concepts which are present in our proposed system are:

- Image Capturing
- Single and Multiple Face Detection
- Single and Multiple Face Recognition
- Attendance Management
- Data Base Storing System

DATASETS AND ALGORITHMS:

- The Datasets used in this system are **the real time facial datasets** to recognize the real time faces.
- Feature extraction approach is used to extract facial points such as face skin tone, nose, lips and so on
- Using **Advanced Haar-Cascade Algorithm with Deep learning and CNN methods** to construct subspaces in between to find connections between query and reference images
- Provide the alert at the time of unauthorized access.
- In addition to this, **Flickr-Faces-HQ Dataset (FFHQ)** is used in terms of age, ethnicity and image background, and also has much better coverage of accessories such as eyeglasses, sunglasses, hats, etc.

1.3.1 ADVANTAGES OF PROPOSED SYSTEM

- Overcome heterogeneous face matching problem
- Complexity is low and performance is high
- Time consuming process
- Real time alert system

2. LITERATURE REVIEW

2.1 TITLE: AUTOMATED ATTENDANCE MANAGEMENT USING ONESHOT LEARNING

AUTHOR: ARVIND VIGNESH K

Attendance in colleges and universities has become a very important and integral routine. In every University almost all the class hour or session starts with the process of taking attendance. As explained earlier, the most common way of taking attendance is the manual process where the staff manually cross checks the student from using the register. This process is time consuming and is inefficient. This also takes a lot of time if there is a lot of students to be addressed. Further there is also the case of proxy which can happen when the teacher is dealing with a lot of students. Thus, traditional process are time consuming, ineffective and unreliable as the time increases. While modern systems like bio-metrics have combated the unreliability factor, they still need hardware and are time consuming as the students need to take time in entering the biometric. This can be rectified by using facial recognition systems. The significant advantage that these systems have over bio-metrics is the fact that they are economically feasible and are not as intrusive as the former. This system proposes a facial recognition system that uses viola jones. It also explains that the system needs a very high tier webcam for better accuracy. This system is demanding a lot of hardware for implementation and it can be used only for laboratories and exam halls as not all class sessions will happen with the students sitting in-front of a computer.

- **Merits**:
    - Proxy is prevented
- **Demerits**:
    - Photo based attack can be occurred

2. TITLE: FACE RECOGNITION BASED ATTENDANCE SYSTEM USING HAAR CASCADE AND LOCAL BINARY PATTERN HISTOGRAM ALGORITHM

AUTHOR: BHARATH TEJ CHINIMILLI

Control of machines and their process with various technologies based on computer software is called automation. In this modern age, these advancements have proven to increase accuracy and also helping us to improve our livelihood. Innovations such as these save lots of labor work. One advancement in the field of automation is the Automated Attendance system replacing the old and traditional attendance marking. The paper-based method of marking attendance is time-consuming and its complexity increases with the increase of overall strength. This case is nullified in this automated version has it saves time and an additional bonus comes with security as it also helps to prevent proxy of attendance. The objective of our proposed system is to create a face recognition-based attendance system with getting a less false positive rate in detecting unknown persons by applying a threshold and save their images. We used Haar cascade for face detection because of their robustness and LBPH algorithm for face recognition. It is robust against monotonic gray scale transformations. Our System even detects and saves the images of any unknown person in the class whose Information is not there in the database. This system provides functionalities such as taking images of students along with their details for the database, training the images in the database and on the camera and start tracking people entering the class. When students enter the classroom this system detects the faces of students who are entering the classroom from the camera and pre-processed for further processing**.**

- **Merits**:
    - It recognizes students even wearing glasses or grown a beard.
- **Demerits**:
    - The dataset is small.

3. TITLE: A MULTIPLE FACE RECOGNITION SYSTEM WITH DLIB'S RESNET NETWORK USING DEEP METRIC LEARNING

AUTHOR: DR PRAVEEN KUMAR S

The model proposed in this paper is very effective as it gives quick and accurate results. Since our model stores only the encodings of the images rather than the images themselves, thereby reducing the space and time needed to retrieve images from the database and process them each time an image needs to be recognized. The system is also capable of batch processing multiple images at a time. Thus, the aim of the paper to demonstrate multiple face detection and recognition is successfully achieved by using Dlib's ResNet Network. Despite many approaches in facial recognition models, we often come across single facial recognition systems from an image. But the detection and recognition of a single face from an image is not very practical in an ever-changing world. For this, we need systems capable of detecting and recognizing multiple faces from a single image with which we can solve many real-world problems with a fewer number of images.So, we proposed an enhanced and efficient model, called HPMR (High Performance Multiple-face Recognition), to detect and recognize multiple faces from a single image. In this paper, we used Dlib's ResNet network with 29 convolution layers to recognize faces. The network supports both Predictor 5 and Predictor 68 model to estimate facial landmarks. This model is trained on a data set of three million faces from well-known data sets like face scrub and VGG using deep metric learning. It uses HOG feature descriptor for multiple face detection. We used this model because it gives us better results compared to other existing models. Among many possible applications for multiple facial recognition, we have implemented an automated attendance system to demonstrate our approach.

- **Merits**:
    - Capable of detecting multiple faces at a time.
- **Demerits**:
    - Complexity is high

4. TITLE: SMART ATTENDANCE NOTIFICATION SYSTEM USING SMTP WITH FACE RECOGNITION

AUTHOR: B. PRUTHVIRAJ GOUD, S. SRAVAN REDDY, K. PRAVEEN KUMAR

This paper exactly focused on producing a secure attendance marking system which is based upon one of the human gestures as Face. There are two stages to implement the approach. One is face detection using Haar classifier. Second one is face recognition using LHBP classifiers which is generated from trained faces. The proposed system is going to record the attendance of the people who are present in a classroom environment autonomously and this is an easiest way to produce the analysis and proof oriented approach makes as reliable applications. Every day we used to call the student and track their attendance hour by hour. It will produce more complexities to faculty member while tracking attendance. It is very if it is for small group of institution and it is different case in large groups. A manual entry leads to produce the time-consuming approach, cost effective, human mistakes, faking is easy and less accuracy. We are proposing a new and smart attendance system which is ready to enter the Daily attendance and tracking and report generations for checking the performance of students. Proposed automated system is working based on one of the human biometric. Face recognition is the one of the latest trend in many aspects and domains like Mobiles, Security. Based on the human face attendance will be marked in day to day life. Using camera captured image will be recognized and stored in database along with date and time. So we can evaluate, generate the report which is in proof oriented approach which gives some relaxation to professional bodies. It provides the advantages as easy tracking, no manual entries, no time wastage, and performance calculation is bit easy.

- **Merits**:
  - It sends mails to student for confirmation.
- **Demerits**:
  - Only support face image datasets.

5. TITLE: AUTOMATED ATTENDANCE SYSTEM USING MULTIPLE FACE RECOGNITION BY CD-LBP ALGORITHM

AUTHORS: Dr. AVINASH KUMAR SINGH

In this paper, Biometric attendance system emerged with marking the attendance through the fingerprints of the students. Although it is better than the manual method, it fails in the fact that it makes the students stand in a long queue which is again time consuming. Face recognition is also available in the same biometric module, it is used for analyzing individuals only which still needs huge time. The proposed system gives an efficient way to automatically recognize the faces of the students and maintain attendance database based on their presence in a real time background. The results have shown improved behaviour over manual attendance system with a higher accuracy. It also tends to highly increase the security of the attendance management system. In attendance system based on fingerprint verification has been developed which can be used to place the student's finger on the sensor during that time without the instructor's intervention. This system provides a fool proof method for marking the attendance RFID based system is introduced where the students take a RFID tag and they have to place that on the RFID card reader to record their attendance. Still this system gives a fraudulent access because an unauthorized person makes use of RFID card and enters into the institution. In Iris recognition system based Daugman's algorithm is developed that does capturing the image of iris. It will recognize and extract the features of it. In the authors proposed a method which is consist of many stages such as facial skin detection, facial features localization, representative features extraction and facial matching. A face identification technique has been presented that is equipped for preparing the picture essentially quick and accomplishes high recognition rates. In a quick and reliable automatic human Support Vector Machines (SVM) and the face identification systems is developed for the need of localizing and feature extraction from it.

- **Merits:**
    - Eliminates the time required for the manual attendance system.
- **Demerits:**
    - Sometimes provide high number of false positive rate

3. REQUIREMENT AND ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is an analysis used in measuring the ability and likelihood to complete a project successfully including all relevant factors. It must account for factors that affect it such as economic, technological, legal and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it.

3.1.1 Operational Feasibility study-- Define the urgency of the problem and the acceptability of any solution; If the system is developed, will it be used? Includes people oriented and social issues: internal issues, such as manpower problems, labour objections, manager resistance, organizational conflicts and policies; also external issues, including legal aspects and government regulations, also social acceptability of the new system.

3.1.2 Technical Feasibility study -- Is the project feasibility within the limits of current technology? Does the technology exist at all? Is it available within given resource constraints (i.e., budget, schedule,)?

3.1.3 Economic Feasibility study (Cost/Benefits Analysis Cost/Benefits Analysis) -- Is the project possible, given resource constraints? Are the benefits that will accrue from the new system worth the costs? What is the savings that will result from the system, including tangible and intangible ones? What are the development and operational costs?

3.1.4 Schedule feasibility study -- Constraints on the project schedule and whether they could be reasonably met

3.2 HARDWARE REQUIREMENTS

- Processor : Intel Core processor
- RAM : 4 GB
- Hard disk : 500 MB to 1 TB
- Compact Disk : 650 Mb>
- Keyboard : Standard keyboard
- Monitor : 15-inch color monitor
- Camera : System Camera (Single Shot)

For Multiple Face Detection High Quality Camera is Required (Kinect Camera)

3.3 SOFTWARE REQUIREMENTS

- Operating system : Windows OS
- Front End : PYTHON LANGUAGE
- Back End : WAMP Server-phpMyAdmin Server
- IDE : PyCharm Community/ Python IDE
- Application : Windows application

3.4 FUNCTIONAL REQUIREMENTS

A Functional Requirement (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called Functional Specification. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behavior of the system.

3.5 NON-FUNCTIONAL REQUIREMENTS

A Non-functional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. Also known as system qualities, nonfunctional requirements are just as critical as functional Epics, Capabilities, Features, and Stories. They ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do not fulfill mandatory requirements imposed by regulatory or standards agencies. In some cases, non-compliance can cause significant legal issues (privacy, security, safety, to name a few).

3.6 STATIC ANALYSIS

Static analysis, also called static code analysis, is the process of analyzing a computer program to find problems in it without actually executing it. Most generally, static analysis is performed on the source code of the program with tools that convert the program into an abstract syntax tree (AST) to understand the code's structure and then find problems in it.

Static Analysis is the testing and evaluation of an application by examining the code without executing the application. This technique of testing helps in evaluating both web and non-web applications and through advanced modeling, can reveal errors that may not manifest themselves until weeks, months or years after release

The primary advantage of static analysis is that it examines all possible execution paths and variable values, not just those invoked during execution. This aspect of static analysis is especially valuable in security assurance, because security attacks often exercise an application in unforeseen and untested way.

Static analysis is a powerful tool to ensure software quality and robustness, and can find a number of issues in code before execution. Some of these categories of issues are:

1.Potential security vulnerabilities

2.Bug risks and anti-patterns

3.Violation of code style guidelines

4.Performance issues

5.Dead or unused code

4. DESIGN

4.1 SYSTEM ARCHITECTURE

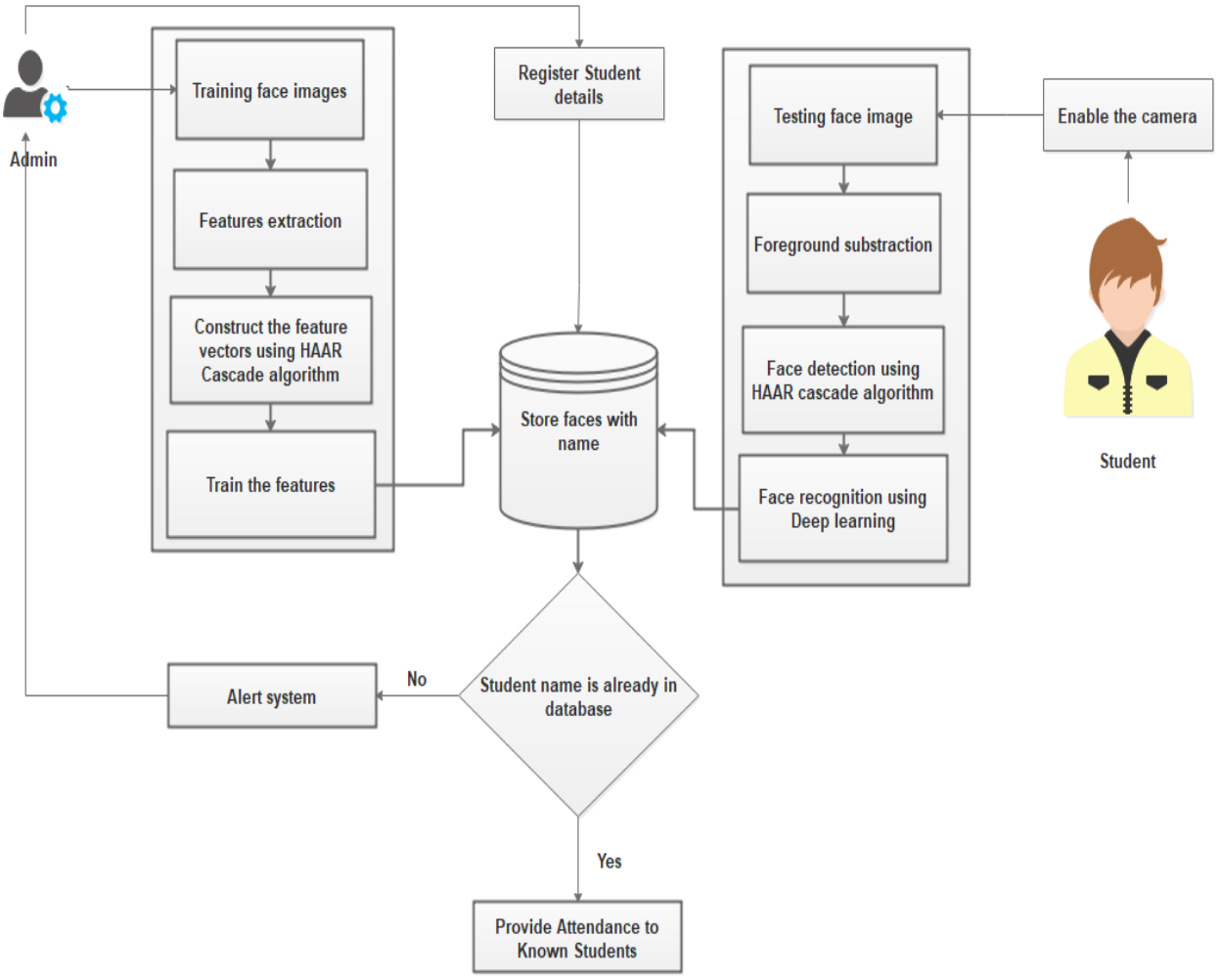


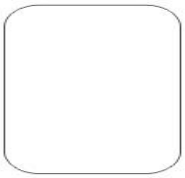
Fig 4.1. System Architecture

-

4.2 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

Data flow Symbols:

Symbol	Description
	**An Entity-** A source of data or a destination for data.
	**Process** or task that is performed by the system.
	A **Data Store**, a place where data is held between processes.
	**Data flow**

LEVEL 0

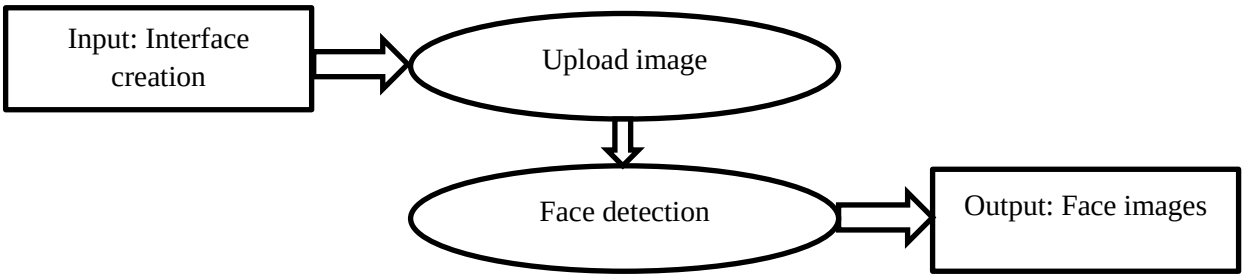


Fig 4.2.1. Level 0

LEVEL 1

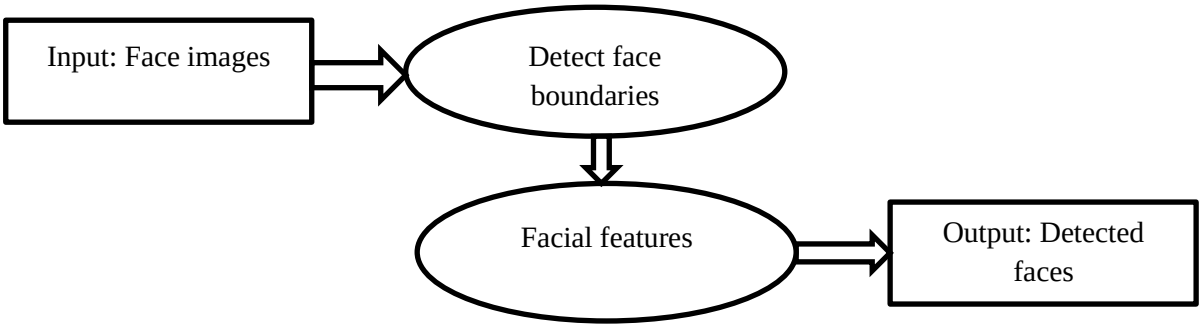


Fig 4.2.2. Level 1

LEVEL 2

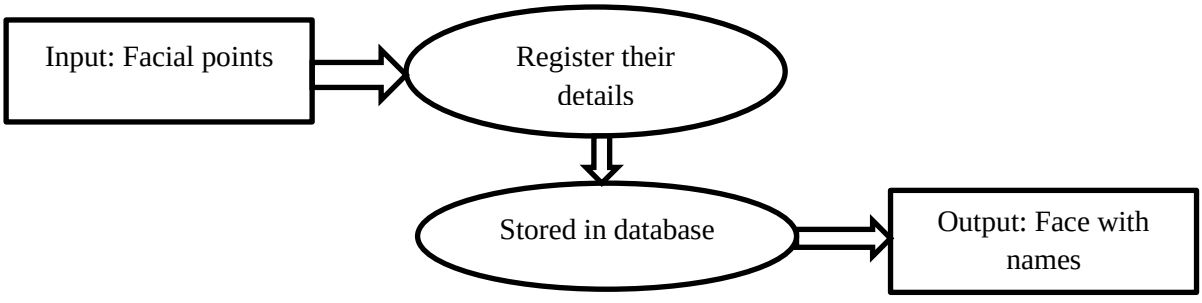


Fig 4.2.3. Level 2

LEVEL 3

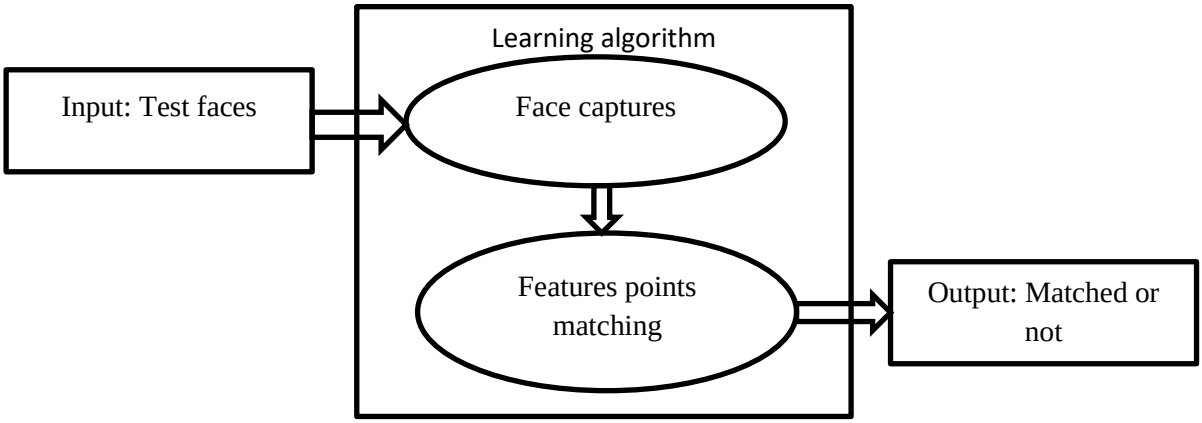


Fig 4.2.4 Level 3

LEVEL 4

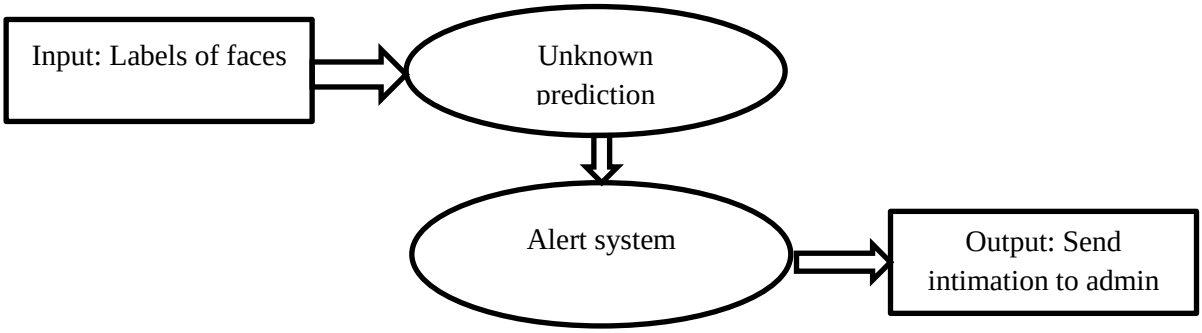


Fig 4.2.5. Level 4

4.3 UML DIAGRAMS

4.3.1 USE CASE DIAGRAM

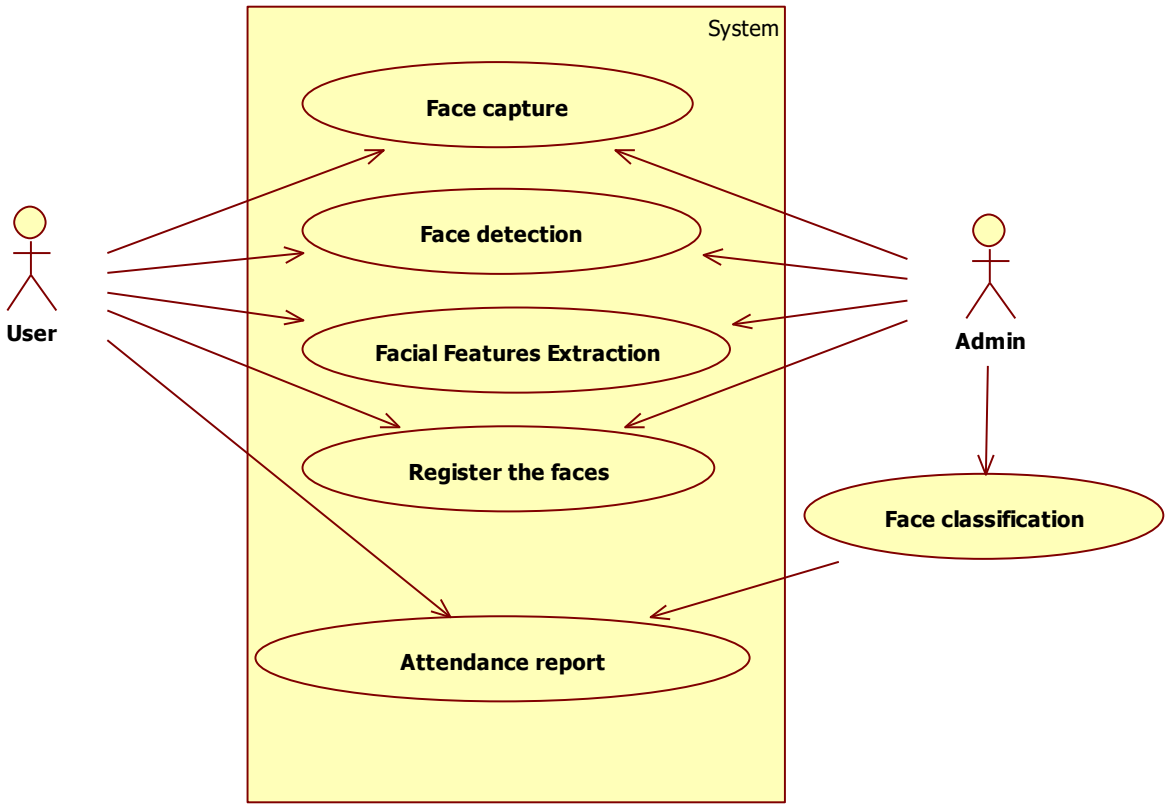


Fig 4.3.1. Use Case Diagram

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

4.3.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

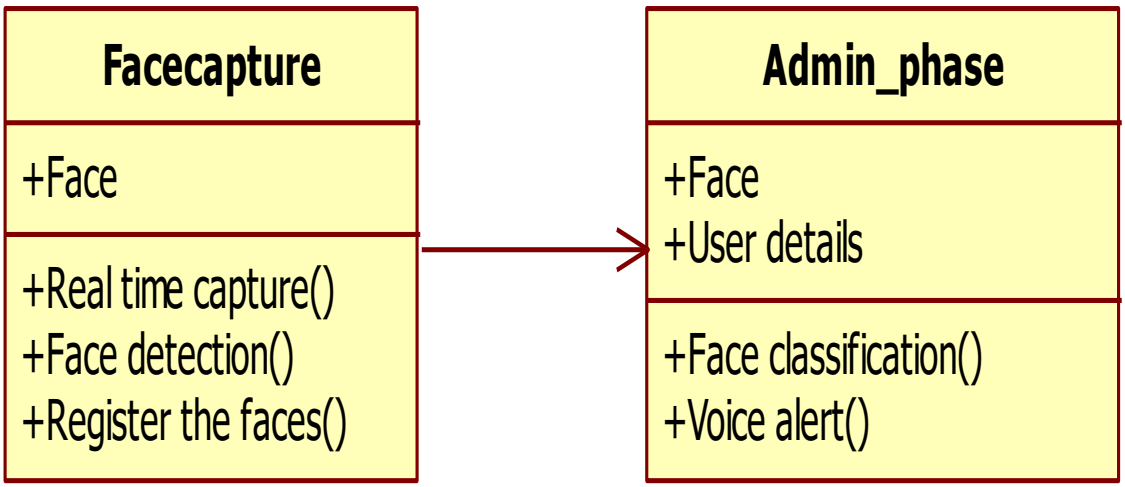


Fig 4.3.2. Class Diagram

4.3.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram.

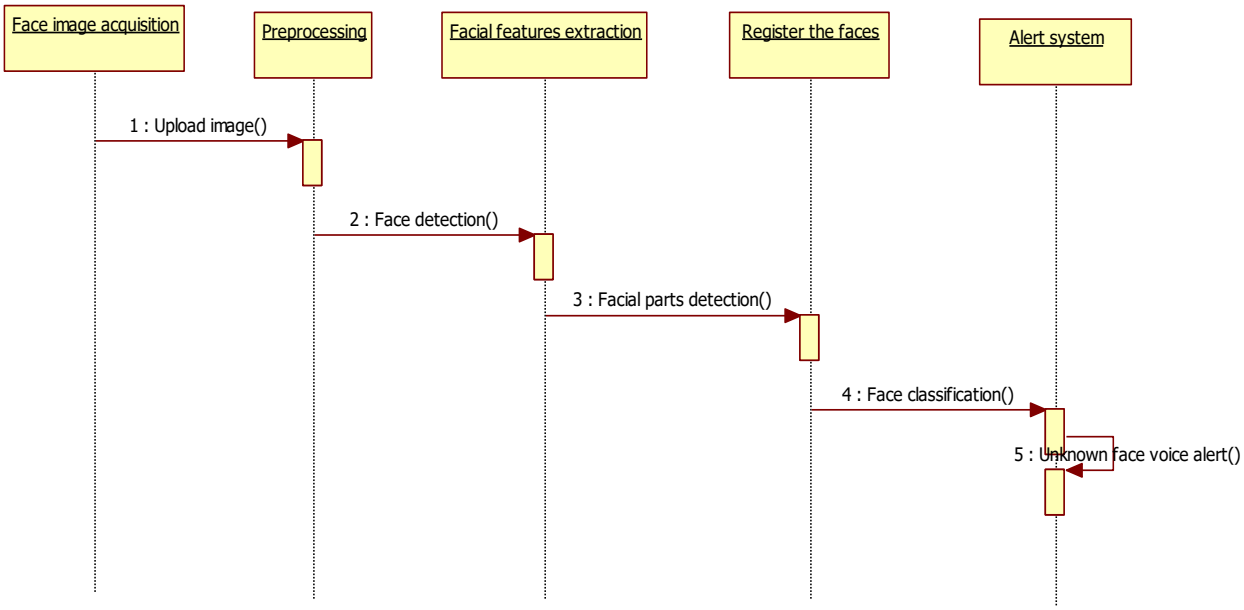


Fig 4.3.3. Sequence Diagram

4.3.4 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

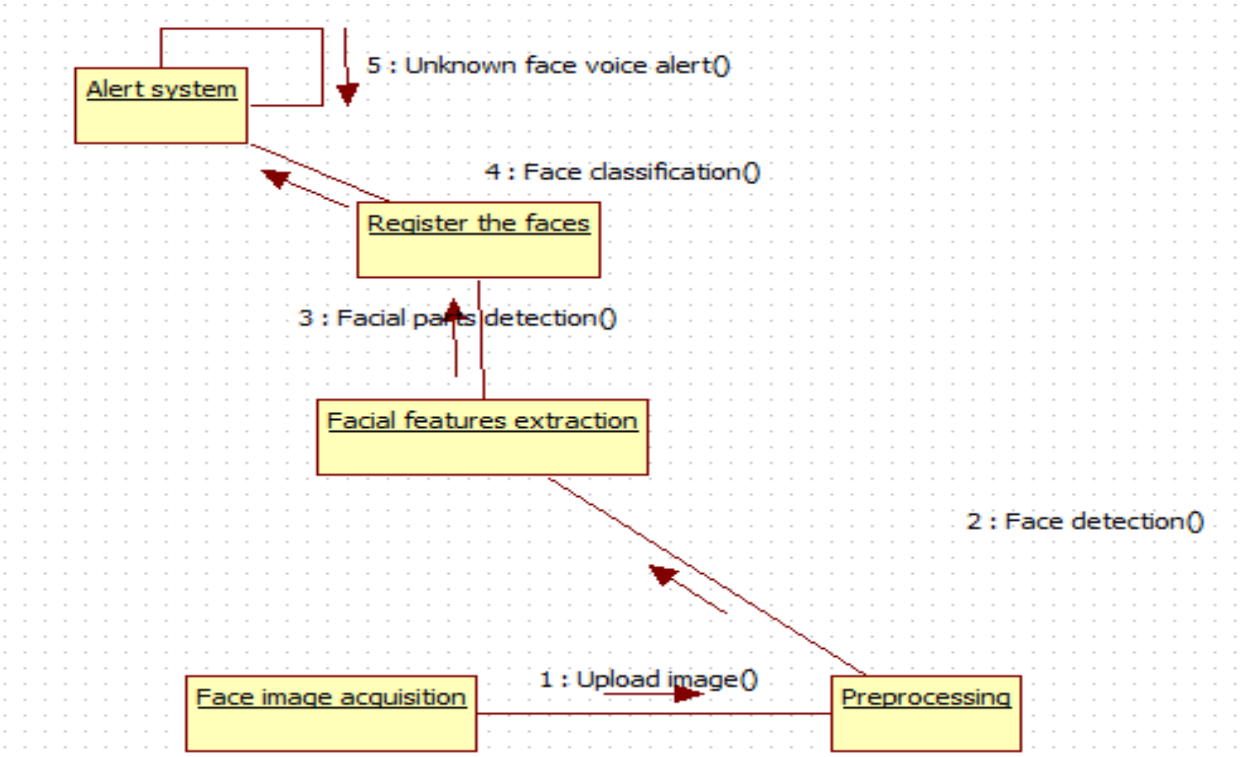


Fig 4.3.4. Collaboration Diagram

4.3.5 ACTIVITY DIAGRAM

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

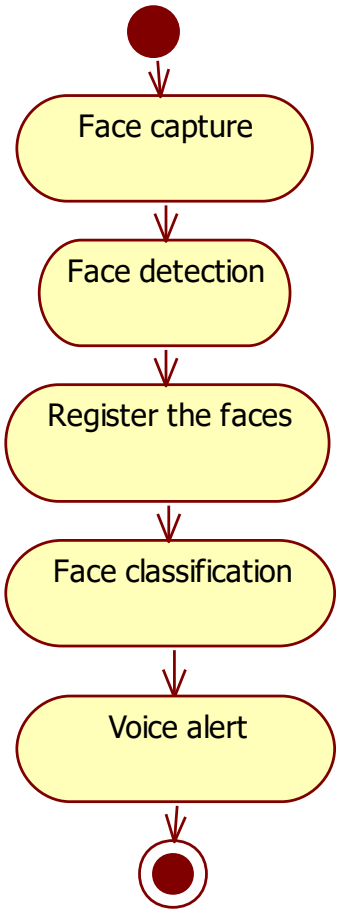


Fig 4.3.5. Class Diagram

5. DESIGN AND MODULE DESCRIPTION

5.1 MODULES

- FACE IMAGE ACQUISITION
- FEATURES EXTRACTION
- REGISTRATION OF FACE BIOMETRICS
- FACE CLASSIFICATION
- ALERT SYSTEM

5.2 MODULES DESCRIPTION

5.2.1 FACE IMAGE ACQUISITION:

A face recognition system is a computer application capable of identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a face database. Recognition algorithms can be divided into two main approaches, geometric, which look at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances. Face recognition from image or video is a popular topic in biometrics research. Face recognition is an interesting and successful application of Pattern recognition and Image analysis. Facial images are essential for intelligent vision-based human computer interaction. Face processing is based on the fact that the information about a user's identity can be extracted from the images and the computers can act accordingly. Many public places usually have surveillance cameras for video capture and these cameras have their significant value for security purpose. It is widely acknowledged that the face recognition has played an important role in surveillance system as it doesn't need the object's cooperation. In this module, admin can train multiple faces. Face may be captured through web cameras or uploaded as still pictures. In this image, user faces without occlusion, straight pose and normal light conditions.

5.2.2 FEATURES EXTRACTION:

Applying human visual property in the recognition of faces, people can identify face from very far distance, even the details are vague. That means the symmetry characteristic is enough to be recognized. Human face is made up of eyes, nose, mouth and chin etc. There are differences in shape, size and structure of those organs, so the faces are differ in thousands ways, and we can describe them with the shape and structure of the organs so as to recognize them. One common method is to extract the shape of the eyes, nose, mouth and chin, and then distinguish the faces by distance and scale of those organs. The other method is to use deformable model to describe the shape of the organs on face subtly. This module, facial features are extracted. And constructed as feature vectors. Facial features include nose part, eye parts and lip part. These values are stored is in the form of matrix.

5.2.3 REGISTRATION OF FACE BIOMETRICS:

Face registration is the process of transforming different sets of data into one coordinate system. Facial features are stored with labels. Image registration or image alignment algorithms can be classified into intensity-based and feature-based. Face recognition systems identify people by their face images. Face recognition systems establish the presence of an authorized person rather than just checking whether a valid identification (ID) or key is being used or whether the user knows the secret personal identification numbers (Pins) or passwords. One of the images is referred to as the reference or source and the others are respectively referred to as the target, sensed or subject images. Image registration involves spatially registering the target image(s) to align with the reference image. Intensity-based methods compare intensity patterns in images via correlation metrics, while feature-based methods find correspondence between image features such as points, lines, and contours. Intensity-based methods register entire images or sub-images. Feature-based methods establish a correspondence between a number of especially distinct points in images. Labeling the faces using their names. Face image registration is the process of transforming different sets of data into one coordinate system. Data may be multiple photographs, data from different sensors, times, depths, or viewpoints.

5.2.4 FACE CLASSIFICATION:

Face recognition have gained a great deal of popularity because of the wide range of applications such as in entertainment, smart cards, information security, law enforcement, and surveillance. It is a relevant subject in pattern recognition, computer vision, and image processing. Face identification is a one-to-many matching process that compares a query face image against all the template images in a face database to determine the identity of the query face. The identification of the test image is done by locating the image in the database that has the highest similarity with the test image. The identification process is a “closed” test, which means the sensor takes an observation of an individual that is known to be in the database. This module is known as login phase or testing phase. Input is in the form of real time video capturing. Video images are splited into still images. Face detection is done in the process. Matching the features using CNN algorithm. The temporal information in video sequences enables the analysis of facial dynamic changes and its application as a biometric identifier for person recognition. We have utilize the human nature that human will have at least small amount of movements such as eyes blinking and/or mouth and face boundary movements. We can get this information easily because dealing with video sequence by which the whole sequence of the object's movements can be obtained.

5.2.5 ALERT SYSTEM:

In many of the access control applications, such as door open, the size of the group of people that need to be recognized is relatively small. The face pictures are also caught under natural conditions, such as frontal faces and indoor illumination. If the feature vectors are not matched means, considered as unknown faces. Create alert for unknown labeling. Finally provide alert, Email Alert to authorized person.

6. IMPLEMENTATION

6.1 FRONT END:

6.1.1 HTML:

The Hyper Text Markup Language, or HTML(Hyper Text Markup Language) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

6.1.2 CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents. In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

6.1.3 PHP:

PHP is a general-purpose scripting language especially suited to web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.[PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Preprocessor. PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. Arbitrary PHP code can also be interpreted and executed via command-line interface (CLI). The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow. Since 2014, work has gone on to create a formal PHP specification. As of January 2021, 72% of PHP websites use discontinued versions of PHP, i.e. PHP 7.2 or lower, which are no longer supported by The PHP Development Team. A large additional fraction uses PHP 7.3, which is only (up to December 6, 2021) "supported for critical security issues only."[13] Over 40% of all PHP websites use version 5.6 or older,[14] that not even Debian supports (Debian 9 supported version 7.0 and 7.1).

6.2 BACK END:

6.2.1 PYTHON:

Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms,including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3.Python consistently ranks as one of the most popular programming languages.

Python is developed under an OSI-approved open-source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation. The community hosts conferences and meetups, collaborates on code, and much more. Python's documentation will help you along the way, and the mailing lists will keep you in touch.

6.2.2 DATABASE SERVER:

6.2.2.1 WAMP SERVER:

Stands for "Windows, Apache, MySQL, and PHP." WAMP is a variation of LAMP for Windows systems and is often installed as a software bundle (Apache, MySQL, and PHP). It is often used for web development and internal testing, but may also be used to serve live websites. The most important part of the WAMP package is Apache (or "Apache HTTP Server") which is used run the web server within Windows. By running a local Apache web server on a Windows machine, a web developer can test webpages in a web browser without publishing them live on the Internet. WAMP also includes MySQL and PHP, which are two of the most common technologies used for - creating dynamic websites. MySQL is a high-speed database, while PHP is a scripting language that can be used to access data from the database. By installing these two components locally, a developer can build and test a dynamic website before publishing it to a public web server. While Apache, MySQL, and PHP are open source components that can be installed individually, they are usually installed together. One popular package is called "WampServer," which provides a user-friendly way to install and configure the "AMP" components on Windows.

6.2.2.2 PHPMYADMIN:

phpMyAdmin is an open-source software tool introduced on September 9, 1998, which is written in PHP. Basically, it is a third-party tool to manage the tables and data inside the database. phpMyAdmin supports various type of operations on MariaDB and MySQL. The main purpose of phpMyAdmin is to handle the administration of MySQL over the web.

It is the most popular application for MySQL database management. We can create, update, drop, alter, delete, import, and export MySQL database tables by using this software. phpMyAdmin also supports a wide range of operation like managing databases, relations, tables, columns, indexes, permissions, and users, etc., on MySQL and MariaDB. These operations can be performed via user interface, while we still have the ability to execute any SQL statement.

phpMyAdmin is translated into 72 languages and also supports both RTL and LTR languages so that the wide range of people can easily use this software. We can run MySQL queries, repair, optimized, check tables, and also execute other database management commands. phpMyAdmin can also be used to perform administrative tasks such as database creation, query execution.

An administrator's tool of sorts, phpMyAdmin is a PHP script meant for giving users the ability to interact with their MySQL databases. WordPress stores all of its information in the MySQL database and interacts with the database to generate information within your WordPress site. A "raw" view of the data, tables and fields stored in the MySQL database is accessible through phpMyAdmin.

The phpMyAdmin program is handy for performing maintenance operations on tables, backing up information, and editing things directly in the event that WordPress is not working. Occasionally, in the Support Forums, someone will post a SQL query of some benefit or other that can be run using phpMyAdmin. Although many of the same tasks can be performed on the MySQL command line, doing so is not an option for many people.

6.3 SAMPLE CODING:

```
from flask import Flask, render_template, flash, request,session
from wtforms import Form, TextField, TextAreaField, validators, StringField, SubmitField
from werkzeug.utils import secure_filename
import mysql.connector
import tkinter as tk
from tkinter import *
import cv2
import csv
import os
i-mport numpy as np
from PIL import Image,ImageTk
import pandas as pd
import datetime
import time

app = Flask(__name__)
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
class ReusableForm(Form):
   name = TextField('Name:', validators=[validators.required()])
def trainimg():
   recognizer = cv2.face.LBPHFaceRecognizer_create()
   global detector
   detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
   try:
      global faces,Id
      faces, Id = getImagesAndLabels("TrainingImage")
```

```
    except Exception as e:
        l='please make "TrainingImage" folder & put Images'
       # Notification.configure(text=l, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
        #Notification.place(x=350, y=400)

    recognizer.train(faces, np.array(Id))
    try:
        recognizer.save("TrainingImageLabel\Trainner.yml")
    except Exception as e:
        q='Please make "TrainingImageLabel" folder'
        #Notification.configure(text=q, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
        #Notification.place(x=350, y=400)
-
    res = "Model Trained"  # +",".join(str(f) for f in Id)
    #Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18, 'bold'))
    #Notification.place(x=250, y=400)
def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faceSamples = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image
```

```
        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y + h, x:x + w])
            Ids.append(Id)
    return faceSamples, Ids
def del_sc1():
    sc1.destroy()
def err_screen():
    global sc1
    sc1 = tk.Tk()
    sc1.geometry('300x100')
    sc1.iconbitmap('AMS.ico')
    sc1.title('Warning!!')
    sc1.configure(background='snow')
    Label(sc1,text='Enrollment & Name')

required!!!',fg='red',bg='white',font=('times', 16, ' bold ')).pack()
    Button(sc1,text='OK',command=del_sc1,fg="black"  ,bg="lawn green"  ,width=9  ,height=1,
activebackground = "Red" ,font=('times', 15, ' bold ')).place(x=90,y= 50)


def del_sc2():
    sc2.destroy()
def err_screen1():
    global sc2
    sc2 = tk.Tk()
```

```
    sc2.geometry('300x100')
    sc2.iconbitmap('AMS.ico')
    sc2.title('Warning!!')
    sc2.configure(background='snow')
    Label(sc2,text='Please enter your subject name!!!',fg='red',bg='white',font=('times', 16, ' bold
')).pack()
    Button(sc2,text='OK',command=del_sc2,fg="black"  ,bg="lawn green"  ,width=9  ,height=1,
activebackground = "Red" ,font=('times', 15, ' bold ')).place(x=90,y= 50)

def Fillattendances():
        sub = "trest"
        now = time.time()  ###For calculate seconds of video
        future = now + 20
        if time.time() < future:
            if sub == '':
                err_screen1()
            else:
                recognizer = cv2.face.LBPHFaceRecognizer_create()  #
cv2.createLBPHFaceRecognizer()
                try:
                    recognizer.read("TrainingImageLabel\Trainner.yml")
                except:
                    e = 'Model not found,Please train model'
                    # Notifica.configure(text=e, bg="red", fg="black", width=33, font=('times', 15,
'bold'))
                    # Notifica.place(x=20, y=250)

                harcascadePath = "haarcascade_frontalface_default.xml"
                faceCascade = cv2.CascadeClassifier(harcascadePath)
```

```
        df = pd.read_csv("StudentDetails\StudentDetails.csv")
        cam = cv2.VideoCapture(0)
        font = cv2.FONT_HERSHEY_SIMPLEX
        col_names = ['Enrollment', 'Name', 'Date', 'Time']
        attendance = pd.DataFrame(columns=col_names)
        while True:
            ret, im = cam.read()
            gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
            faces = faceCascade.detectMultiScale(gray, 1.2, 5)
            for (x, y, w, h) in faces:
                global Id

                Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
                if (conf < 70):
                    print(conf)
                    global Subject
                    global aa
                    global date
                    global timeStamp
Subject = "sample5"
                    ts = time.time()
                    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
                    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                    aa = df.loc[df['Enrollment'] == Id]['Name'].values
                    global tt
                    tt = str(Id) + "-" + aa
                    En = '15624031' + str(Id)
                    attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]
                    cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
```

```
                cv2.putText(im, str(tt), (x + h, y), font, 1, (255, 255, 0,), 4)

            else:
                Id = 'Unknown'
                tt = str(Id)
                cv2.rectangle(im, (x, y), (x + w, y + h), (0, 25, 255), 7)
                cv2.putText(im, str(tt), (x + h, y), font, 1, (0, 25, 255), 4)
        if time.time() > future:
            break

        attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
        cv2.imshow('Filling attedance..', im)
        key = cv2.waitKey(30) & 0xff
        if key == 27:
            break

    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour, Minute, Second = timeStamp.split(":")
    fileName = "Attendance/" + Subject + "_" + date + "_" + Hour + "-" + Minute + "-" +
Second + ".csv"
    attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
    print(attendance)
    attendance.to_csv(fileName, index=False)

    ##Create table for Attendance
    date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
    DB_Table_name = str(Subject + "_" + date_for_DB + "_Time_" + Hour + "_" +
```

```
Minute + "_" + Second)
        import pymysql.connections

        ###Connect to the database
        try:
            global cursor
            connection = pymysql.connect(host='localhost', user='root', password='', db='face')
            cursor = connection.cursor()
        except Exception as e:
            print(e)

        sql = "CREATE TABLE " + DB_Table_name + """
        (ID INT NOT NULL AUTO_INCREMENT,
         ENROLLMENT varchar(100) NOT NULL,
         NAME VARCHAR(50) NOT NULL,
         DATE VARCHAR(20) NOT NULL,
         TIME VARCHAR(20) NOT NULL,
            PRIMARY KEY (ID)
            );
        """
        ####Now enter attendance in Database
        insert_data = "INSERT INTO sample(ID,ENROLLMENT,NAME,DATE,TIME)
VALUES (0, %s, %s, %s,%s)"
        VALUES = (str(Id), str(aa), str(date), str(timeStamp))
        try:
            cursor.execute(sql)  ##for create a table
            cursor.execute(insert_data, VALUES)  ##For insert data into table
        except Exception as ex:
            print(ex)  #
```

```
            M = 'Attendance filled Successfully'
            print(M)
            print(Id)
            print(str(aa))
            print(date)
            listToStr = ''.join(map(str, aa))
            print(timeStamp)
            conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
            cursor = conn.cursor()
            cursor.execute("select * from studentatten where sid='" + str(Id)+ "' and name='" +
str(listToStr) + "' and date='"+str(date)+"'")
            data = cursor.fetchone()
            if data is None:
                print("hai")
                conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
                cursor = conn.cursor()
                cursor.execute(
                    "insert into studentatten values('" + str(Id) + "','" + str(listToStr) + "','" + str(
                        date) + "','" + str(timeStamp) + "')")
                conn.commit()
                conn.close()

            else:
                conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
                cursor = conn.cursor()
```

```
            cursor.execute("update studentatten set otime='"+str(timeStamp)+"'")
            conn.commit()
            conn.close()

        #Notifica.configure(text=M, bg="Green", fg="white", width=33, font=('times', 15,
'bold'))
        #Notifica.place(x=20, y=250)

        cam.release()
        cv2.destroyAllWindows()

        import csv
        import tkinter
        root = tkinter.Tk()
        root.title("Attendance of " + Subject)
        root.configure(background='snow')
        cs = 'C:/Users/acer/PycharmProjects/student/' + fileName
        with open(cs, newline="") as file:
            reader = csv.reader(file)
            r = 0

            for col in reader:
                c = 0
                for row in col:
                    # i've added some styling
                    label = tkinter.Label(root, width=8, height=1, fg="black", font=('times', 15, '
bold '),
                                          bg="lawn green", text=row, relief=tkinter.RIDGE)
                    label.grid(row=r, column=c)
```

```
                    c += 1
                r += 1
            root.mainloop()
            print(attendance)
@app.route("/")
def homepage():

    return render_template('index.html')

@app.route("/admin")
def admin():

    return render_template('admin.html')

@app.route("/adminhome")
def adminhome():
    trainimg()
    conn = mysql.connector.connect(user='root', password='', host='localhost', database='student')
    cursor = conn.cursor()
    cursor.execute("select * from register")
    data = cursor.fetchall()



    return render_template('adminhome.html',data=data)

@app.route("/addstudent")
def addstudent():

    return render_template('addstudent.html')
```

```
@app.route("/view")
def view():-


    return render_template('view.html')


@app.route("/student")
def student():


    return render_template('student.html')


@app.route("/adminlog",methods=['GET','POST'])
def adminlog():
    if request.method == 'POST':
        uname=request.form['uname']
        password=request.form['password']
        print(uname)
        print(password)
        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
        cursor = conn.cursor()
```

```
        cursor.execute("select * from admin where uname='"+uname+"' and
password='"+password+"'")
        data=cursor.fetchone()
        if data is None:
            return "user name and password incorrect"
        else:
            return render_template("adminhome.html")


@app.route("/studentregister", methods=['GET', 'POST'])
def studentregister():
    if request.method == 'POST':
        studentid = request.form['staffid']
        name = request.form['name']
        gender = request.form['gender']
        bgroup = request.form['bgroup']
        desg = ''
        depart = request.form['depart']
        subject = ''
        email = request.form['email']
        pnumber = request.form['pnumber']
```

```
        address = request.form['address']

        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')

        cursor = conn.cursor()

        cursor.execute("insert into register
values('"+studentid+"','"+name+"','"+gender+"','"+bgroup+"','"+desg+"','"+depart+"','"+subject+"'
,'"+email+"','"+pnumber+"','"+address+"',')")

        conn.commit()

        conn.close()


        return render_template("register1.html")


@app.route("/studentlogin",methods=['GET','POST'])

def studentlogin():

    if request.method == 'POST':

        uname=request.form['uname']

        password=request.form['password']

        session['sid']=password

        print(uname)

        print(password)
```

```
        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
        cursor = conn.cursor()
        cursor.execute("select * from register where name='"+uname+"' and
studentid='"+password+"'")
        data=cursor.fetchone()
        if data is None:
            return "user name and password incorrect"
        else:
            conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
            cursor = conn.cursor()
            cursor.execute("select * from register where studentid='" + password + "'")
            data = cursor.fetchall()
            return render_template("studenthome.html",data=data)


@app.route("/getimage",methods=['GET','POST'])
def getimage():
    if request.method == 'POST':


        l1 =request.form['sid']
```

```
l2 =request.form['name']
if l1 == '':
    err_screen()
elif l2 == '':
    err_screen()
else:
    try:
        cam = cv2.VideoCapture(0)
        detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
        Enrollment = request.form['sid']
        Name = request.form['name']
        sampleNum = 0
        while (True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x, y, w, h) in faces:
                cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                # incrementing sample number
                sampleNum = sampleNum + 1
```

```
                # saving the captured face in the dataset folder
                cv2.imwrite("TrainingImage/ " + Name + "." + Enrollment + '.' + str(sampleNum)
+ ".jpg",
                            gray[y:y + h, x:x + w])
                cv2.imshow('Frame', img)
            # wait for 100 miliseconds
            # break if the sample number is morethan 100
            elif sampleNum > 70:
                break
        cam.release()
        cv2.destroyAllWindows()
        ts = time.time()
        Date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
        Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        row = [Enrollment, Name, Date, Time]
        with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
            writer = csv.writer(csvFile, delimiter=',')
            writer.writerow(row)
            csvFile.close()
        res = "Images Saved for Enrollment : " + Enrollment + " Name : " + Name + "Model
Trained successfully"
```

```
            trainimg()
            return res
            ##Notification.place(x=250, y=400)
        except FileExistsError as F:
            f = 'Student Data already exists'
            #Notification.configure(text=f, bg="Red", width=21)
            #Notification.place(x=450, y=400)
@app.route("/Attendance")
def testatten():
    Fillattendances()
    return "Attendance Successfully"
@app.route("/view1",methods=['GET','POST'])
def view1():
    if request.method == 'POST':
        date = request.form['date']
        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
        cursor = conn.cursor()
        cursor.execute("select * from studentatten where date='" + date + "'")
        data = cursor.fetchall()
```

```
        return render_template("view.html",data=data)
@app.route("/studenthome")
def studenthome():
        sid = session['sid']
        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
        cursor = conn.cursor()
        cursor.execute("select * from register where studentid='" + sid + "'")
        data = cursor.fetchall()
        return render_template("studenthome.html",data=data)
@app.route("/sview")
def sview():
        sid = session['sid']
        conn = mysql.connector.connect(user='root', password='', host='localhost',
database='student')
        cursor = conn.cursor()
        cursor.execute("select * from studentatten where sid='" + sid + "'")
        data = cursor.fetchall()
        return render_template("studentview.html", data=data)
if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```

6.4 SCREENSHOTS:

6.4.1 WELCOME PAGE:

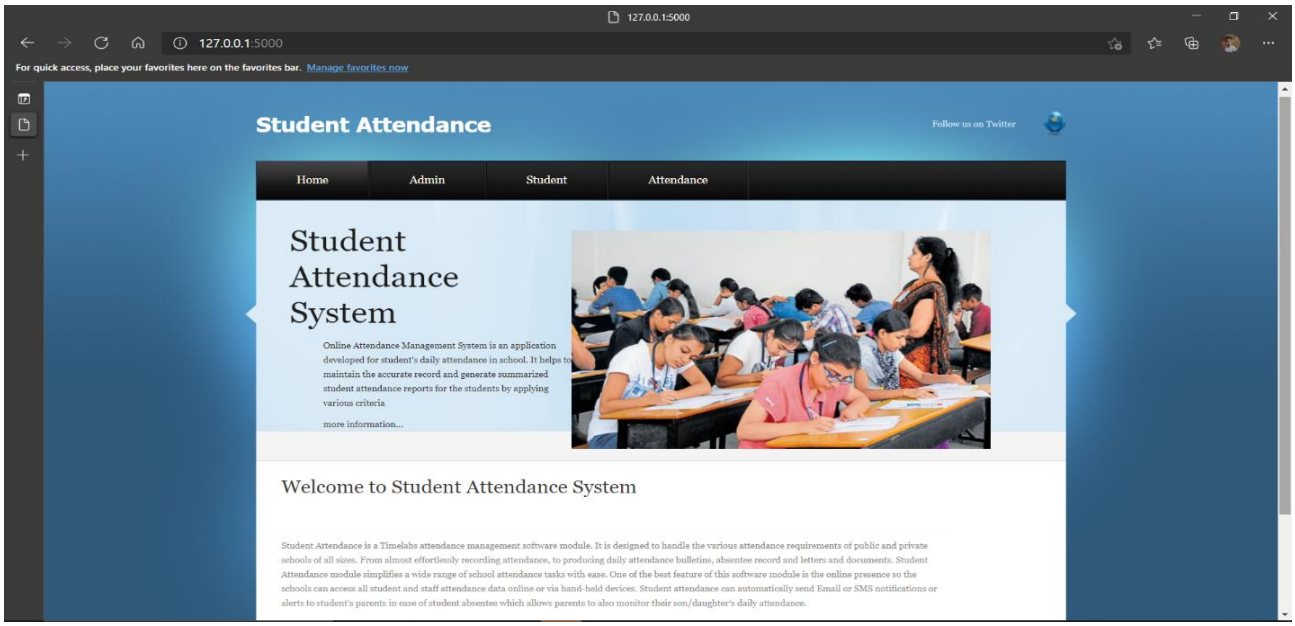


Fig 6.4.1. Welcome Page

6.4.2 ADMIN HOME:

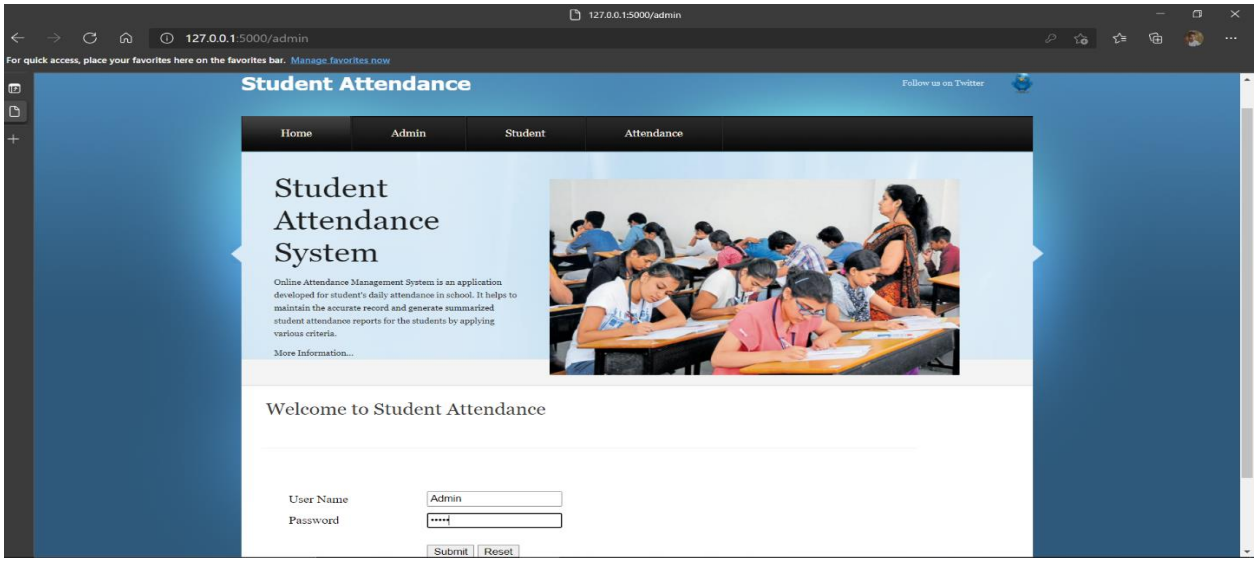


Fig 6.4.2. Admin Page

6.4.3 STUDENT DETAILS:

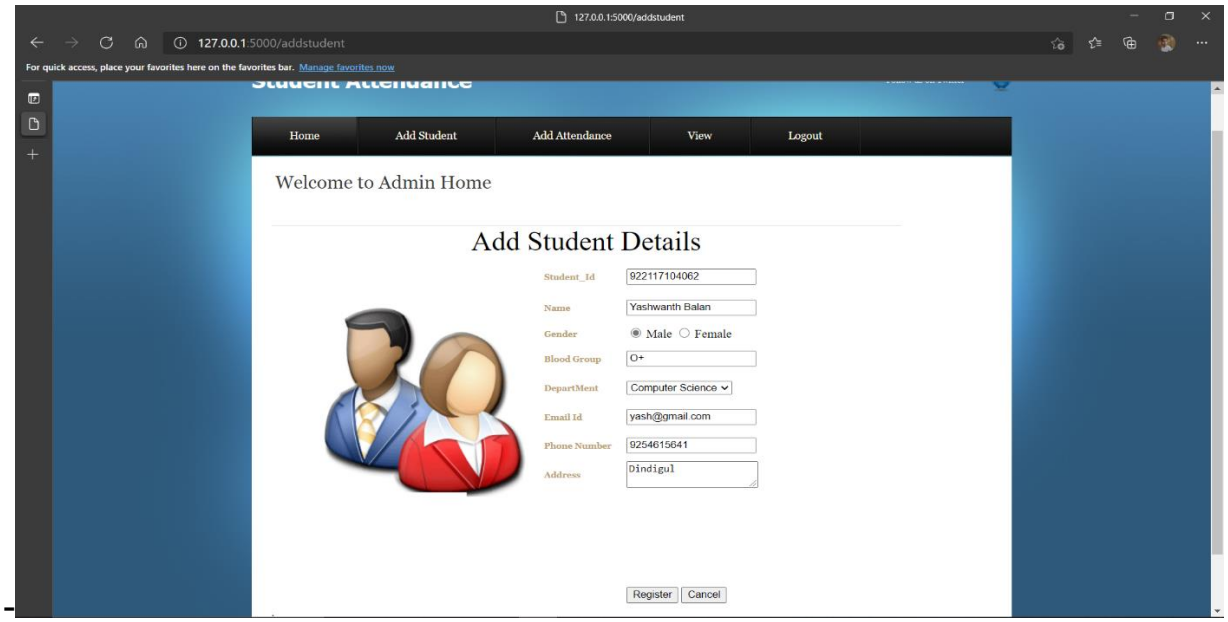


Fig 6.4.3. Student Details

6.4.4 FACE REGISTRATION:

6.4.4.1 FACE REGISTRATION-1

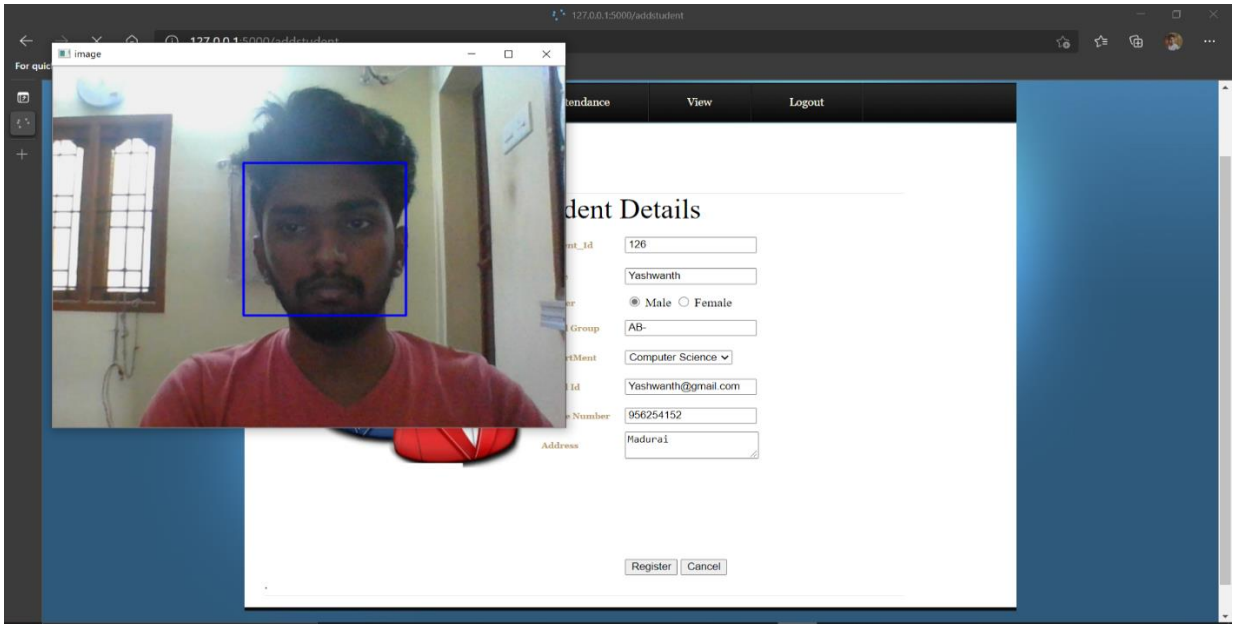


Fig 6.4.4.1 Face Registration

6.4.4.2 FACE REGISTRATION-2

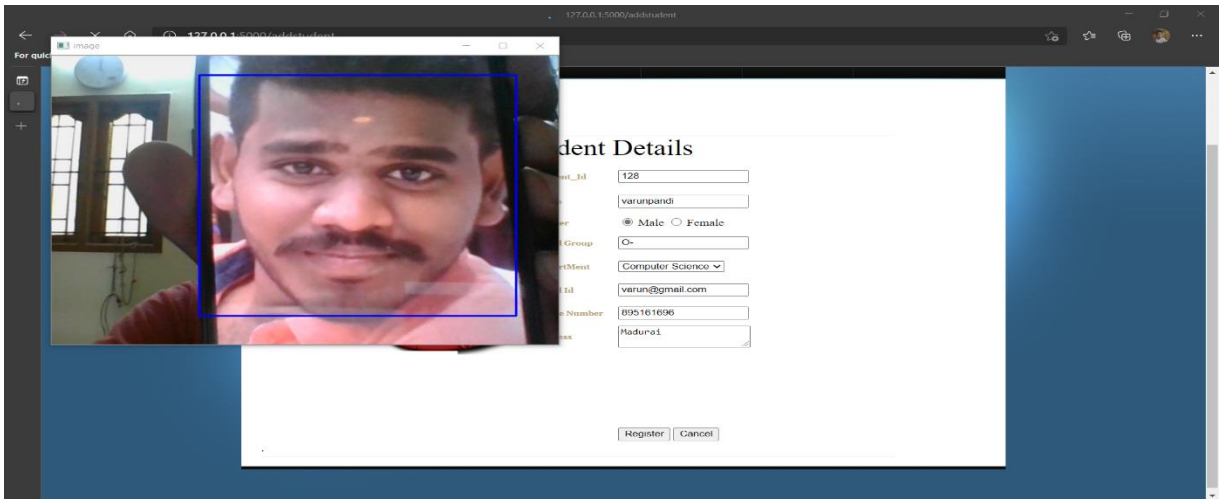


Fig 6.4.4.2. Face Registration

6.4.4.3 FACE REGISTRATION-3

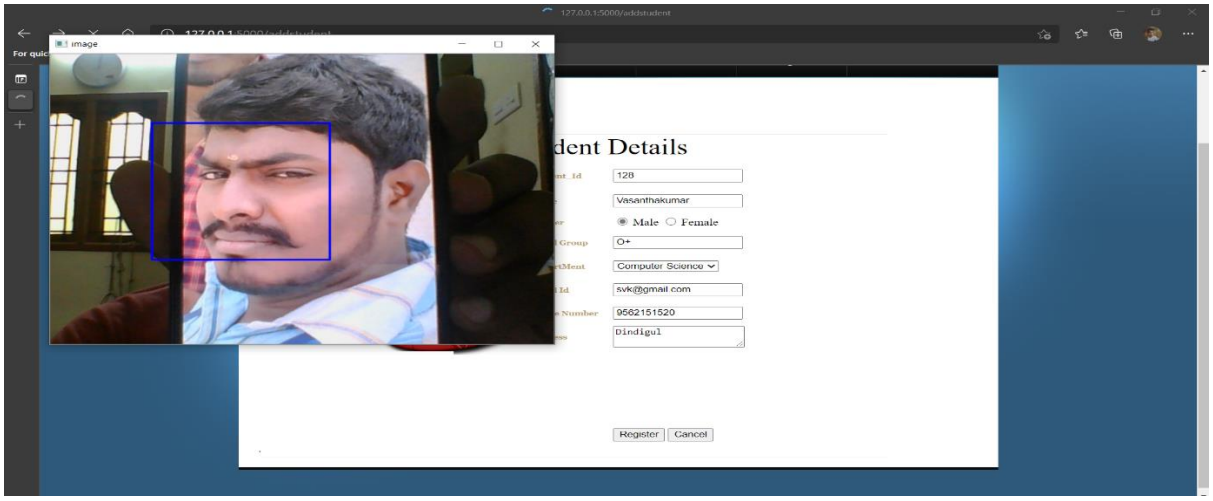


Fig 6.4.4.3. Face Registration

6.4.5 TRAINING FACE:

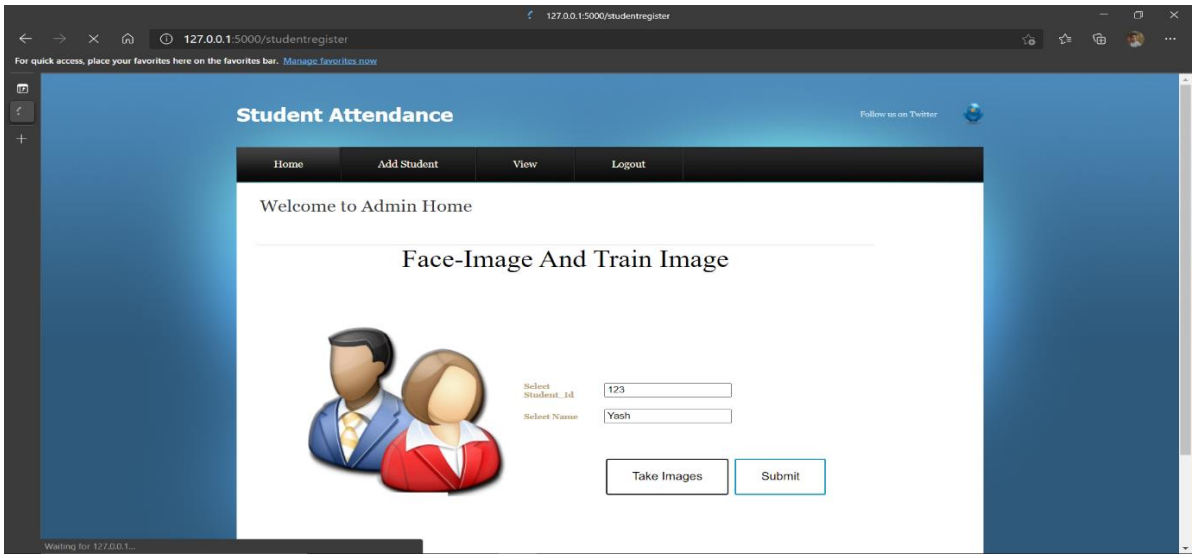


Fig 6.4.5. Face Training

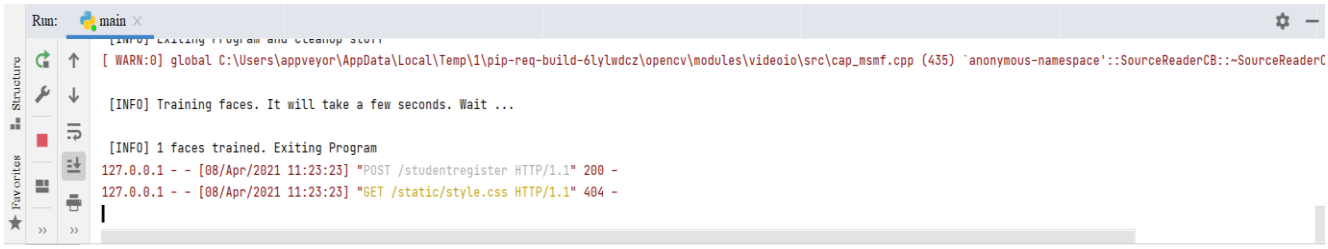


Fig 6.4.5.1. Face Training

6.4.6 DATASET CLASSIFICATION:

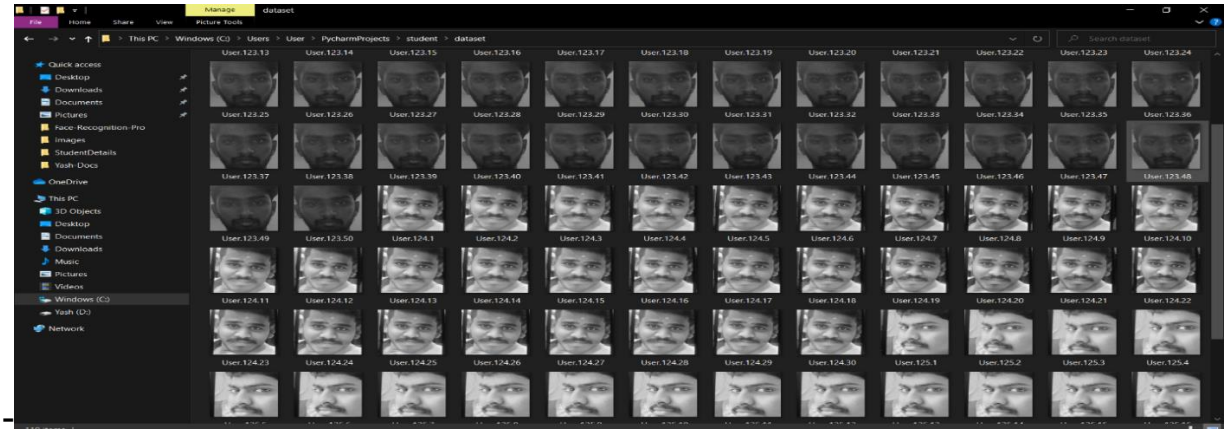


Fig 6.4.6. Dataset Classification

6.4.7 ATTENDANCE PERSON:

6.4.7.1 ATTENDANCE-PERSON-1:

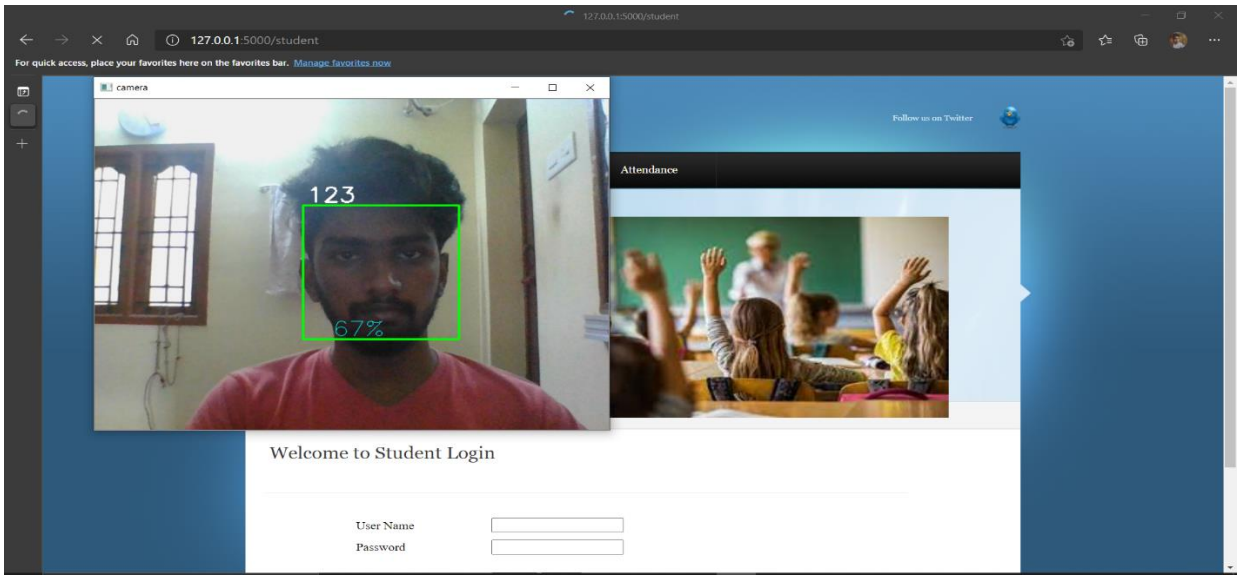


Fig 6.4.7.1 Attendance-Person 1

6.4.7.2 ATTENDANCE-PERSON 2:

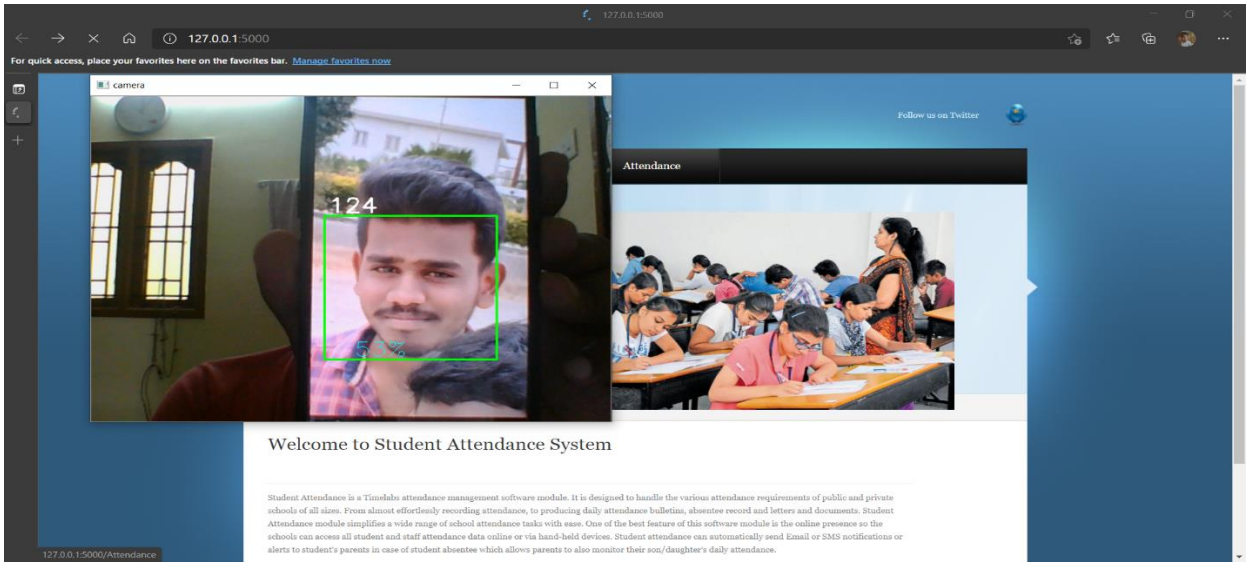


Fig 6.4.7.2 Attendance-Person 2

6.4.7.3 ATTENDANCE-PERSON 3:

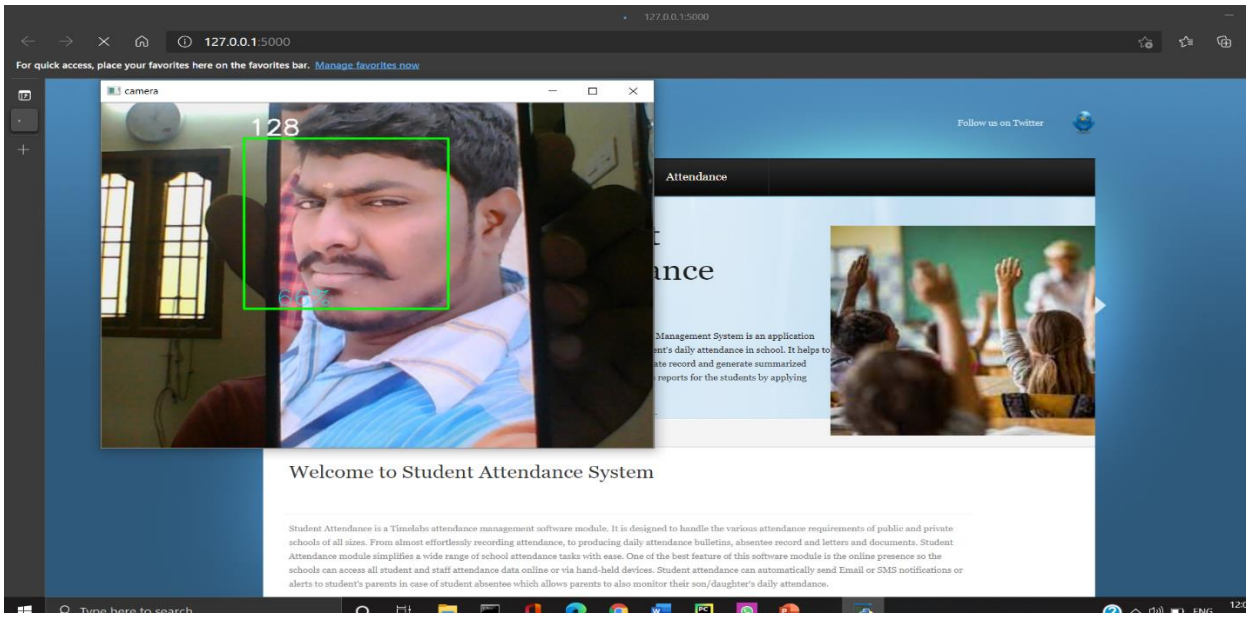


Fig. 6.4.7.3 Attendance-Person 2

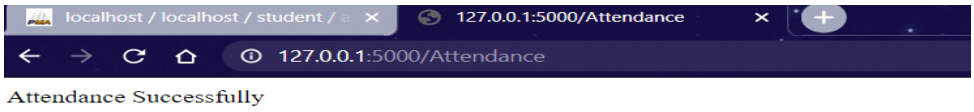


Fig 6.4.7.4 Attendance Successful

6.4.8 ATTENDANCE DETAILS:

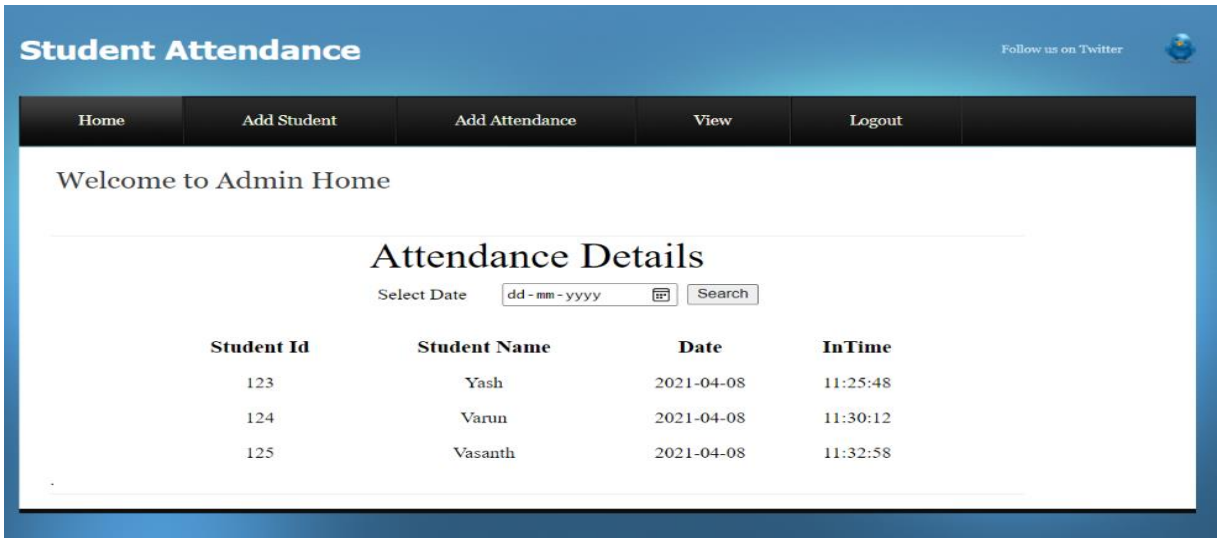


Fig 6.4.8. Attendance Details

6.4.9 UNAUTHORIZED ACCESS:

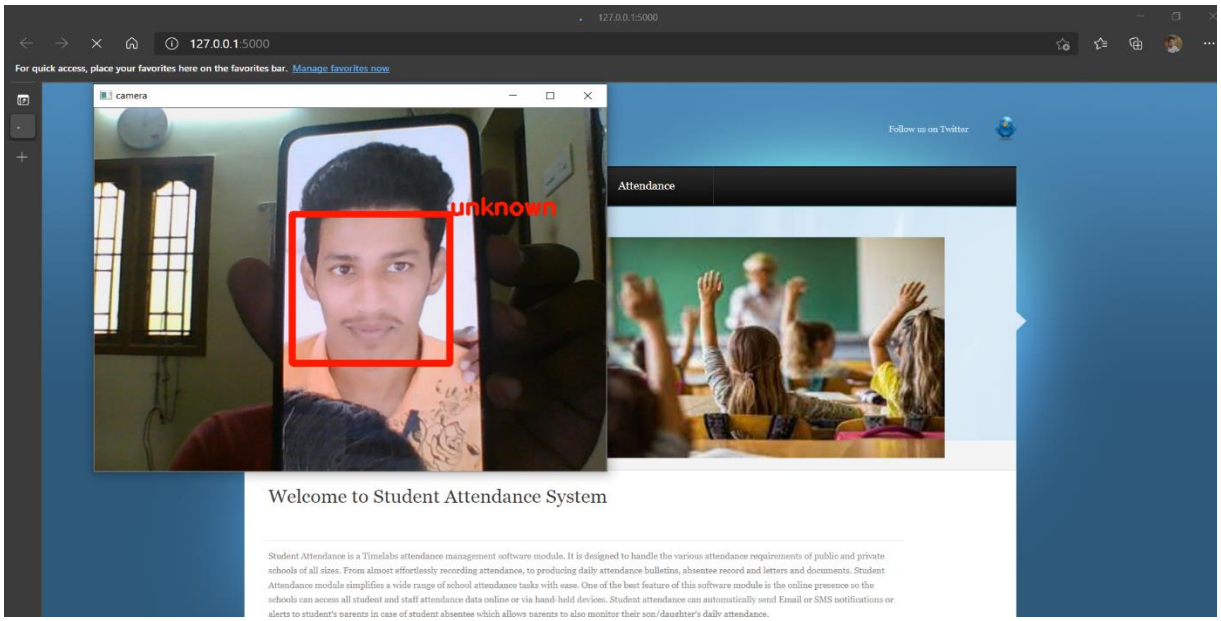


Fig 6.4.9. Un-Authorized User

6.4.10 ALERT SYSTEM:

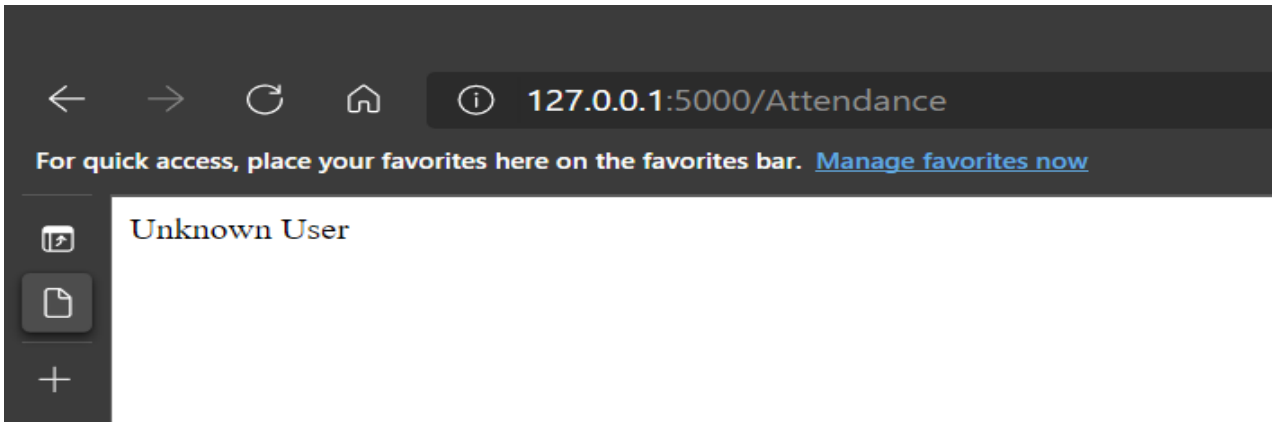


Fig 6.4.10. Alert Message

6.5 RESULT AND DISCUSSION:

- ✓ In this System, we have obtained the result of Automatic Face Recognition Student Attendance System using Advanced HAAR Cascade Algorithm with improved accuracy.
- ✓ Finally, Student's attendance is automatically collected, verified and stored in the database successfully.

7. TESTING

Software testing is a method of assessing the functionality of a software program. There are many different types of software testing but the two main categories are dynamic testing and static testing. Dynamic testing is an assessment that is conducted while the program is executed; static testing, on the other hand, is an examination of the program's code and associated documentation. Dynamic and static methods are often used together.

Testing is a set activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it is vital success of the system.

Testing Objectives:

There are several rules that can serve as testing objectives, they are

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has high probability of finding an undiscovered error.
3. A successful test is one that uncovers an undiscovered error.

If testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrates that software functions appear to the working according to the specification, that performance requirements appear to have been met.- There are three ways to test a program

1. For Correctness
2. For Implementation efficiency
3. For Computational Complexity.

Tests for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

Tests used for implementation efficiency attempt to find ways to make a correct program faster or use less storage. It is a code-refining process, which reexamines the implementation phase of algorithm development. Tests for computational complexity amount to an experimental analysis

of the complexity of an algorithm or an experimental comparison of two or more algorithms, which solve the same problem.

The data is entered in all forms separately and whenever an error occurred, it is corrected immediately. A quality team deputed by the management verified all the necessary documents and tested the Software while entering the data at all levels. The development process involves various types of testing. Each test type addresses a specific testing requirement. The most common types of testing involved in the development process are:

• Unit Test.

• Functional Test

• Integration Test

7.1 UNIT TESTING

The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test. Unit test depends upon the language on which the project is developed. Unit tests ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2 FUNCTIONAL TESTING

Functional test can be defined as testing two or more modules together with the intent of finding defects, demonstrating that defects are not present, verifying that the module performs its intended functions as stated in the specification and establishing confidence that a program does what it is supposed to do.

7.3 INTEGRATION TESTING

In integration testing modules are combined and tested as a group. Modules are typically code modules, individual applications, source and destination applications on a network, etc. Integration Testing follows unit testing and precedes system testing. Testing after the product is code complete. Betas are often widely distributed or even distributed to the public at large in hopes that they will buy the final product when it is released.

8. CONCLUSION AND FUTURE ENHANCEMENT

8.1 CONCLUSION

In this project, we reviewed face recognition technique for still images and video sequences. Most of these existing approaches need well-aligned face images and only perform either still image face recognition or video-to video match. They are not suitable for face recognition under surveillance scenarios because of the following reasons: limitation in the number (around ten) of face images extracted from each video due to the large variation in pose and lighting change; no guarantee of the face image alignment resulted from the poor video quality, constraints in the resource for calculation influenced by the real time processing. So we can propose a local facial feature-based framework for still image and video-based face recognition under surveillance conditions. This framework is generic to be capable of vide to face matching in real-time. While the training process uses static images, the recognition task is performed over video sequences. Our results show that higher recognition rates are obtained when we use video sequences rather than statics based on HAAR cascade and Convolutional Neural network algorithm. Evaluation of this approach is done for still image and video based face recognition on real time image datasets with SMS alert system.

8.2 FUTURE ENHANCEMENTS

In future work, we extend the framework to implement various algorithms to provide video face matching with improved accuracy rate. Videos provide an automatic and efficient way for feature extraction to reduce the features based on dimensionality reduction. In our further enhancement, we planned to implement a **Student Class Activity Monitoring System**. Activities like Listening the class, sleeping Etc.

9. REFERENCES

[1] Shah K., Bhandare D., Bhirud S. (2021) Face Recognition-Based Automated Attendance System. In: Gupta D., Khanna A., Bhattacharyya S., Hassanien A.E., Anand S., Jaiswal A. (eds) International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing, vol 1165. Springer, Singapore. http://doi-org-443.webvpn.fjmu.edu.cn/10.1007/978-981-15-5113-0_79

[2] Anju V Das, Anjana Shyju, Thomas Varghese, Nisha Mohan P M, "Face Recognition Based Attendance System using machine learning algorithm" International Journal of Innovative Research in Technology (IJIRT), April 2021

[3] Kumar, Praveen. "A Multiple Face Recognition System With Dlib's Resnet Network Using Deep Metric Learning." Journal of Critical Reviews 7.6 (2020): 856-859.

[4] Goud, B. P., S. Sravan Reddy, and K. Praveen Kumar. "Smart Attendance Notification System using SMTP with Face Recognition." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 9 (2020): 337-342.

[5] Gnanaprakash, P. Harini, "Smart Attendance System Using Cd-Lbp Algorithm" International Journal of Scientific & Technology Research Volume 9, Issue 02, February 2020

[6] Bussa, Sudhir, et al. "Smart Attendance System using OPENCV based on Facial Recognition." 2017ss

[7] Bhat, Aruna, et al. "Deep-learning based group-photo Attendance System using One Shot Learning." 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). IEEE, 2020.

[8] Sharanya, T., et al. "Online Attendance using Facial Recognition." 2016

[9] Elias, Shamsul J., et al. "Face recognition attendance system using Local Binary Pattern (LBP)." Bulletin of Electrical Engineering and Informatics 8.1 (2019): 239-245.

[10] Harikrishnan, J., et al. "Vision-face recognition attendance monitoring system for surveillance using deep learning technology and computer vision." 2019.

CONFERENCE CERTIFICATES:

BANNARI AMMAN
INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University,
Approved by AICTE, Accredited by NAAC with 'A' Grade
Stay Ahead

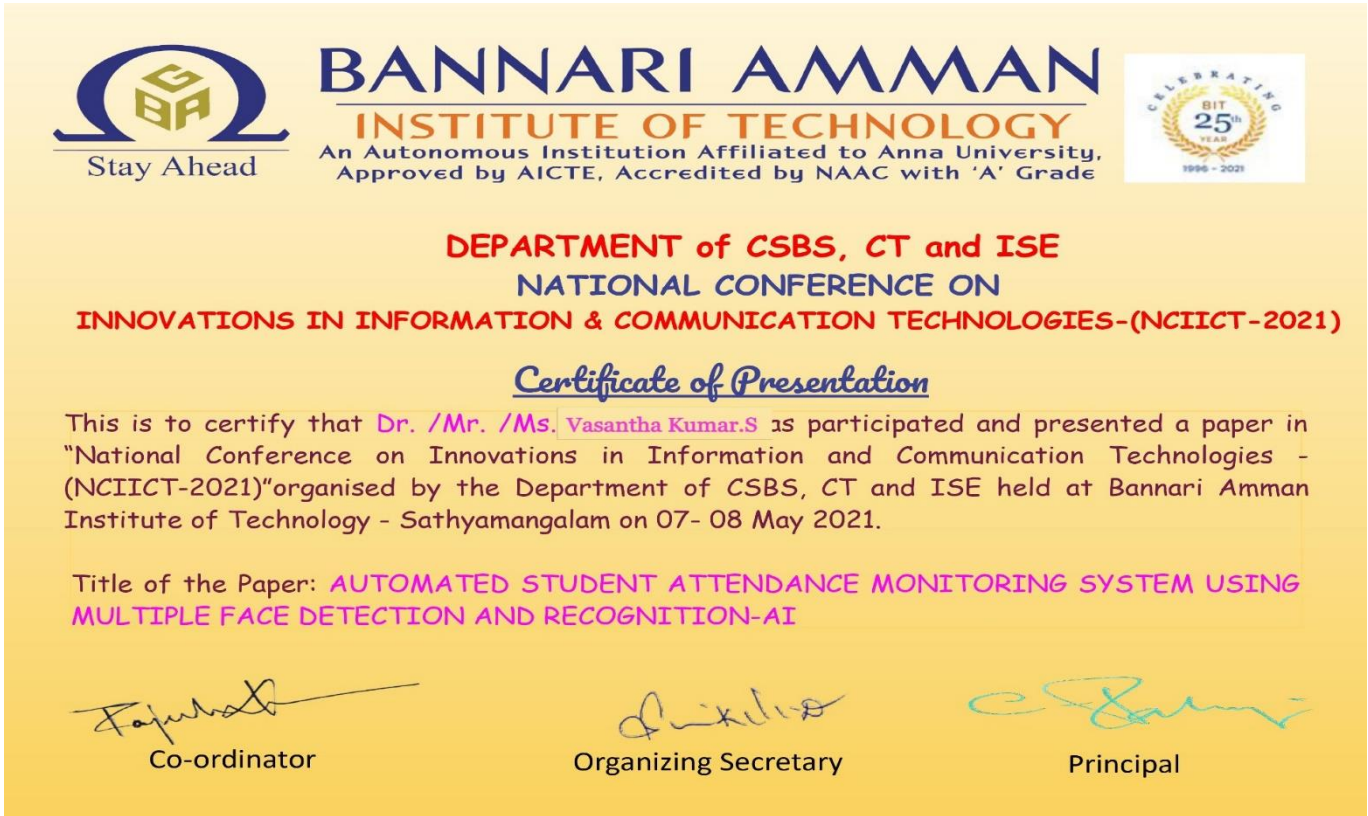
DEPARTMENT of CSBS, CT and ISE
NATIONAL CONFERENCE ON
INNOVATIONS IN INFORMATION & COMMUNICATION TECHNOLOGIES-(NCIICT-2021)

Certificate of Presentation

This is to certify that Dr. /Mr. /Ms. Varunpandi R has participated and presented a paper in "National Conference on Innovations in Information and Communication Technologies - (NCIICT-2021)"organised by the Department of CSBS, CT and ISE held at Bannari Amman Institute of Technology - Sathyamangalam on 07- 08 May 2021.

Title of the Paper: AUTOMATED STUDENT ATTENDANCE MONITORING SYSTEM USING MULTIPLE FACE DETECTION AND RECOGNITION-AI

Co-ordinator | Organizing Secretary | Principal



BANNARI AMMAN
INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University,
Approved by AICTE, Accredited by NAAC with 'A' Grade
Stay Ahead

DEPARTMENT of CSBS, CT and ISE
NATIONAL CONFERENCE ON
INNOVATIONS IN INFORMATION & COMMUNICATION TECHNOLOGIES-(NCIICT-2021)

Certificate of Presentation

This is to certify that Dr. /Mr. /Ms. Vasantha Kumar.S as participated and presented a paper in "National Conference on Innovations in Information and Communication Technologies - (NCIICT-2021)"organised by the Department of CSBS, CT and ISE held at Bannari Amman Institute of Technology - Sathyamangalam on 07- 08 May 2021.

Title of the Paper: AUTOMATED STUDENT ATTENDANCE MONITORING SYSTEM USING MULTIPLE FACE DETECTION AND RECOGNITION-AI

Co-ordinator | Organizing Secretary | Principal

Stay Ahead

BANNARI AMMAN
INSTITUTE OF TECHNOLOGY
An Autonomous Institution Affiliated to Anna University,
Approved by AICTE, Accredited by NAAC with 'A' Grade

CELEBRATING BIT 25th YEAR 1996 - 2021

DEPARTMENT of CSBS, CT and ISE
NATIONAL CONFERENCE ON
INNOVATIONS IN INFORMATION & COMMUNICATION TECHNOLOGIES-(NCIICT-2021)

Certificate of Presentation

This is to certify that Dr. /Mr. /Ms. YASHWANTH BALAN A has participated and presented a paper in "National Conference on Innovations in Information and Communication Technologies - (NCIICT-2021)"organised by the Department of CSBS, CT and ISE held at Bannari Amman Institute of Technology - Sathyamangalam on 07- 08 May 2021.

Title of the Paper: AUTOMATED STUDENT ATTENDANCE MONITORING SYSTEM USING MULTIPLE FACE DETECTION AND RECOGNITION-AI

Co-ordinator | Organizing Secretary | Principal