BIOMETRIC AUTHENTICATOR

Project work submitted in partial fulfillment of

the requirements for the degree of

MASTER OF COMPUTER APPLICATIONS

Thiruvalluvar University, Serkadu,Vellore

By

JAUSH JEBARSON R

(REGNo: 20923P08017)

Under the Guidance of

S SRINIVASAN., MCA., M.Phil.,

Assistant Professor of MCA

PG Department of Computer Applications,

Shanmuga Industries arts and science college,

Thiruvannamalai- 606603

SHANMUGA INDUSTRIES ARTS AND SCIENCE COLLEGE

(Co-Education)

(AFFILIATED TO THIRUVALLUVAR UNIVERSITY)

TIRUVANNAMALAI–606 603.

MARCH-2025

BIOMETRIC AUTHENTICATOR

Bonafide Work Done by

JAUSH JEBARSON R

(REGNo: 20923P08017)

Project work submitted in partial fulfillment of the requirement for the degree of

MASTER OF COMPUTER APPLICATIONS

Shanmuga Industries Arts & Science College - Tiruvannamalai

Affiliated college from Thiruvalluvar University,

Serkkadu,Vellore–632 115.

INTERNAL GUIDE HEAD OF THE DEPARTMENT

Submitted for the Viva-Voce Examination held on _____

External Examiner External Examiner

DECLARATION

I **JAUSH JEBARSON R** hereby declare that the project work, entitled **"BIOMETRIC AUTHENTICATOR"** submitted to the Thiruvalluvar University, in partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications is a record of original and independent project work done by me during DEC 2024 – MAR 2025 under the supervision and guidance of **S SRINIVASAN MCA., M.Phil.,** Assistant professor of PG Department of Computer Applications, Shanmuga Industries Arts and Science College, Tiruvannamalai and it has not formed the basis for the award of any Degree/ Diploma/Associate ship/Fellowship or other similar title to any Candidate in any University.

Counter signed

Internal Guide **Signature of the Candidate**

ACKNOWLEDGEMENT

I wish to acknowledge my indebtedness and deep sense of gratitude to the Honourable Secretary **Mr. L. VIJAY ANAND** and the Treasurer **Mr. E. SRIDHAR.,** Shanmuga Industries Arts and Science College, Tiruvannamalai.

Also I like to express my gratitude to **Dr.K.ANANDARAJ,M.Sc.,M.Phil.,Ph.D., Principal,** for his constant support and motivation throughout the project.

I am obliged and indebted to thank **Dr. U. UDHAYAKUMAR, M.Sc., M.Phil., Ph.D., SET, Director, Assistant Professor& Head, PG Department of Computer Applications** who helped me to complete my project successfully.

I am very thankful to my **Guide S SRINIVASAN, MCA., MPhil., Assistant professor of PG Department of Computer Applications**, to guide me to complete the dissertation successfully. I am grateful to him for his encouragement, invaluable guidance throughout my project.

I also thank all staff members of our college who had helped me in successful completion of this project.

JAUSH JEBARSON R

ABSTRACT

Biometric authentication is an advanced security method that verifies identity using unique biological traits, such as fingerprints, facial features, or iris patterns. This project focuses on developing a Biometric Authenticator system on Attendance System providing an automated, secure and efficient solution for tracking attendance in workplaces, educational institutionsand other organizations. This architecture involves integrating biometric data acquisition,preprocessing, and authentication modules with a web-based backend. The biometric data is captured using specialized hardware or APIs. This data is processed for feature extraction and securely stored in a SQL database. The authentication process compares input biometric data with stored templates using efficient matching algorithms.

TABLE OF CONTENTS

CHAPTER 1

CHAPTER 1
INTRODUCTION

INTRODUCTION

Creating and developing a computational prototype of face identification is quite challenging because faces are complicated, multidimensional, and meaningful graphicals. Additionally, faces are a natural class of objects and stand in unambiguous differences to sine-wave vents. Unlike most early visual functions, for which we may build detailed examples of retinal or striate activity, face recognition is a very high-level task for which computational approaches. Face recognition systems can currently suggest broad constraints on the corresponding neural activity. Moreover, machine learning is a high technological task that we can assign to any high-performance computers. As a result, face recognition/identification systems are considered as machine learning tool for detecting and learning about face expression of people.

In today's fast-paced world, automation and artificial intelligence have transformed various sectors, including attendance management. Traditional methods of attendance tracking, such as manual registers or RFID-based systems, often suffer from inefficiency, inaccuracies, and security vulnerabilities. To overcome these challenges, it provides a more reliable, secure, and convenient solution.

Face recognition technology uses computer vision and deep learning algorithms to identify individuals based on their facial features. By integrating this technology into an attendance management system, organizations can automate attendance tracking, minimize human intervention, and enhance accuracy. This system works by capturing images of individuals, analyzing facial landmarks, and matching them against a stored database to verify identity.

The main objective of this project is to develop a robust, real-time face recognition-based attendance system that can be used in educational institutions, workplaces, and other organizations. The system ensures efficiency, prevents proxy attendance, and reduces administrative workload.

This report provides a detailed overview of the system's development, including the methodology, algorithms used, implementation, and performance evaluation.

1.1 FESIBILITY STUDY

The Biometric Authenticator using PHP is assessed based on its technical, economic, operational, legal, and schedule feasibility to ensure successful implementation. Technically, the system is feasible as it uses widely available technologies such as PHP for backend development, MySQL for database management These tools are well-supported and compatible, ensuring efficient real-time attendance tracking. Economically, the system is cost-effective since it leverages open-source software, reducing development expenses. The primary costs involve server hosting and hardware, such as cameras, which are affordable compared to other biometric systems. Operationally, the system enhances accuracy, eliminates proxy attendance, and provides a seamless, contactless experience, making it highly beneficial for institutions and workplaces.

TECHNICAL FEASIBILITY

The Biometric Authenticator using PHP is technically feasible as it utilizes well-established and widely available technologies. The backend is developed using PHP, with MySQL for database management, ensuring efficient storage and retrieval of attendance records.The frontend is designed using HTML, CSS, and JavaScript to provide a user-friendly interface. Since all required technologies are open-source and compatible, the system can be efficiently developed and deployed with minimal technical challenges.

ECONOMICAL FEASIBILITY

The Biometric Authenticator using PHP is economically feasible as it leverages open-source technologies, reducing development costs. PHP and MySQL are free to use, eliminating the need for expensive software licenses. The primary costs involve server hosting, hardware (such as cameras), and maintenance, which remain relatively low compared to traditional biometric systems. Additionally, automating attendance tracking reduces administrative workload and operational expenses in the long run. Given its cost-effectiveness and long-term benefits, the system proves to be a financially viable solution for organizations.

1.2 OBJECTIVES OF THE STUDY

The objective of this project is to design a web based client-server application that allows users to make a registration. A server has to use face oriented computer vision to ensure that the user is a real person. Regarding the client's view, the look & feel and the usability will not be taken into account when evaluating the compliance of the objectives. The same applies to the quality of the communication between client and server. From this, it has to be inferred that the project will focus on the design of the validation system and its integration in the server. The following tables present the requirements of the application, functional and non-functional, sorted by priority

APPLICATIONS

The Face Recognition Attendance System has a broad range of applications across various sectors, offering a highly efficient, secure, and automated approach to attendance tracking. One of its primary applications is in educational institutions, such as schools, colleges, and universities, where maintaining student and faculty attendance records is crucial. Traditional methods like manual roll calls or RFID cards are time-consuming and prone to proxy attendance. By implementing face recognition technology, educational institutions can automate attendance marking, reduce human errors, and ensure accurate record-keeping.

In corporate offices, the system enhances workforce management by automating employee check-ins and check-outs. Organizations can use it to monitor working hours, track late arrivals or early departures, and integrate the data with payroll systems to ensure accurate salary calculations. Unlike fingerprint-based biometric systems, which require physical contact, face recognition provides a completely contactless and hygienic alternative, making it especially useful in post-pandemic workplace environments. Moreover, it eliminates issues related to lost or stolen RFID cards, making attendance tracking more reliable.

The system also finds applications in healthcare facilities, where accurate shift management of doctors, nurses, and other staff is critical. Hospitals and clinics operate 24/7, requiring a robust attendance system to ensure that the required personnel are present during their shifts. Since hygiene is a major concern in healthcare settings, using a contactless facial recognition system minimizes the spread of infections, offering a safer alternative to fingerprint-based biometric systems.

CHAPTER 2

CHAPTER 2
PROJECT DESCRIPTION

2.1 METHODOLOGY

The Face Recognition Attendance System using PHP follows a structured methodology to ensure effective design, development, and implementation. The methodology consists of several key phases, including requirement analysis, system design, database development, face recognition integration, system implementation, and testing & evaluation.

Face recognition/identification is a behavior that humans perform routinely and effortlessly, in our daily lives. The person's face identification process goes through four main phases.

First, the system detects the individual face. Second, the face of the person will be compared to the other faces in the system's database. Finally, after comparison, the output identifies who is the detected person.

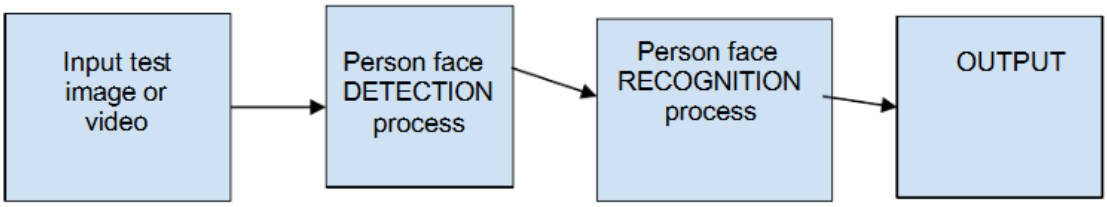


Figure 1: Face recognition process

The methodology ensures a systematic approach to designing and implementing the Face Recognition Attendance System. By following a structured workflow from requirement analysis to deployment, the system achieves high accuracy, reliability, and ease of use. This methodology ensures that the project meets its objectives of providing automated, contactless, and secure attendance tracking.

CHAPTER 3

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented.

1. A Web Camera – For Capturing Biometric Data
2. Windows OS Capability System
3. RAM 4GB and Above
4. 2.8 GHz Processor and Above
5. 4 GB Disk Space and Above

3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's and tracking the team's progress throughout the development activity.

1. PHP - For Backend Processing
2. XAMPP - Environment For Integration
3. SQL - Database for storing encrypted Data
4. Biometric Libraries and APIs
5. Visual Studio Code 64bit Arm 64

CHAPTER 4

CHAPTER 4

SYSTEM DESIGN

4.1 DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

4.2 FUTURE ENHANCEMENT

Robots with 3D vision and liveness detection are transforming facial recognition attendance systems by enhancing accuracy and security. Using 3D depth-sensing cameras, these robots detect facial features more precisely and prevent spoofing. Advanced liveness detection methods, such as blink detection and pulse analysis, ensure authenticity.

Beyond attendance tracking, AI-powered robots monitor employee behavior, detect absences, and analyze workforce productivity. They can assess fatigue and stress levels to improve workplace well-being. Drones with facial recognition will track attendance in large industrial areas, while robots will verify remote workers attendance during virtual meetings. By automating workforce management, these systems increase efficiency, reduce manual HR tasks and prevent time fraud

4.4 EXISTING SYSTYEM

The existing attendance system relies on traditional methods such as manual registers, Radio Frequency based systems, and fingerprint authentication. Manual attendance tracking is prone to

human errors and proxy attendance, while RFID systems require ID cards that can be lost or misused. While Fingerprint-based authentication are also secue but its consume more space and time expenses are effectively higher. Additionally, these systems often lack integration with real-time reporting tools and the possibility of forgery, making it difficult for organizations to track attendance efficiently. Furthermore, maintaining hardware like fingerprint scanners requires regular upkeep, adding to operational costs.

4.5 PROPOSED SYSTEM

To overcome and conquer these limitations,I have planned to introduce facial authentication. This advanced solution leverages AI-powered algorithms and high-resolution cameras to identify individuals based on unique facial features, eliminating the possibility of proxy attendance. Unlike traditional methods, it is contactless, making it more efficient. The system is can also integrated with cloud-based databases, enabling real-time attendance tracking, automated report generation, and seamless connectivity with Lead or Student tracking platforms. Moreover, it requires minimal maintenance compared to fingerprint scanners, making it a cost-effective and reliable solution for modern Society By enhancing security, accuracy, and efficiency, the proposed system significantly improves the overall tracking process.

SOFTWARE USED

4.6. PHP

PHP (Hypertext Preprocessor) is a widely used open-source scripting language primarily designed for web development. It is a server-side language, meaning it runs on the server and generates dynamic web content before sending it to the user's browser. PHP is known for its simplicity, flexibility, and efficiency, making it one of the most popular languages for building dynamic websites and web applications.

How PHP Works

1. A user requests a PHP page via a web browser.
2. The request is sent to the web server (Apache, Nginx, etc.), where PHP is installed.
3. The PHP script is executed on the server.
4. If the script interacts with a database (e.g., MySQL), PHP retrieves the required data.
5. The server processes the script and sends the final HTML output to the browser.

4.6.1 FEATURES:

1. Open-Source & Cross-Platform – PHP is free to use and runs on various platforms like Windows, Linux, and macOS.
2. Server-Side Execution – PHP scripts are executed on the server, generating HTML, which is then sent to the client's browser.
3. Database Integration – PHP supports multiple databases, including MySQL, PostgreSQL, MongoDB, and SQLite, making it ideal for data-driven applications.
4. Embedded in HTML – PHP code can be embedded within HTML, allowing developers to create dynamic web pages easily.
5. Extensive Library Support – PHP has a vast collection of built-in functions and libraries for handling files, images, encryption, and more.

4.7 SQL DATABASE

SQL (Structured Query Language) is a standard programming language used to manage and manipulate relational databases. It allows users to store, retrieve, update, and delete data efficiently. SQL is widely used in web development, data analysis, and enterprise applications, providing a structured way to handle large datasets and maintain data integrity.

Features of SQL

1. Data Querying – SQL allows users to retrieve specific data from databases using commands like SELECT, WHERE, and JOIN.

2. Data Manipulation – It enables insertion (INSERT), updating (UPDATE), and deletion (DELETE) of data records.

3. Database Management – SQL helps in creating and managing databases with commands like CREATE DATABASE, DROP DATABASE, and ALTER TABLE.

4. Data Integrity & Security – SQL enforces data consistency using constraints like PRIMARY KEY, FOREIGN KEY, and UNIQUE.

5. Scalability – SQL-based databases efficiently handle large amounts of data, making them suitable for enterprise applications.

6. Transaction Control – Supports operations like COMMIT, ROLLBACK, and SAVEPOINT to manage database transactions.

7. Standardized Language – SQL follows ANSI and ISO standards, making it compatible with multiple database management systems (DBMS).

4.8 ANALYSIS OF PROJECT

As we go through the project, we realize that there are weaknesses and strengths in the project. The developed software is qualified to identify faces correctly. However, the program can search through files without needing a database. Animated images could not be challenging for the software. This means it can detect the face of the individual even though the individual face is moving. In addition to that, the web application can detect more than one face in an image. On the other hand, the weakness of the project is that it needs two servers to run properly. It needs a server to detect faces and a server for uploading a face image to the application file, which needs more CPU and GPU resources.

SQL plays a crucial role in ensuring the smooth functioning of a Face Recognition Attendance System, particularly in data storage, retrieval, and management. The system requires fast, secure, and scalable database operations, which SQL efficiently handles. Below is a detailed analysis of SQL's impact on the system:

1. Data Storage & Management

SQL databases, particularly MySQL, provide a structured way to store and manage user information, attendance logs, and facial recognition data.

- Each user has a unique ID, linked to facial data, timestamps, and role-based access.
- Attendance records are stored in a well-organized relational structure, preventing data duplication and inconsistency.
- Efficient indexing techniques improve query performance for large datasets.

2. Performance & Efficiency

SQL databases support high-speed data retrieval, which is essential for real-time attendance marking.

- Indexes and optimized queries ensure that checking attendance history is fast.
- Joins and subqueries allow efficient fetching of attendance logs based on user IDs, dates, or departments.
- SQL ensures high concurrency, meaning multiple users can access the database without performance lag.

3. Security & Access Control

Data security is a critical aspect of an attendance system, as it involves sensitive user information. SQL provides:

- User authentication & role-based access (GRANT and REVOKE commands) to prevent unauthorized modifications.
- Encryption mechanisms for storing passwords and sensitive facial data.
- Backup & recovery features to protect data from loss due to system failures.

4. Integration with PHP & Face Recognition Module

SQL seamlessly integrates with PHP to facilitate dynamic web applications.

- The PHP-MySQL connection allows real-time data storage and retrieval for attendance records.
- The face recognition module (built using Python & OpenCV) communicates with SQL through APIs for storing and verifying facial data.

.

CHAPTER 5

CHAPTER 5

IMPLEMENTATION

5.1 ARCHITECTURE

The architecture of the application responds to a multilayered design. As developers, this allows us to modify different sections of the architecture without affecting the whole implementation, making for a more scalable design. Four layers have been designed.

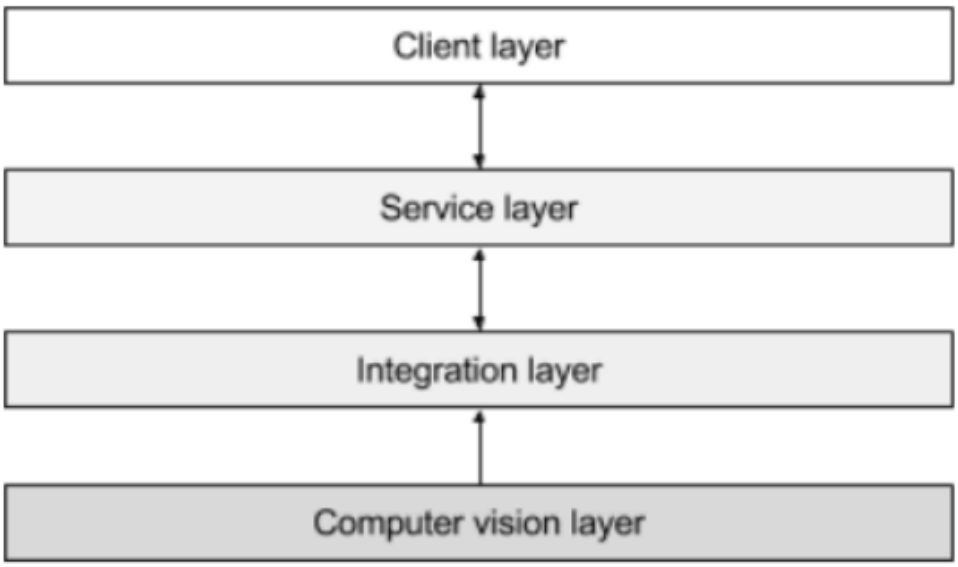


Figure 5. Representation of the multilayered architecture of the application.

Nearest to the client lies the client layer, which includes the view visible for the user and all the logic associated to it, including the communication with the server. The service layer packs the logic responsible for attending client's requests. The server has to make use of computer vision techniques in order to attend the requests. The integration layer communicates the service layer with these resources, which are contained within the computer vision layer. The problem of identity validation through artificial vision implies that the system has to be able to find faces in digital images, it has to be able to determine if two image faces belong tothe same person and it has to be able to read text from images. From now on, this problems will be referred as face detection, face verification and optical character recognition (OCR).

The architecture splits the solution to the aforementioned problems in three components:

- Detector component, as a face detection solution.
- Verifier component, as a face verification solution.
- Reader component, as an OCR solution

Additionally, two data structures have been implemented to allow the components to share information between them and with the server.

- Image, as a representation of a digital image understandable inside the application context. In order to support communication between client-server layers, Image objects can be parsed from base 64 encoded strings, and converted back to base 64.
- Face, as an extension of Image. Stores the coordinates of both eyes alongside the face image

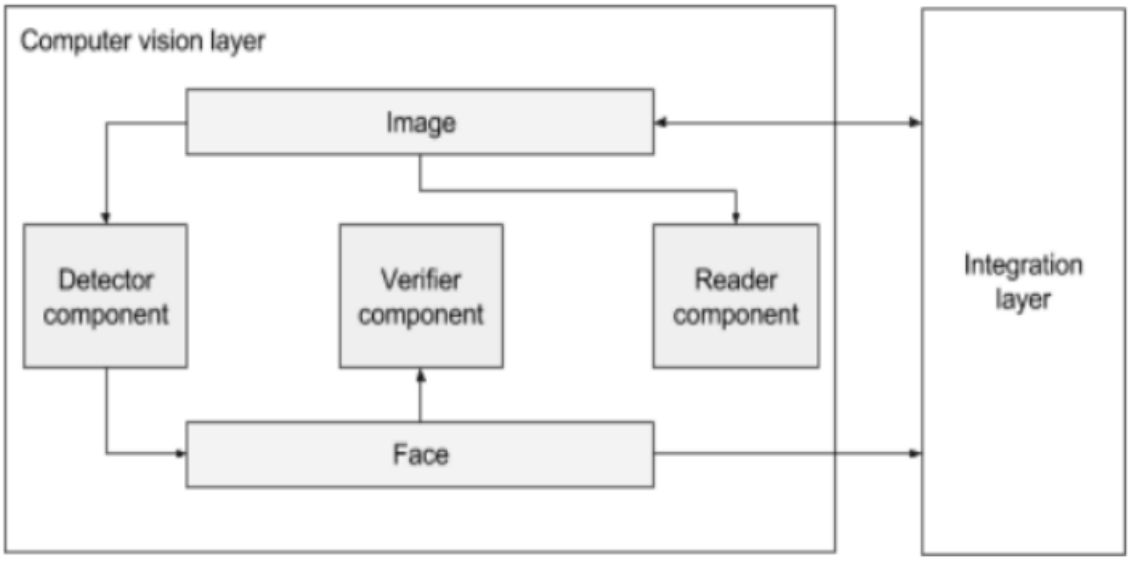


Figure 6. Representation of the internal structure of the computer vision layer.

At a code level, each of the components and data structures in the computer vision layer has been encapsulated into a class. Before delving into the implementation of the layers and components, there are some particularities about the problems the project faces and the technologies chosen to

solve them. The following chapters have been organized thematically by problem, providing a detailed view of the technologies and the implementation of the associated layers and components.

5.2 CLIENT AND SERVICE LAYER

The application has been distributed following a client-server structure. As stated in Chapter 1.4, nor the look & feel and the usability, nor the quality of the communication between client and server are being considered as objectives of the project. Taking this into account, the implementation of client and server layers poses two problems:

1. A logic for the server has to be designed to be able to attend client's requests.

2. The computer vision resources necessary for the verification process have to be accessible and usable for the aforementioned logic. That means, that the resources have to be integrated into the server.

This chapter describes how the information flows between the two layers, and the tasks performed from the server to attend the client's requests.

5.3 TECHNOLOGIES

Regarding the service layer, I have decided to use PHP as the programming language for the server. The decision stems from both this being the most extended language for server programming and my desire to improve my skills on it,. Regarding the client layer, I have decided to use HTML5 for the design, CSS for the style and JavaScript for the behavior. As the design on this layer has a lower priority

CHAPTER 6

CHAPTER 6
TESTING

6.1 SOFTWARE TEESTING SPECIFICATION

Purpose: The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope: This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

DEVELOPERS RESPONSIBILITIES OVERVIEW:-

The developer is responsible for:

• Developing the system, which meets the SRS and solving all the requirements of the system

• Demonstrating the system and installing the system at client's location after the acceptance testing is successful.

• Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.

• Conducting any user training that might be needed for using the system.

• Maintaining the system for a period of one year after installation.

FUNCTIONAL REQUIREMENTS:-

OUTPUT DESIGN:

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

• External Outputs, whose destination is outside the organization.

• Internal Outputs whose destination is with in organization and they are the User's main interface with the computer.

• Operational outputs whose use is purely with in the computer department.

• Interface outputs, which involve the user in communicating directly

OUTPUT DEFINITION:

The outputs should be defined in terms of the following points:

▪ Type of the output

▪ Content of the output

▪ Format of the output

▪ Location of the output

▪ Frequency of the output

▪ Sequence of the output

• Operational, which are computer department's communications to the system

• Interactive, which are inputs entered during a dialogue.

Keeping in view the above description of the input types and input media, it can be said that most of the inputs are of the form of internal and interactive.

As Input data is to be the directly keyed in by the user, the keyboard can be considered to be the most suitable input device.

ERROR AVOIDANCE:

At this stage care is to be taken to ensure that input data remains accurate form the stage at which it is recorded upto the stage in which the data is accepted by the system. This can be achieved only by means of careful control each time the data is handled.

DATA VALIDATION:

Procedures are designed to detect errors in data at a lower level of detail. Data validations have been included in the system in almost every area where there is a possibility for the user to commit errors. The system will not accept invalid data. Whenever an invalid data is keyed in, the system immediately prompts the user and the user has to again key in the data and the system will accept the data only if the data is correct.

Validations have been included where necessary. The system is designed to be a user friendly one. In other words the system has been designed to communicate effectively with the user. The system has been designed with pop up menus

USER INTERFACE SYSTEMS CAN BE BROADLY CLASIFIED AS:

1. User initiated interface the user is in charge, controlling the progress of the user/computer dialogue. In the computer-initiated interface, the computer selects the next stage in the interaction.

2. Computer initiated interfaces

In the computer initiated interfaces the computer guides the progress of the user/computer dialogue. Information is displayed and the user response of the computer takes action or displays further information.

COMPUTER-INITIATED INTERFACES:

The following computer – initiated interfaces were used:

1. The menu system for the user is presented with a list of alternatives and the user chooses one; of alternatives.

2. Questions – answer type dialog system where the computer asks question and takes action based on the basis of the users reply. Right from the start the system is going to be menu driven, the opening menu displays the available options.

Right from the start the system is going to be menu driven, the opening menu displays the available options. Choosing one option gives another popup menu with more options. In this way every option leads the users to data entry form where the user can key in the data.

ERROR MESSAGE DESIGN:

The design of error messages is an important part of the user interface design. As user is bound to commit some errors or other while designing a system the system should be designed to be helpful by providing the user with information regarding the error he/she has committed. This application must be able to produce output at different modules for different inputs.

PERFORMANCE REQUIREMENTS:

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

• The system should be able to interface with the existing system

• The system should be accurate

• The system should be better than the existing system

6.2 FACE POSITIONING AND ALIGNMENT

Earlier localization of facial feature points focused on two or three key points, such as locating the center of the eyeball and the center of the mouth, but later introduced more points and added mutual restraint to improve the accuracy and stability of positioning Sex. The article Active Shape Models-Their Training and Application [5] is a model of dozens of facial feature points and texture and positional relationship constraints considered together for calculation. Although ASM has more articles to improve, it is worth mentioning that the AMM model, but also another important idea is to improve the original article based on the edge of the texture model. The regression-based approach presented in the paper Boosted Regression Active Shape Models is better than the one based on the categorical apparent model. The article Face Alignment by Explicit Shape Regression is another aspect of ASM improvement and an improvement on the shape model itself. Is based on the linear combination of training samples to constrain the shape, the effect of alignment is currently seen the best The purpose of the facial feature point positioning is to further determine facial feature points (eyes, mouth center points, eyes, mouth contour points, organ contour points, etc.) on the basis of the face area detected by the face detection / tracking, s position

PHP-CPP has been decided as the technology to use in the integration, the components in the computer vision layer are expected to implemented in C++. Then, the types used in the components will not necessarily match the types available in PHP. The architecture interposes an additional layer between the service layer (the PHP server) and the computer vision layer. By using the PHP-CPP library, the layer defines a series of C++ classes that wrap the components found in the computer vision layer so that its resources can be used in PHP.

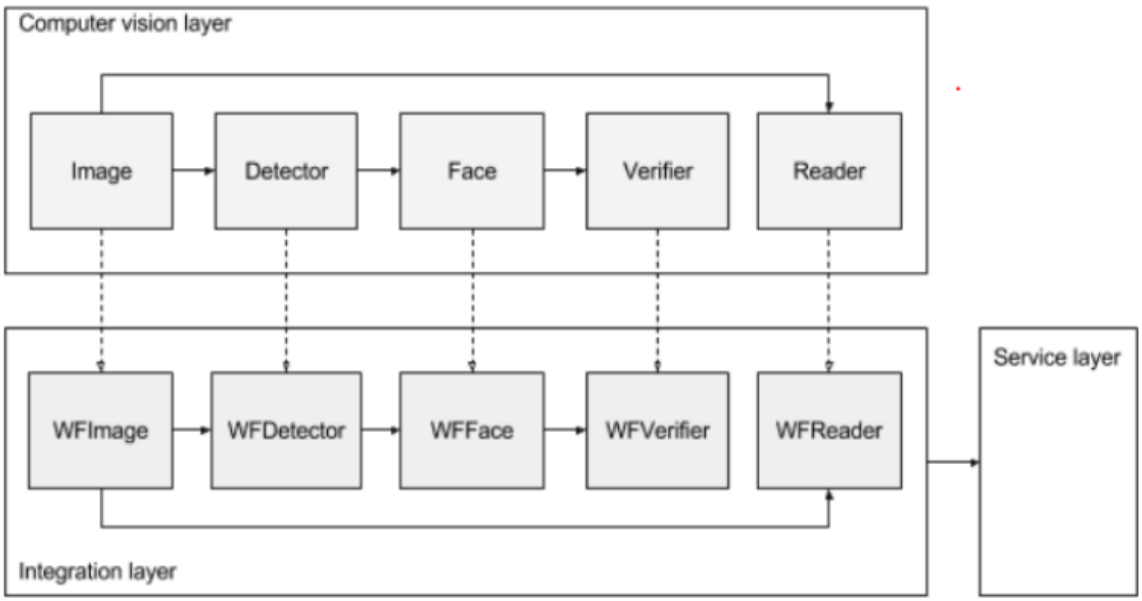


Figure 14. Structure of both the computer vision layer and the integration layer. The components in the integration layer match those found in the computer vision layer.

FACE DETECTION

The problem of identity validation through artificial vision implies that the system has to be able to find human faces in digital images. From now on, this problem will be referred as face detection. The design of the application assumes that the user has easy access to a camera device or webcam. Then, the system can expect all the face images to be frontal face images taken in relatively controlled conditions, alleviating the cost of face detection.

There are multiple libraries which can be useful when implementing the face detector. As I have previously decided the language with which extend PHP, the libraries studied limit to those available for that language. The table below summarizes what these libraries have to offer to this project.

6.3 ERROR DETECTION:

Even though every effort is make to avoid the occurrence of errors, still a small proportion of errors is always likely to occur, these types of errors can be discovered by using validations to check the input data.

The detector component is responsible for finding human faces in digital images. Given an input Image object, the detector extracts one face from it and returns it as a Face object. The face detection process is performed in three subsequent steps:

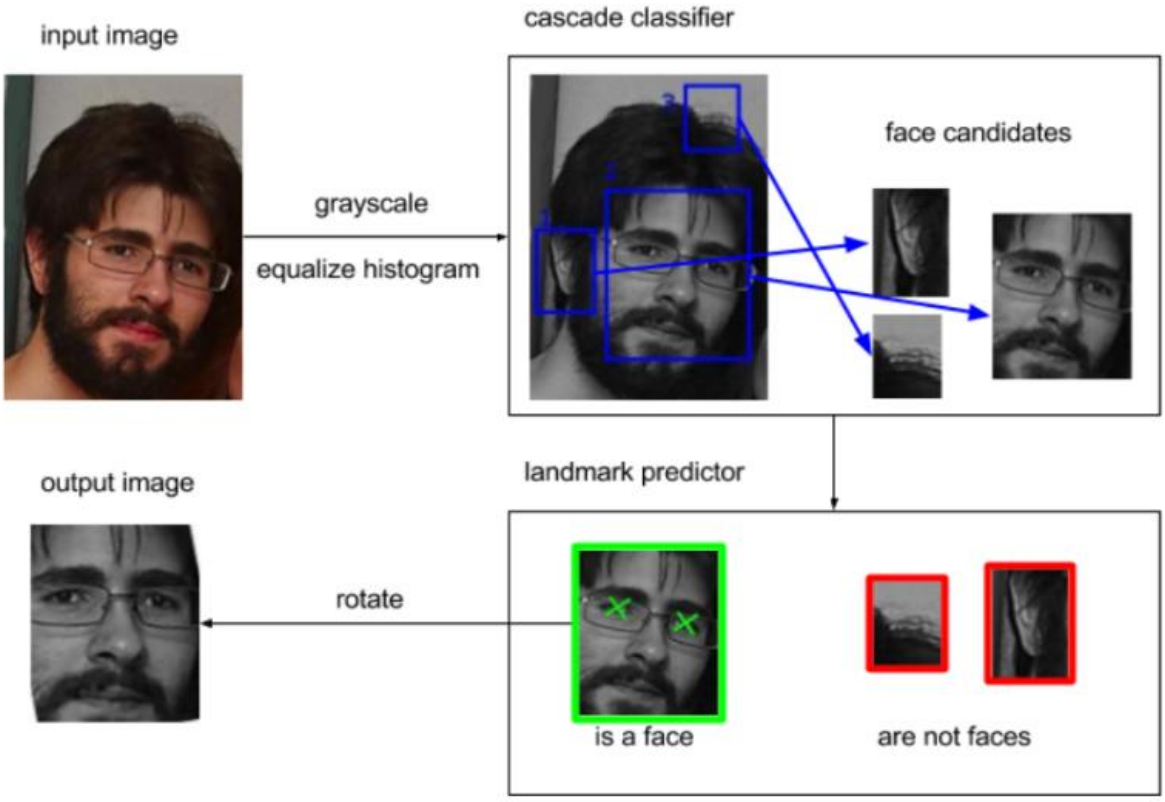
1. The detector takes an input image and searches in it those regions candidate to be faces. The candidate search is implemented with a Haar features based cascade classifier, provided by the library OpenCV. As the classifier is prone to false positives, it is expected that some candidates do not contain any face.

2. Then the detector searches on each of the candidates for facial features, using a face landmark predictor from the library dlib. The candidate is discarded if it does not contain at least a left eye and a right eye. Due to the accuracy of the predictor, it is expected for all the remaining candidates to contain a face.

3. Finally, the detector rotates the image so that the face ends up straightened taking the eyes coordinates as reference.

4. The detector takes an input image and searches in it those regions candidate to be faces. The candidate search is implemented with a Haar features based cascade classifier, provided by the library OpenCV. As the classifier is prone to false positives, it is expected that some candidates do not contain any face.

Since the classifier makes use of Haar feautures, input images have to be grayscaled. OpenCV proposes incrementing the global contrast of input images by histogram equalization, as a mean to improve the classifier's performace. Before attempting any face detection, the detector has to be loaded. Loading the detector means to provide it with necessary training data for both the cascade classifier and the face landmark predictor.



The component depends on the server layer to provide the training data sets. These are passed into the component as directory paths to the resources. The component itself takes responsibility on the training of both the classifier and the predictor. Once the classifier and the detector have been trained, the state persists for the whole component lifecycle, and therefore there is no need to reload it between detections.

By default, the server creates and loads one detector by session, being the first detector created stored in the user's session. The server can be configured to generate one detector per request. This should not hinder the performance of the application because the cost of loading the detector is minimal when compared to the cost of validating a reference or comparing two faces.

If trained with the reference image as a positive example and a set of random face images as negative examples, FaceRecognizer can be employed as a face verificator. Alternatively, face verification might be approached as an independent problem to face recognition. As the user is unknown to the application, we can depend on unattended machine learning to decide if the resemblance between two given face images is enough for them to belong to a same individual. The

dlib library provides deep learning solutions, allowing for an unattended machine learning approach.

The main problem with deep learning, though, is the training cost. The amount of images needed to train a deep neural network is usually several magnitude orders greater than the amount needed to train a classifier.

A residual network is a deep learning framework developed at Microsoft Research as a solution to the degradation problem on deep networks. This problem describes how the network's accuracy gets saturated and then degrades rapidly at increasing depths. Unexpectedly, such degradation is not caused by overfitting, and adding more layers to a suitably deep model leads to higher training error.

6.4 FACE VERIFIER

This component is responsible for deciding whether the face images contained in two given Face objects belong to the same person or not. Compared to the face detector, the verifier is more straightforward. The verifier uses a residual deep neural network to compare the images contained in two given Face objects. The network is being implemented with dlib library, and for each comparison made it returns a distance of dissimilarity between the images. To make the decision, the distance is compared with a threshold value. That way, two face images are considered to belong to a same person if the distance is inferior to the threshold. The component uses a default threshold of 0.6. Although the network is insensitive to the size of the input faces, it expects them to be 3- channeled images. In chapter 3.3 it is stated that the face detector grayscales the images when performing the detection, and hence we can expect the Face objects passed to the verifier to also contain grayscaled images. Then, the verifier component has to rescale the images back to 3 color channels before attempting the comparison.

CHAPTER 7

CHAPTER 7

CONCLUSION

7.1 SOURC CODE

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

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link href="resources/images/logo/attnlg.png" rel="icon">
    <title>Dashboard</title>
    <link rel="stylesheet" href="resources/assets/css/styles.css">
                <link href="https://cdnjs.cloudflare.com/ajax/libs/remixicon/4.2.0/remixicon.css"
rel="stylesheet">
</head>

<body>
    <?php include 'includes/topbar.php'; ?>
    <section class="main">
        <?php include 'includes/sidebar.php'; ?>
        <div class="main--content">
            <div class="overview">
                <div class="title">
```

```
    <h2 class="section--title">Overview</h2>
    <select name="date" id="date" class="dropdown">
        <option value="today">Today</option>
        <option value="lastweek">Last Week</option>
        <option value="lastmonth">Last Month</option>
        <option value="lastyear">Last Year</option>
        <option value="alltime">All Time</option>
    </select>
</div>
<div class="cards">
    <div class="card card-1">

        <div class="card--data">
            <div class="card--content">
                <h5 class="card--title">Registered Students</h5>
                <h1><?php total_rows('tblstudents') ?></h1>
            </div>
            <i class="ri-user-2-line card--icon--lg"></i>
        </div>

    </div>
    <div class="card card-1">

        <div class="card--data">
```

```
            <div class="card--content">
                <h5 class="card--title">Units</h5>
                <h1><?php total_rows("tblunit") ?></h1>
            </div>
            <i class="ri-file-text-line card--icon--lg"></i>
        </div>

    </div>

    <div class="card card-1">

        <div class="card--data">
            <div class="card--content">
                <h5 class="card--title">Registered Lectures</h5>
                <h1><?php total_rows('tbllecture') ?></h1>
            </div>
            <i class="ri-user-line card--icon--lg"></i>
        </div>

    </div>
  </div>
</div>

<div class="table-container">
```

```
<a href="manage-lecture" style="text-decoration:none;">
    <div class="title">
        <h2 class="section--title">Lectures</h2>
        <button class="add"><i class="ri-add-line"></i>Add lecture</button>
    </div>
</a>
<div class="table">
    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Email Address</th>
                <th>Phone No</th>
                <th>Faculty</th>
                <th>Date Registered</th>
                <th>Settings</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <?php
                $sql = "SELECT l.*, f.facultyName
        FROM tbllecture l
        LEFT JOIN tblfaculty f ON l.facultyCode = f.facultyCode";
```

```
                    $stmt = $pdo->query($sql);
                    $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
                    if ($result) {
                        foreach ($result as $row) {
                            echo "<tr id='rowlecture{$row["Id"]}'>";
                            echo "<td>" . $row["firstName"] . "</td>";
                            echo "<td>" . $row["emailAddress"] . "</td>";
                            echo "<td>" . $row["phoneNo"] . "</td>";
                            echo "<td>" . $row["facultyName"] . "</td>";
                            echo "<td>" . $row["dateCreated"] . "</td>";
                              echo "<td><span><i class='ri-delete-bin-line delete' data-id='{$row["Id"]}'
data-name='lecture'></i></span></td>";
                            echo "</tr>";
                        }
                    } else {
                        echo "<tr><td colspan='6'>No records found</td></tr>";
                    }
                    ?>

                </tbody>
            </table>
        </div>
    </div>
    <div class="table-container">
```

```
<a href="manage-students" style="text-decoration:none;">
    <div class="title">
        <h2 class="section--title">Students</h2>
        <button class="add"><i class="ri-add-line"></i>Add Student</button>
    </div>
</a>
<div class="table">
    <table>
        <thead>
            <tr>
                <th>Registration No</th>
                <th>Name</th>
                <th>Faculty</th>
                <th>Course</th>
                <th>Email</th>
                <th>Settings</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $sql = "SELECT * FROM tblstudents";
            $stmt = $pdo->query($sql);
            $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
            if ($result) {
```

```
                    foreach ($result as $row) {
                        echo "<tr id='rowstudents{$row["Id"]}'>";
                        echo "<td>" . $row["registrationNumber"] . "</td>";
                        echo "<td>" . $row["firstName"] . "</td>";
                        echo "<td>" . $row["faculty"] . "</td>";
                        echo "<td>" . $row["courseCode"] . "</td>";
                        echo "<td>" . $row["email"] . "</td>";
                            echo "<td><span><i class='ri-delete-bin-line delete' data-id='{$row["Id"]}' data-name='students'></i></span></td>";
                        echo "</tr>";
                    }
                } else {
                    echo "<tr><td colspan='6'>No records found</td></tr>";
                }

                ?>

                </tbody>
            </table>
        </div>

    </div>
    <div class="table-container">
        <a href="create-venue" style="text-decoration:none;">
            <div class="title">
```

```
        <h2 class="section--title">Lecture Rooms</h2>
        <button class="add"><i class="ri-add-line"></i>Add room</button>
    </div>
</a>
<div class="table">
    <table>
        <thead>
            <tr>
                <th>Class Name</th>
                <th>Faculty</th>
                <th>Current Status</th>
                <th>Capacity</th>
                <th>Classification</th>
                <th>Settings</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $sql = "SELECT * FROM tblvenue";
            $stmt = $pdo->query($sql);
            $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
            if ($result) {
                foreach ($result as $row) {
                    echo "<tr id='rowvenue{$row["Id"]}'>";
```

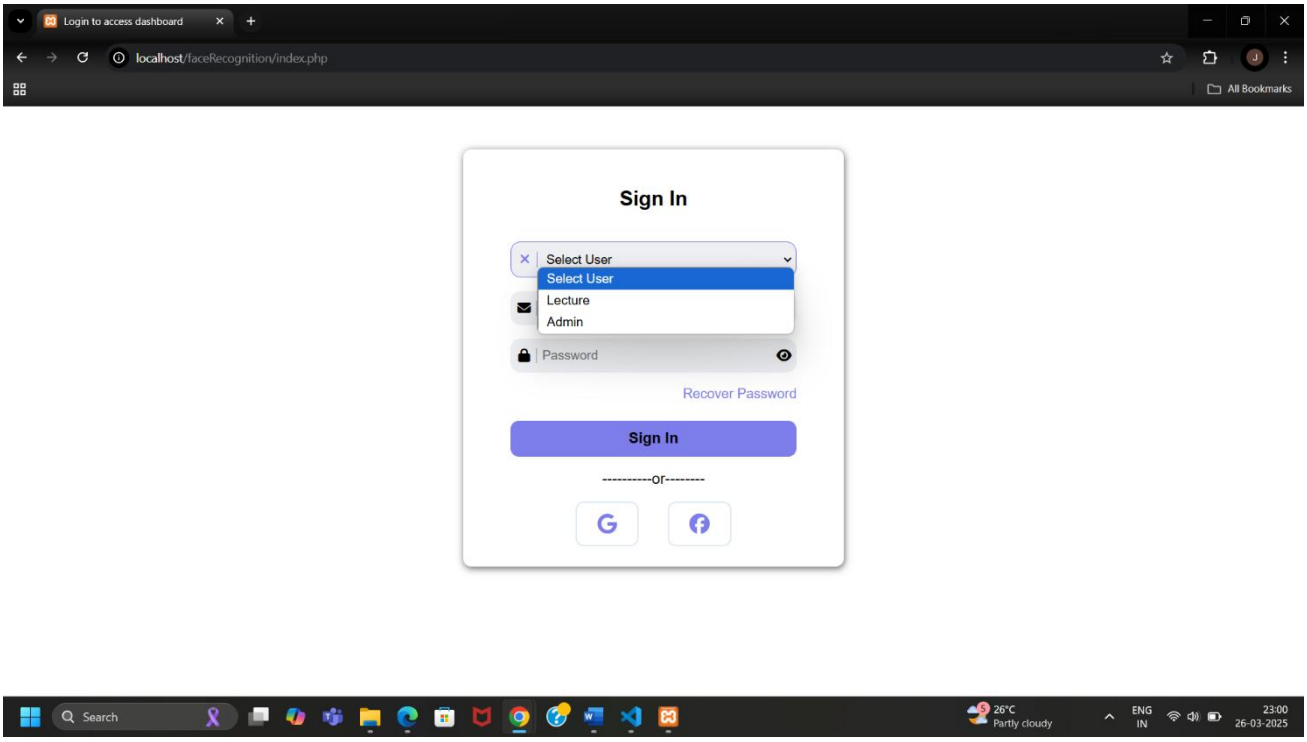
```
                    echo "<td>" . $row["className"] . "</td>";
                    echo "<td>" . $row["facultyCode"] . "</td>";
                    echo "<td>" . $row["currentStatus"] . "</td>";
                    echo "<td>" . $row["capacity"] . "</td>";
                    echo "<td>" . $row["classification"] . "</td>";
                        echo "<td><span><i class='ri-delete-bin-line delete' data-id='{$row["Id"]}' data-name='venue'></i></span></td>";
                    echo "</tr>";
                }
            } else {

                echo "<tr><td colspan='6'>No records found</td></tr>";
            } ?>
        </tbody>
    </table>
</div>

</div>
<div class="table-container">
    <a href="manage-course" style="text-decoration:none;">
        <div class="title">
            <h2 class="section--title">Courses</h2>
            <button class="add"><i class="ri-add-line"></i>Add Course</button>
        </div>
    </a>
```

```
<div class="table">
    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Faculty</th>
                <th>Total Units</th>
                <th>Total Students</th>
                <th>Date Created</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <?php
            $sql = "SELECT
        c.name AS course_name,c.Id AS Id,
        c.facultyID AS faculty,
        f.facultyName AS faculty_name,
        COUNT(u.ID) AS total_units,
        COUNT(DISTINCT s.Id) AS total_students,
        c.dateCreated AS date_created
        FROM tblcourse c
        LEFT JOIN tblunit u ON c.ID = u.courseID
        LEFT JOIN tblstudents s ON c.courseCode = s.courseCode
```

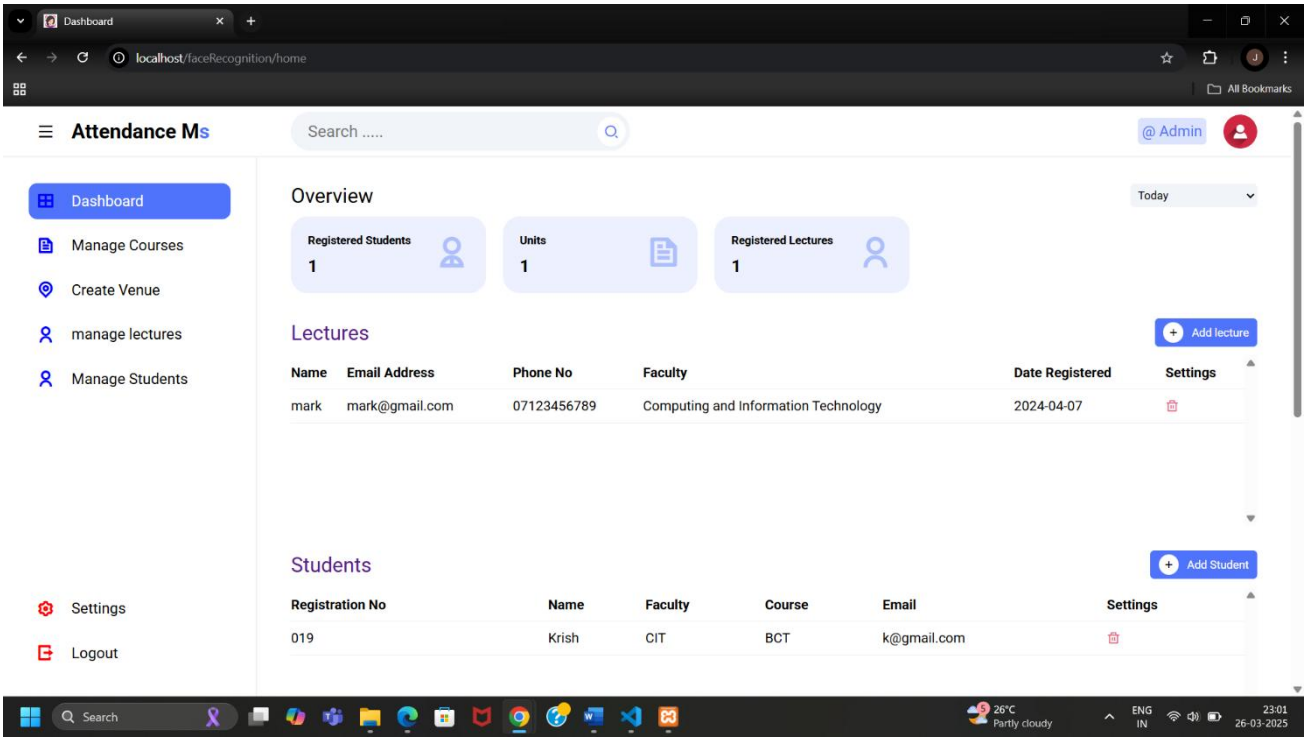
```
                LEFT JOIN tblfaculty f on c.facultyID=f.Id
GROUP BY c.ID";
$stmt = $pdo->query($sql);
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
if ($result) {
foreach ($result as $row) {
echo "<tr id='rowcourse{$row["Id"]}'>";
echo "<td>" . $row["course_name"] . "</td>";
echo "<td>" . $row["faculty_name"] . "</td>";
echo "<td>" . $row["total_units"] . "</td>";
echo "<td>" . $row["total_students"] . "</td>";
echo "<td>" . $row["date_created"] . "</td>";
echo "<td><span><i class='ri-delete-bin-line delete' data-id='{$row["Id"]}' data-name='course'></i></span></td>";
 echo "</tr>";
 } else {
 echo "<tr><td colspan='6'>No records found</td></tr>";
 }
 ?>
 </tbody>
 </table>
</div></section>
<?php js_asset(["active_link", "delete_request"]) ?>
</body>
</html>
```

7.2 SAMPLE OUTPUT

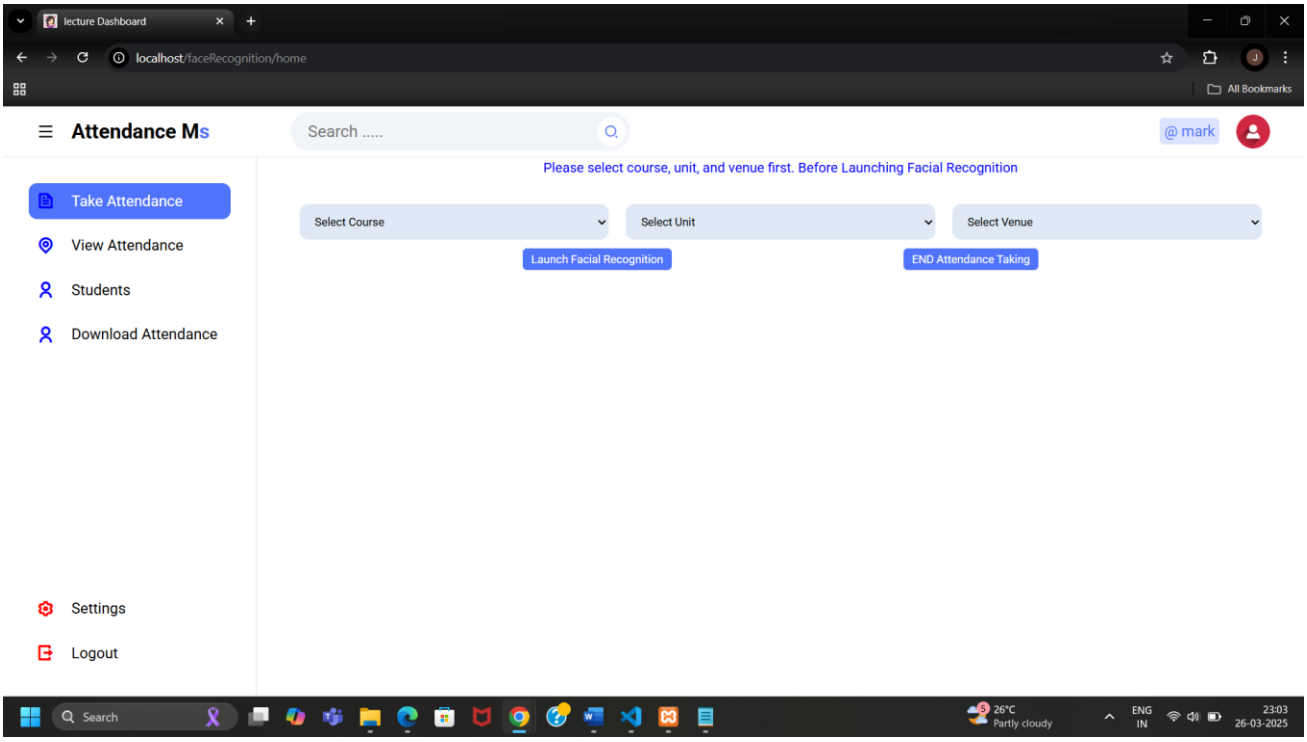
1 LOGIN PAGE



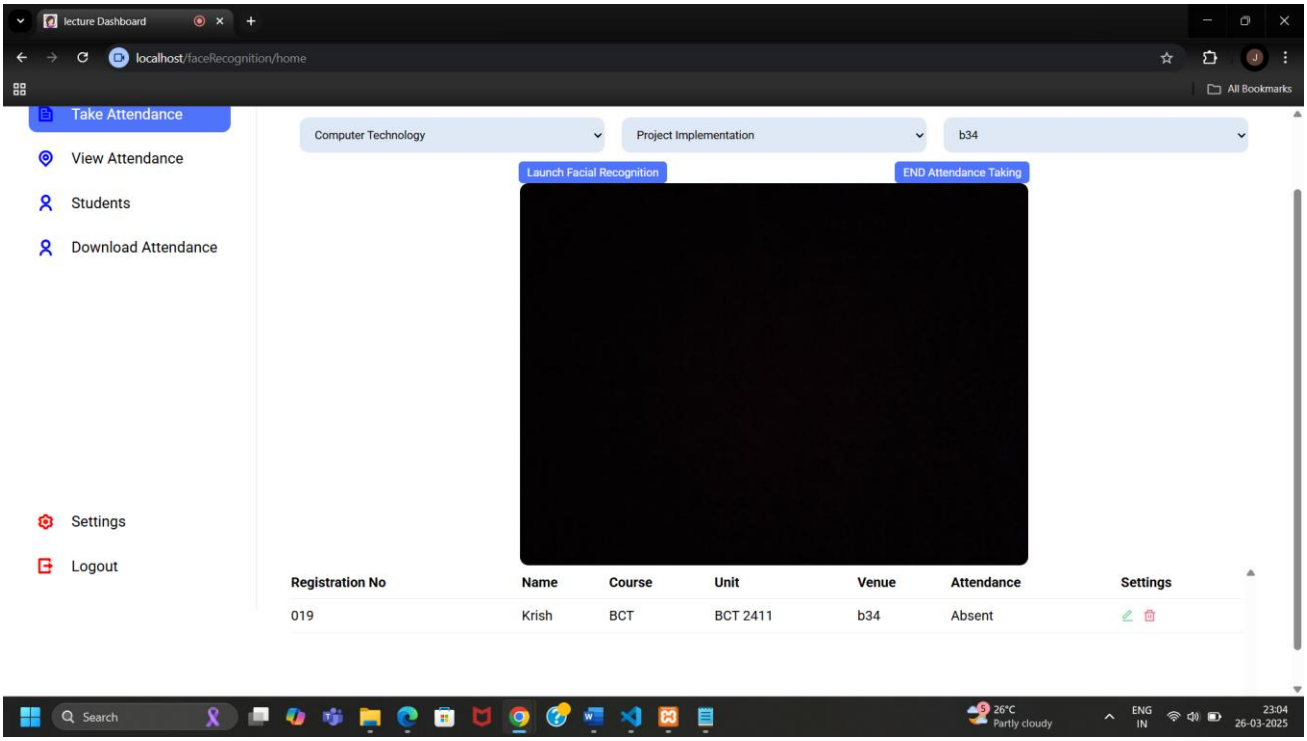
2 ADMIN PAGE



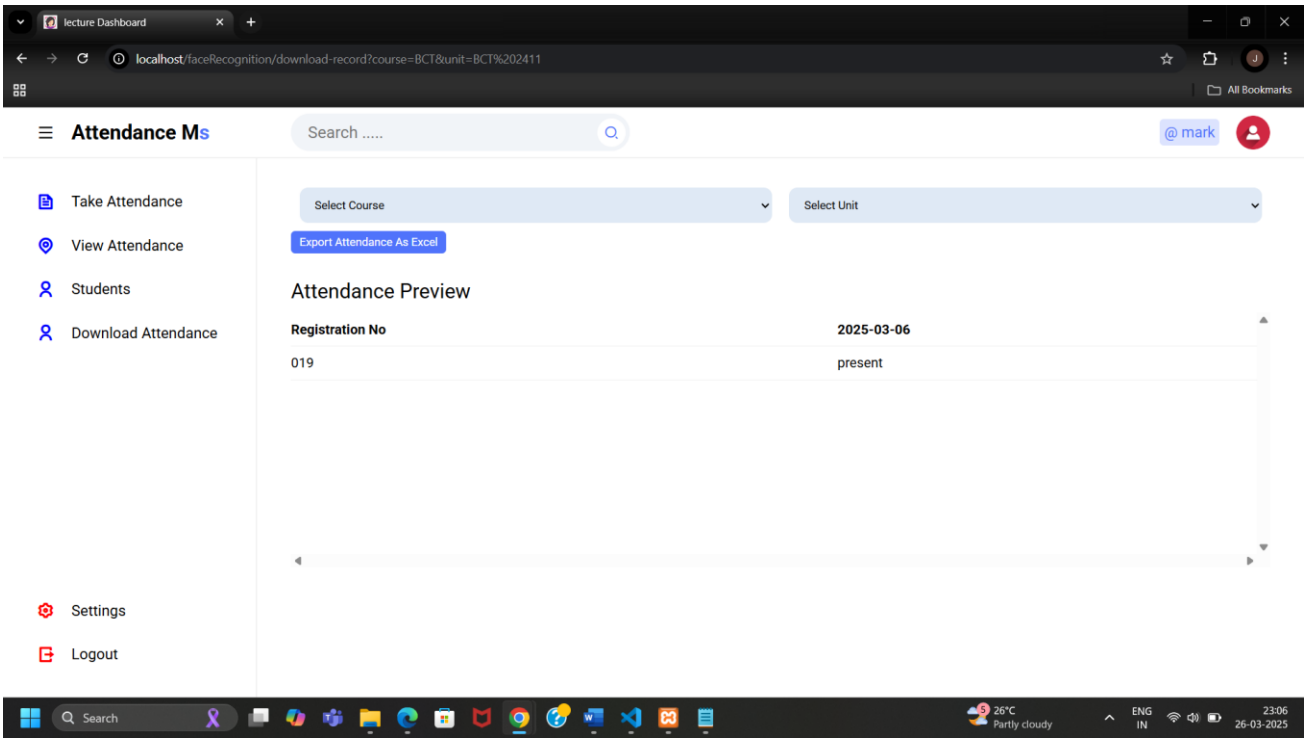
3 LECTURE PAGE



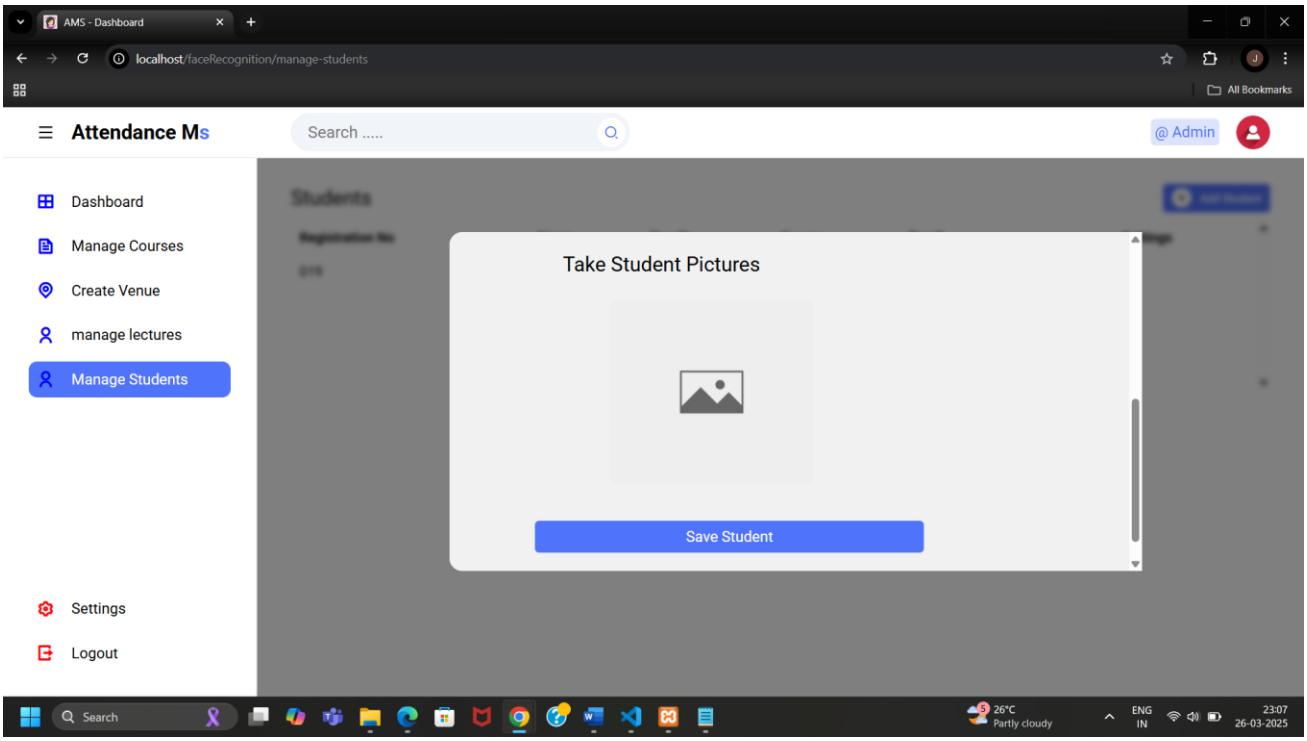
4 ATTENDACE SESSION



5 ATTENDANCE PREVIEW



6 STUDENT REGISTRATION



7.3 CONCLUSION

In Conclusions, Face recognition systems are part of facial image processing applications and their significance as a research area are increasing recently. Implementations of system are crime prevention, video surveillance, person verification, and similar security activities. The face recognition system implementation can be part of Universities. Face Recognition Based Attendance System has been envisioned for the purpose of reducing the errors that occur in the traditional (manual) attendance taking system. The aim is to automate and make a system that is useful to the organization such as an institute. The efficient and accurate method of attendance in the office environment that can replace the old manual methods. This method is secure enough, reliable and available for use. Proposed algorithm is capable of detect multiple faces, and performance of system has acceptable good results.