FACE RECOGNITION ATTENDANCE SYSTEM

Submitted by

VIGNESH S

(Register No: 713521CA854)

PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment for the award of the degree
of

MASTER OF COMPUTER APPLICATIONS

SNS COLLEGE OF TECHNOLOGY, COIMBATORE–641 035
(AN AUTONOMOUS INSTITUTION)

DEPARTMENT OF COMPUTER APPLICATIONS

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report entitled **“FACE RECOGNITION ATTENDANCE SYSTEM”** is the bonafide work of **Mr. VIGNESH S**, who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any otherproject report or dissertation on the basis of which a degree or award was conferred on an earlieroccasion on this or any other candidate.

HEAD OF THE DEPARTMENT

PROJECT GUIDE

Submitted for the Viva – Voice examination held at SNS COLLEGE OF TECHNOLOGY, held on ____________________

Internal Examiner

External Examiner

DECLARATION

I **VIGNESH S** declare that the project work entitled **“FACE RECOGNITION ATTENDANCE SYSTEM”** submitted to **SNS COLLEGE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTION),** Coimbatore of the requirements for the award of Degree of **Master of Computer Application** is a record of original work done by me under the guidance of **DR.NANDHINI.N MCA, M.Phil. Ph.D.,**Head of the Department in Computer Applications (PG), **SNS COLLEGE OF TECHNOLOGY(AN AUTONOMOUS INSTITUTION),** Coimbatore.

Place: Coimbatore

Date:

Signature of the candidate

VIGNESH S

ACKNOWLEDGEMENT

First of all I solicit my humble thanks to god almighty for being with me, guiding me theright way throughout my project work.

I express my heart-felt gratitude to the management of SNS College of technology, Coimbatore, **Dr.S.N.SUBRAMANIAN**, Chairman, **Dr.S.RAJALAKSHMI**, Correspondent, Technical Director, **Dr.S.NALIN VIMAL KUMAR** and **Dr.V.S.VELUSWAMY**, Founder Trustee for providing me with all sorts of support in completion of this project.

I extend my sincere thanks to **Dr.S.CHENTHUR PANDIAN,** our respected Principal, SNS College of Technology, for the facilities offered to complete this project.

I extend my sincere thanks to **Dr.S.SUNDARARAJAN,** Professor and Head, Department Master of Computer Applications, SNS College of Technology, which paved the way for the successful completion of my project. His instruction and presence has guided me a lot for completing this project.

I extend cheerful thanks to my internal guide **DR.NANDHINI.N MCA, M.Phil. Ph.D.,** for her support and guidance throughout this project.

I am also grateful to all other faculty members of the Department of Computer Application (PG), SNS COLLEGE OF TECHNOLOGY (AN AUTONOMOUS INSTITUTION), Coimbatore for being co-operative and friendly throughout the course.

ABSTRACT

The face recognition attendance system is an innovative solution that allows for a quick and easy way to take attendance using OpenCV, LBPHFaceRecognizer and Haar Cascades. With this system, students can simply stand in front of a camera and have their faces scanned to record their attendance, and also they can view their attendance. Using this system teacher’s may scan the student face to record the hour wise attendance and generate the day wise, month wise report. This system scan the features of student face and match with already scanned face features from the dataset to ensure the attendance. This technology can significantly reduce the time and effort required for traditional attendance-taking methods, and also providing a more secure and efficient way of tracking attendance records. This system also eliminates the chances of fake attendance because of the face being used as a biometric for authentication. Hence, this system can be implemented in a field where attendance plays an important role. The system is based on image comparison of Haar Cascades on the basis of the encoded values of the face from the image from database with the image recorded by LBPHFaceRecognizer the system in runtime.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Facial recognition attendance systems are innovative solutions that use advanced facial recognition technology to automate the attendance-taking process. Traditional attendance methods can be time-consuming, error-prone, and susceptible to fraud. Facial recognition attendance systems aim to address these issues by providing a quick, efficient, and secure way of tracking attendance records. This technology to analyse facial features and match them to a database of known faces. When a user stands in front of a camera,

This system captures their face and compares it to the database to verify their identity. If the system recognizes the user's face, their attendance is automatically recorded. This eliminates the need for physical attendance sheets or manual data entry, streamlining the attendance-taking process and saving time for both students/employees and administrators. Overall, facial recognition attendance systems offer a modern and efficient way to manage attendance records, improving productivity and reducing errors.

Moreover, face recognition attendance systems enhance security by preventing buddy punching (when someone clocks in or out on behalf of another person) and unauthorized access. The technology is difficult to spoof, as it can detect and differentiate between real faces and fake representations like masks or photographs. This ensures greater integrity and accountability in attendance tracking.

However, it is essential to consider the potential drawbacks and challenges associated with face recognition attendance systems, including accuracy limitations, privacy concerns, bias, user acceptance, cost, system vulnerabilities, and legal considerations. Addressing these issues appropriately is crucial to ensure the ethical and effective implementation of such systems.

Overall, face recognition attendance systems offer a modern and efficient approach to attendance management, leveraging cutting-edge technology to improve accuracy, streamline processes, and enhance security in various organizational settings.

1.1 COMPANY PROFILE

Neelavath Software Solutions in Coimbatore provide web designs in coimbatore at efficient, optimized and easy to read format across a variety of devices and web browsers.

Our authentic Website Development Service will be designed more adaptive to the media that resides them.

Neelavath take immense proud in stating that, we render excellent User Interface andUser Experience(UI/UX) Designs for various business domains and across all various verticals in Coimbatore, Tamil Nadu, India.

Building an E-commerce website is not easy but we Neelavath Software Solutions havebeen executing E-commerce web development in Coimbatore with great success for clients globally and we can do the same for you as well.

We are one of the top brands for WordPress development services in Coimbatore and have completed several high performance projects in the niche.

We have considerable experience developing small blogging sites to large complex WP-based websites. Logo is something that reciprocates your identity.

We offer authentic Logo designs in Coimbatore, which acts as an attraction. We willwork together to create a cohesive logo that describes your brand.

Website : http://neelavathsoftware.in.com/
Industries : Software Development Company
Size : 11-50 employees
Headquarters : Coimbatore, TAMILNADU
Type : Self-Owned
Founded : 2011
CEO: Anjali

CHAPTER 2

SYSTEM ANALYSIS

2.1 EXISTING SYSTEM:

The facial recognition attendance system uses a camera to capture images of students' faces and then uses facial recognition technology to match the images to the student database. The system has been performing well since its implementation. The system is user-friendly and easy to use. students simply need to stand in front of the camera for a few seconds for the system to capture their face. can easily track attendance and generate reports. The facial recognition attendance system is secure, with user authentication required to access the system. The system is also compliant with data privacy regulations, and student data is encrypted and stored securely. Time Clock Plus is a comprehensive attendance management system that includes face recognition capabilities. It allows employees to clock in and out using facial recognition technology, capturing real-time attendance data. The system integrates with payroll and HR systems for efficient attendance tracking.

2.1.1 DRAWBACKS:

Accuracy and Reliability: Face recognition technology is generally reliable, but it is not flawless. Factors such as poor lighting conditions, facial hair, glasses, makeup, or facial expressions can impact the accuracy of the system. False negatives or false positives can occur, leading to errors in attendance records.

Cost and Infrastructure: Implementing a face recognition attendance system requires investment in suitable hardware, such as high-resolution cameras, processing units, and storage facilities. The initial setup costs, maintenance, and infrastructure requirements can be substantial, especially for large-scale deployments.

2.2 PROPOSED SYSTEM:

The proposed facial recognition attendance system offers several benefits over Time Clock Plus, including improved accuracy, time-saving, real-time tracking, and cost savings. Implementation of the proposed system would involve several steps, including hardware procurement, installation and database creation.

2.2.1 ADVANTAGES:

1. Improved accuracy: The facial recognition technology is highly accurate, reducing the number of errors associated with manual attendance tracking.
2. Time saving: The proposed system would save time for employees and HR personnel by eliminating the need for manual attendance tracking.
3. Cost saving: The proposed system would reduce costs associated with manual attendance tracking, resulting in cost savings for the department.

2.3 FEASIBILITY STUDY

The feasibility study of the system is an important criterion for software development. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Feasibility and risk analysis are related with each other. If project risk is great, the feasibility of producing quality software is suspected.

During product engineering, however, we concentrate our attention on four primary areas of interest Software is said to be successfully completed and utilized, only if it is feasible in all ways. The following three areas were taken in to consideration while making feasibility study of the proposed system.

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

2.3.1 Technical Feasibility

Technical Feasibility strategy is built computer system hardware, software, etc. and to some extent how it can support the proposed addition. This involves financial considerations to accommodate technical enhancements. Technical support plays a vital role for the success of the project.

Technical Feasibility is mainly concerned with the study of function, performance, and constraints of the proposed project that may affect the ability to achieve the system performance. By conducting an efficient technical feasibility we can to ensure that the project works to solve the existing problem area. Since the project is designed with HTML, Python as front end and SQLite as back end.

Since both the technologies one open source, no need to make an environment in software purchase and license issues are totally eliminated, So the technical requirements ofthe proposed system can be fulfilled with little efforts that make no risks and leads to proceed.

2.3.2 Economic Feasibility

The project developed in the reasonable cost and less maintenance would reduce thetotal cost of the project. Because of this less investment the Economic feasibility can be achieved. Economic feasibility is concerned with comparing the development cost with the income/benefit derived from the developed system. It is also concerned with the cost incurred in the implementation of the software.

Since this project is developed using ReactJS and MongoDB which is more commonly available and even the cost involved in the installation process is also not matter.Similarly it is easy for the persons to operate the software since almost all the people are aware of computer system.

Even if we want to train the persons in these area the cost involved in training is also very less. Hence this project has good economic feasibility.

2.3.3 Operational Feasibility

The user interface is designed with user constraints, many risks/error occurredduring the system handling is totally avoided. Window has user-friendly interface, which makes all the operations easy to use and no extra training is needed in this regard. Since theuser will be dealing only with the client-side, the website used lot of plug-ins for user selection of option for images and thus lot of time is saved. The project is developed by the way of considering the end user. Because the operations of the website is carried out through the end users.

The development or the proposed system moves into the next stage, because of result of feasibility encourages to proceeding.

CHAPTER 3

SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application isthe physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. AnHCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects ofhardware requirements.

PROCESSOR	:	Intel Core i5-8250u CPU @1.60GHz 1.80Ghz
RAM	:	4 GB
HARD DISK	:	1 TB

3.2 SOFTWARE REQUIREMENTS

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package andneed to be installed separately before the software is installed.

OPERATING SYSTEM	:	Windows 10
FRONT END	:	HTML, CSS using Python
BACK END	:	SQLite

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 FRONT END:

4.1.1 HTML:

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents froma 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 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 providesa 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.

HTML can have embedded programs written in a scripting language such as JavaScript which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of the content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSSstandards, has encouraged the use of CSS over explicit presentational HTML since 1997.

4.1.2 CASCADING STYLE SHEET(CSS):

CSS Stands for “Cascading Style Sheet.” Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page’s HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page’s HTML, commonly used styles need to be defined only once in a CSS document.. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS givesWeb developers more exact control over how Web pages will look than HTML does. This is whymost Web pages today incorporate cascading style sheets.

4.1.3 PYTHON:

Python is a general purpose, dynamic, high level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. Python supports multi programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

4.1.4 PYTHON DJANGO:

Django Tutorial provides basic and advanced concepts of Django. Our Django Tutorial is designed for beginners and professionals both. Django is a Web Application Framework which is used to develop web applications. Our such as introduction, features, installation, environment setup, admin interface, cookie, form validation, Model, Template Engine, Migration, MVT etc. All the topics are explained in detail so that reader can get enough knowledge of Django. Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement. This framework uses a famous tag line: The web framework for perfectionists with deadlines

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

4.2 BACK-END:

4.2.1 SQLite:

SQLite is embedded relational database management system. It is self-contained, serverless, zero configuration and transactional SQL database engine.

SQLite is free to use for any purpose commercial or private. In other words, “SQLite is an open source, zero-configuration, self-contained, stand alone, transaction relational database engine designed to be embedded into an application”.

SQLite is different from other SQL databases because unlike most other SQL databases, SQLite does not have a separate server process. It reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

CHAPTER 5
PROJECT DESCRIPTION

5.1 MODULE DESCRIPTION:

The project entitled facial recognition attendance system is to design a web based application. This application facilities to track and manage student attendance accurately and efficiently. In this project there are two main modules. They are.

- Admin module
- Student module

ADMIN MODULE:

The admin module performs operations such as manage student details, Student attendance details, take attendance, send message for absentees, send message for condonation students and manage attendance record.

Manage Student details

This manages the overall student details in the system and it can be viewed and the details can also be deleted

Student Attendance details

The student attendance details such as entry-time, exit-time and break hour can be viewed to whether the student is present or not on the present date.

Take Attendance

The take attendance in the system is used to take facial attendance of the student through camera and the attendance will be updated, it works by comparing the image with already trained image of the student.

Send Mail

The send mail module is used to send mail to the student those who are absent on that day, and this module consists send mail for Re-appear to the student those who have attendance percentage below 60%.

Manage Attendance record

This manages the overall attendance record in the system and it can be viewed in the graphical representation.

STUDENT MODULE:

The admin module performs operations such as add face for attendance, train face dataset, view attendance percentage, view daily attendance, view weekly attendance, view monthly attendance.

Add Face for Attendance

The add face for attendance is used to store set of photos of our own in the folder to match the student face at the time of attendance.

Train face dataset

The train face dataset is used to train the images of our own to give the accurate face recognition at the time of attendance

View Attendance Percentage

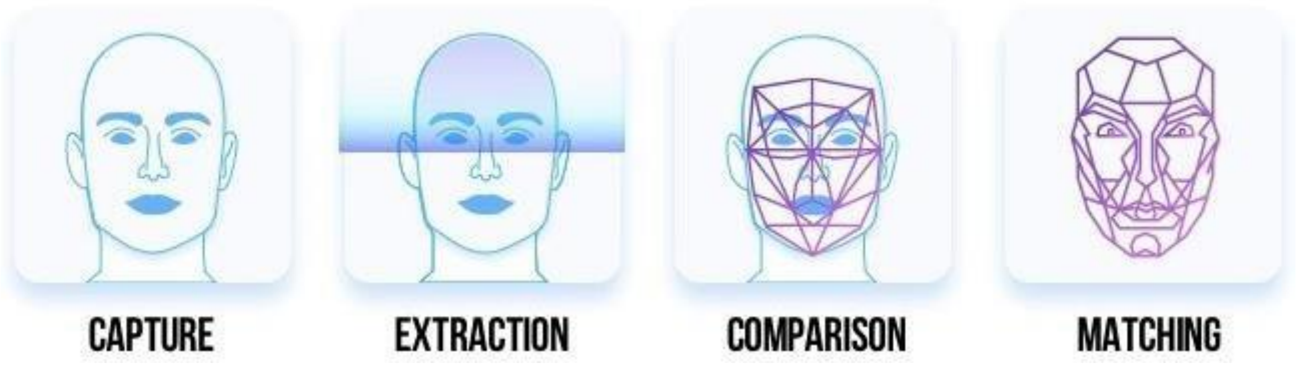
The view attendance percentage is used to view the attendance percentage of the student to till date.

View Attendance

The view attendance is used to view on day wise, week wise and monthly attendance in the graphical representation, it shows the attendance in the bar chart and pie chart.

5.2 ALGORITHM DESCRIPTION

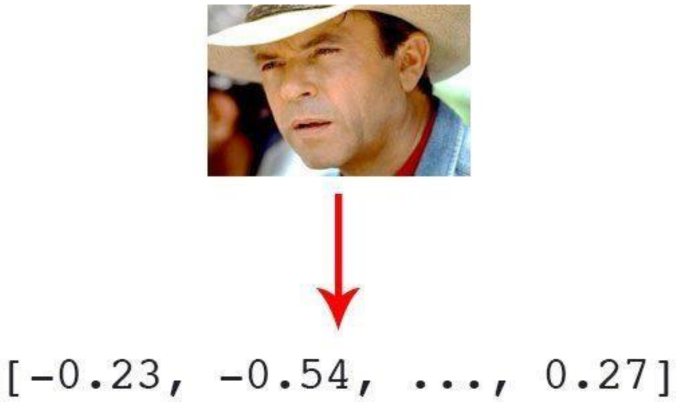
Working of Facial Recognition



1. **Concept of feature vector**: Every Machine Learning algorithm takes a dataset as input and learns fromthis data. The algorithm goes through the data and identifies patterns in the data. The challenging part is to convert a particular face into numbers – Machine Learning algorithms only understand numbers.
2. This numerical representation of a "face" (or an element in the training set) is termed as a feature vector.A feature vector comprises of various numbers in a specific order.
3. You can take various attributes to define a face like:
   - Height/width of face (cm)
   - Color of face (R,G,B)
   - Height/width of parts of face like nose & lips (cm)
   - We can consider the ratios as feature vector after rescaling
4. A feature vector can be created by organising these attributes to into a table, say, for a certain set ofvalues of attributes your table may look like this:

Height of face (cm)	**Width of face (cm)**	**Average color of face (R, G, B)**	**Width of lips (cm)**	**Height of nose(cm)**
23.1	15.8	(255, 224, 189)	5.2	4.4

image now becomes a vector that could be represented as [23.1, 15.8, 255, 224, 189, 5.2, 4.4]. Now can add a number of other features like hair color & spectacles. Keep in mind that a simple model gives the best result. Adding a greater number of features may not give accurate results (See overfitting and underfitting).



Machine learning helps you with two main things:

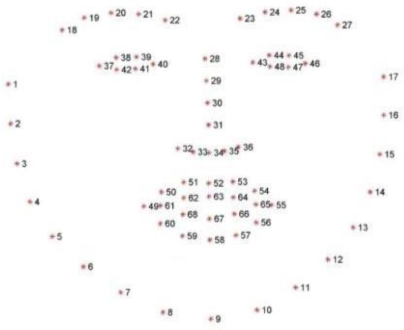
- **Deriving the feature vector**: As it is a difficult process to involve all features by name, we convert it to feature vector. This is then used by the algorithm. A Machine Learning algorithm can intelligently label out many of such features.
- **Matching algorithms**: Once the feature vectors have been obtained, a Machine Learning algorithm needs to match a new image with the set of feature vectors present in the corpus.

DCNN Basic Architecture

There are two main parts to a CNN architecture

A **convolution tool** that separates and identifies the various features of the image for analysis in a process called as Feature Extraction

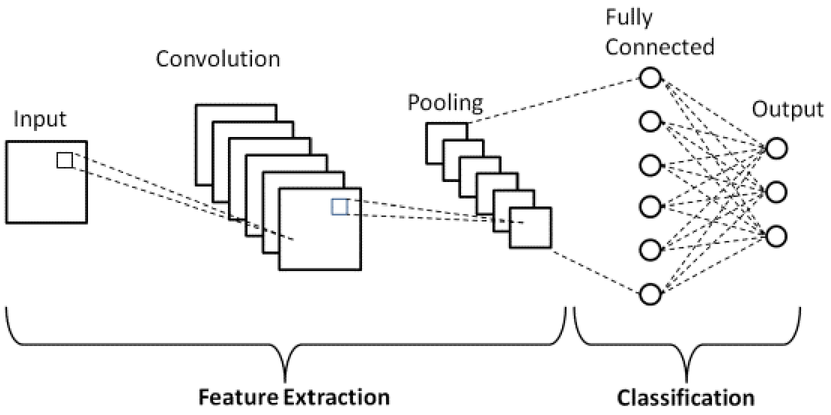
A **fully connected layer** that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.



Face Landmark Points

There are mostly two steps to detect face landmarks in an image which are given below:

- **Face detection:** Face detection is the first methods which locate a human face and return a value in x,y,w,h which is a rectangle.
- **Face landmark:** After getting the location of a face in an image, then we have to through points inside of that rectangle.



LBPHFaceRecognizer:

The Local Binary Pattern Histogram (LBPH) algorithm is a face recognition algorithm based on a local binary operator, designed to recognize both the side and front face of a human. However, the recognition rate of the LBPH algorithm is limited, if the conditions, such as in the expression diversification, disorientation, and a change in the lighting performance manifest.

Working of the LBPH algorithm

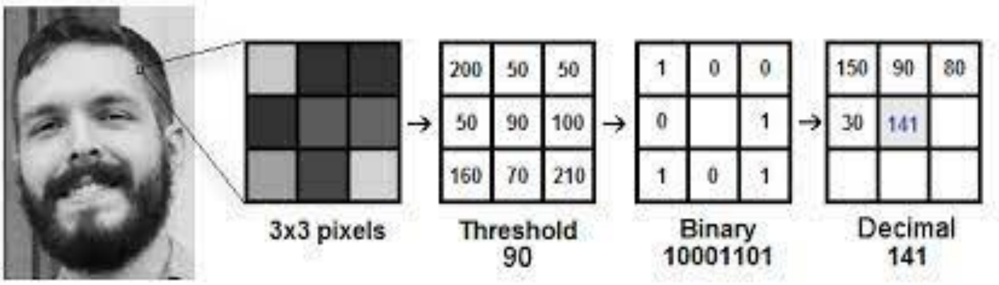
The LBPH algorithm typically makes use of 4 parameters:

- ***Radius:*** The distance of the circular local binary pattern from the center pixel to its circumference and usually takes a value of 1.
- ***Neighbors:*** The number of data points within a circular local binary pattern. Usually, the value of 8.
- ***Grid X:*** The number of cells in the horizontal plane, is usually a value of 8.
- ***Grid Y:*** The number of cells in the vertical plane, is usually a value of 8.

Two modes of operation of face recognition

The face recognition algorithm generally operates in one of two modes:

1. *Authentication of a facial image:* This mode does facial recognition by a 1x1 comparison. The comparison is done between an input image and a specific image within the database. In many cases, this is the face that requires authentication at the time of this mode of facial recognition.
2. *Face recognition:* in this mode, it is a 1xN, a comparison of the input face image with all the pictures that have been saved in the database to output the images of the user which conforms to the input face image.



200	50	50
50	90	100
160	70	210

1	0	0
0		1
1	0	1

150	90	80
30	141	

Haar Cascades:

Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Viola and Michael Jones in their paper Rapid Object Detection using a Boosted Cascade of Simple Features .Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.

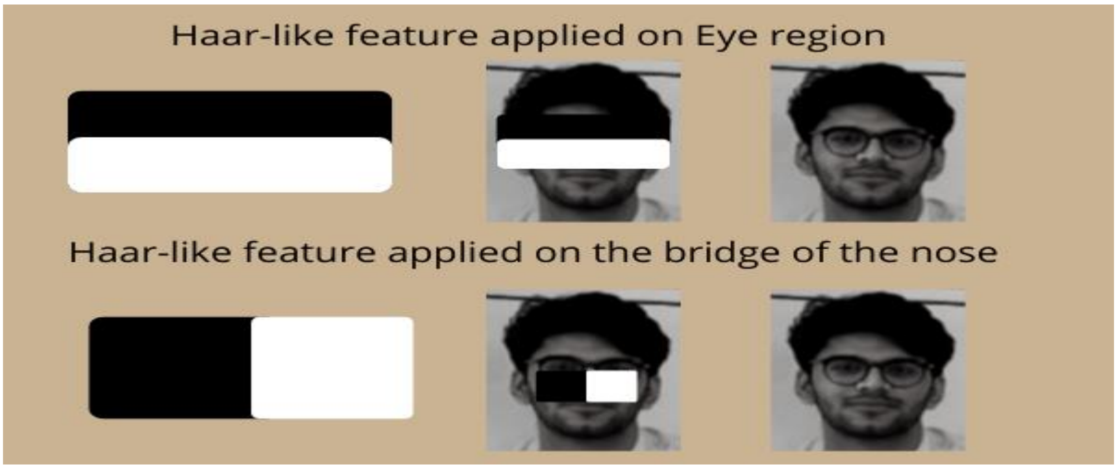
Positive images – These images contain the images which we want our classifier to identify.
Negative Images – Images of everything else, which do not contain the object we want to detect.

Haar-feature selection: A Haar-like feature consists of dark regions and light regions. It produces a single value by taking the difference of the sum of the intensities of the dark regions and the sum of the intensities of light regions. It is done to extract useful elements necessary for identifying an object. The features proposed by viola and jones are:

Creation of Integral Images: A given pixel in the integral image is the sum of all the pixels on the left and all the pixels above it. Since the process of extracting Haar-like features involves calculating the difference of dark and light rectangular regions, the introduction of Integral Images reduces the time needed to complete this task significantly.

AdaBoost Training: This algorithm selects the best features from all features. It combines multiple "weak classifiers" (best features) into one "strong classifier". The generated "strong classifier" is basically the linear combination of all "weak classifiers".

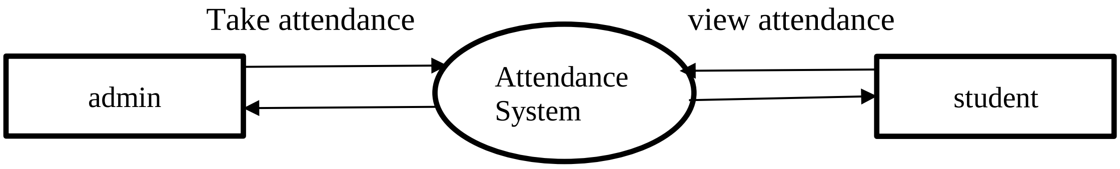
Cascade Classifier: It is a method for combining increasingly more complex classifiers like AdaBoost in a cascade which allows negative input (non-face) to be quickly discarded while spending more computation on promising or positive face-like regions. It significantly reduces the computation time and makes the process more efficient.


Haar-like feature applied on Eye region
Haar-like feature applied on the bridge of the nose

5.3 DATA FLOW:

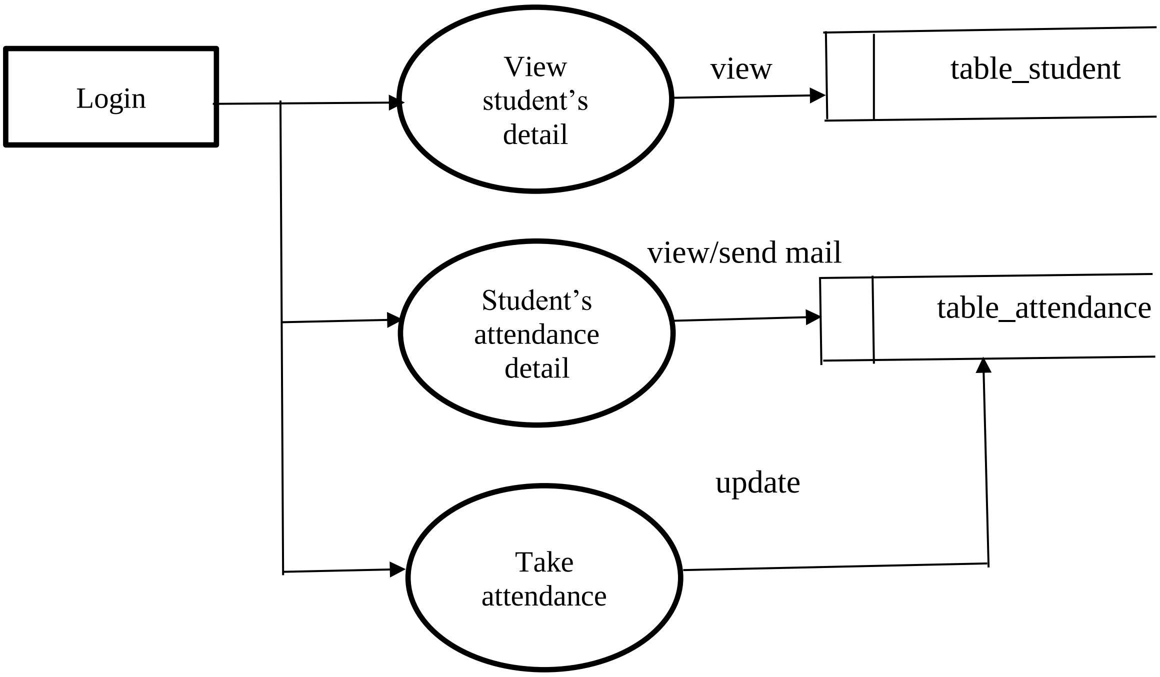
DATA FLOW DIAGRAM:

Level 0:



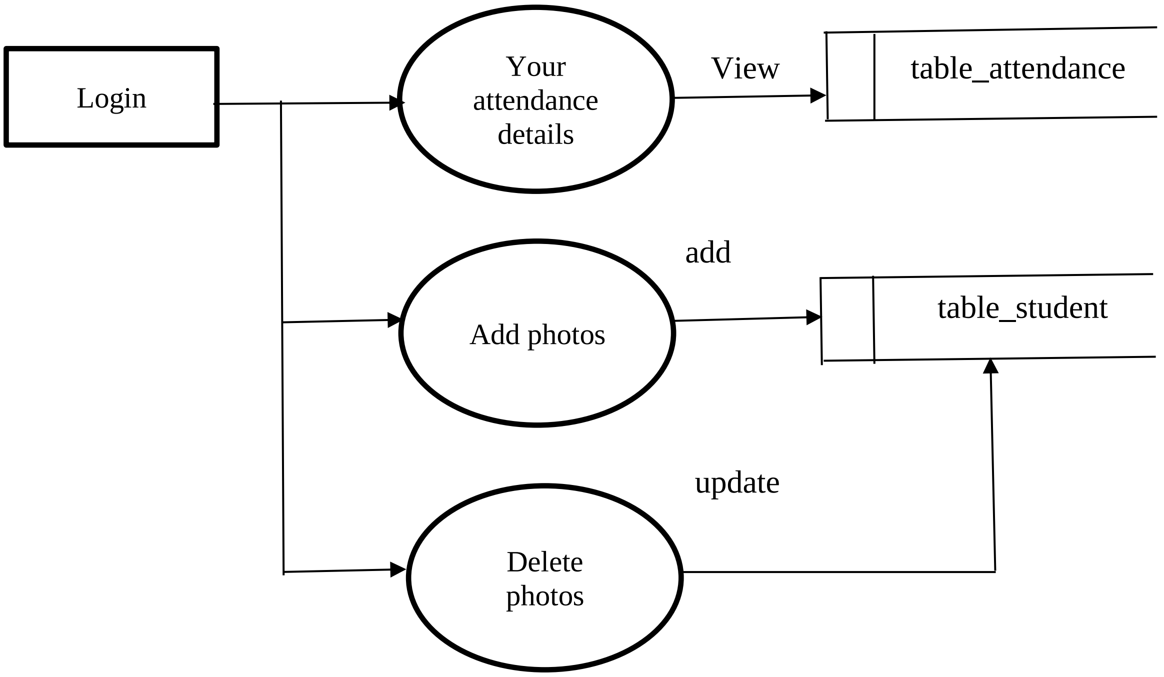
Admin:

Level 1:

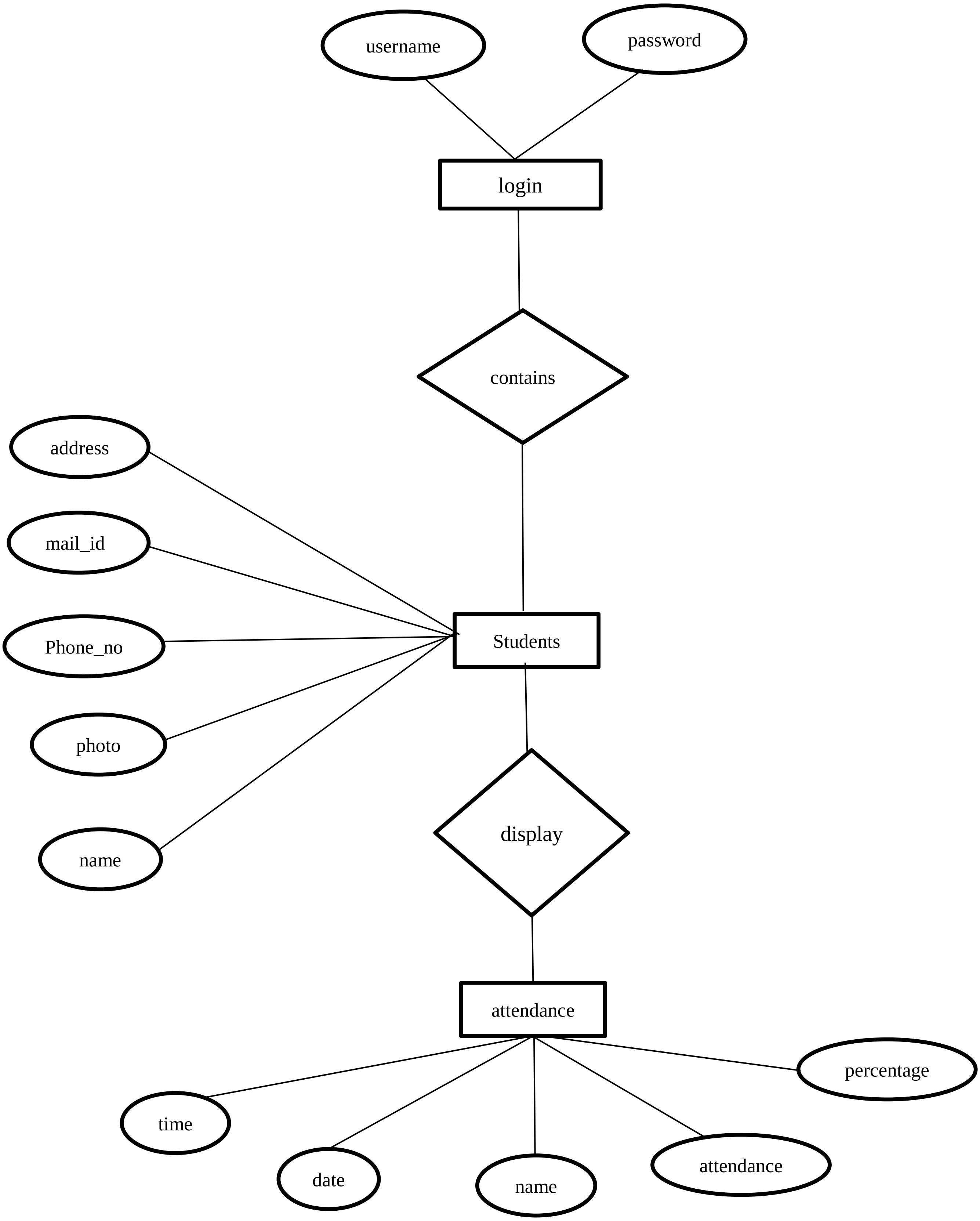


Student:

Level 1:



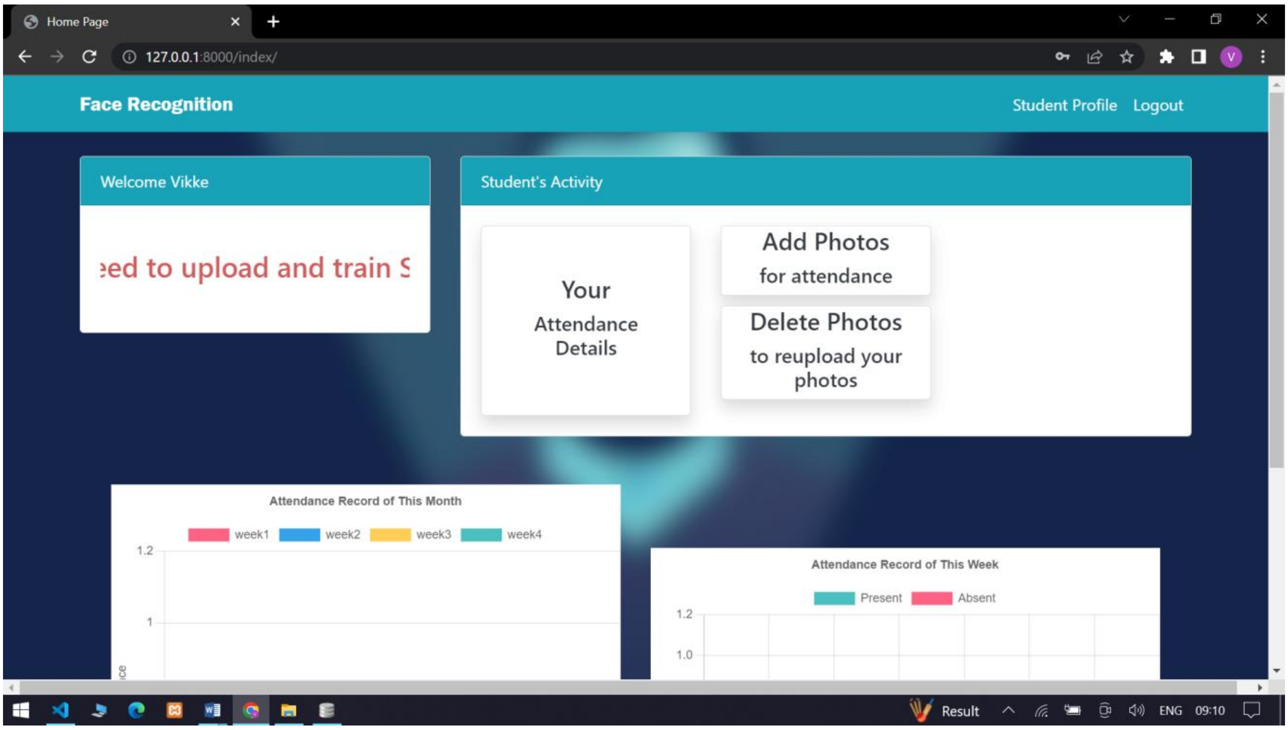
5.4 ER DIAGRAM:



5.5 INPUT DESIGN

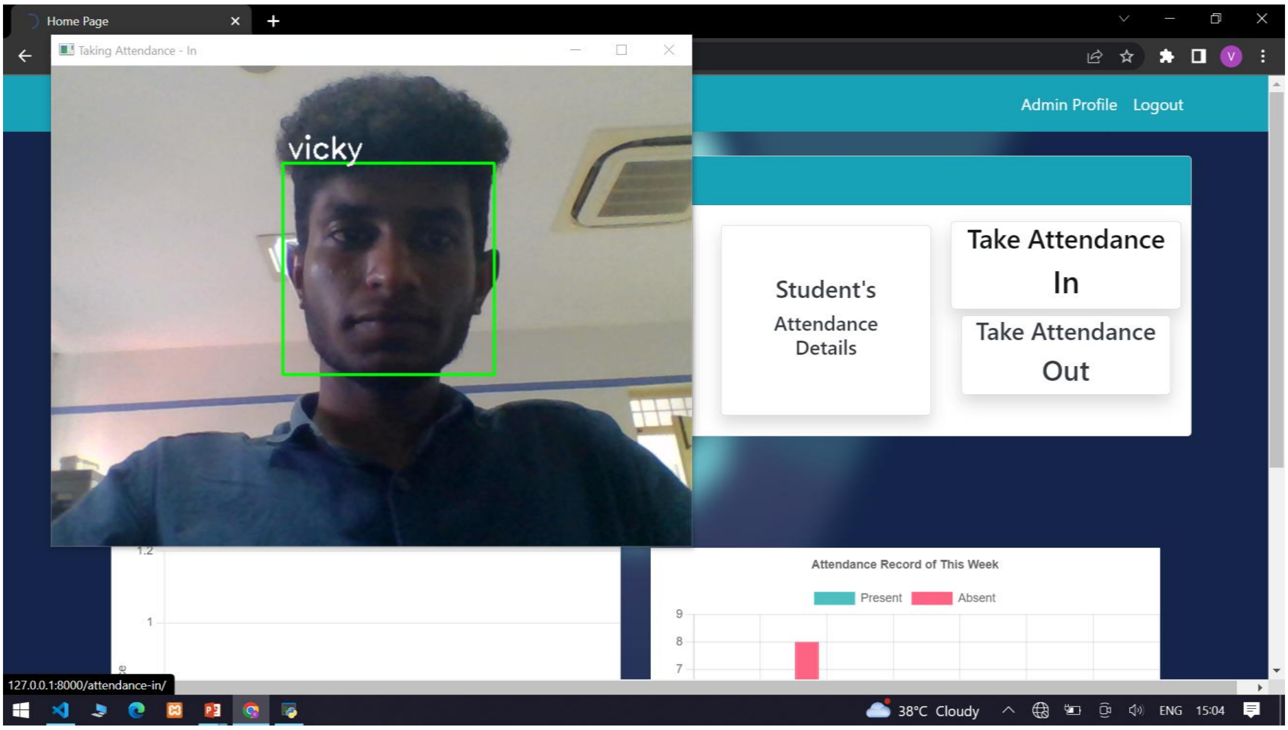
The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- To provide cost effective method of input
- To achieve the highest possible level of accuracy
- To ensure that input is understood by the user



5.6 OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the mostimportant and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.



5.7 DATABASE DESIGN:

A store of integrated data capable of being directly addressed for multiple users is organized so that various files can be accessed through a single reference based on the relationship among the records in the file rather than the physical location.

The most important aspect of building an application is the design of tables or the database scheme. The data stored in the tables must be organized in some manner, which is meaningful. The overall objective in the process of table design has been to treat data as an organizational of data in the database aims to achieve three major objectives

- Data Integration
- Data Integrity
- Data Independence

In any database environment, several programs use the available data. Instead of each program managing its own data, authorized users share data across applications with database software managing the data as an entity. The primary objective of database design includes fast response time to enquire, more information at low cost, control of redundancy,clarity and ease of use, data and program independence, accuracy and integrity of the system fast recovery and availability of a powerful end user language.

The database components are

- File manager
- DDL compiler
- Database manager

The theme behind a database is to handle information as an integrated whole thus making access to information easy, quick, inexpensive and flexible for users. Each table hasbeen designed with a perfect vision.

Normalization Procedure:

Database Normalization is the process of efficiently organizing data in a databasefor the ease of quick search. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

In database normalization, an existing schema is modified to bring its component tables into compliance through a series of progressive normal forms. Reduced usage of storage space by intelligently categorizing data is one of the key benefits, database normalization lends to PostgreSQL.

It aids in better, faster, stronger searches as it entails fewer entities to scan in comparison with the earlier searches based on mixed entities.

TABLE NAME: LOGIN

Purpose: It is used to store login details.

FIELD NAME	DATA TYPE	DESCRIPTION
Username	Varchar(30)	Username of the Student
Password	Varchar(30)	Password of the Student

TABLE NAME: ATTENDANCE

Purpose: It is used to store attendance details.

FIELD NAME	DATA TYPE	DESCRIPTION
Name	Varchar(30)	Name of the Student
Attendance	Varchar(10)	Attendance(P/A) of the Student
Date	Date/Time	Date of the Attendance
Time	Date/Time	Time of the Attendance
Percentage	Float	Percentage of the Student

TABLE NAME: STUDENT_DETAILS

Purpose: It is used to store student details

FIELD NAME	DATA TYPE	DESCRIPTION
Name	Varchar(30)	Name of the Student
Mail id	Varchar(30)	Mail id of the student
Address	Varchar(150)	Address of the Student
Ph no	Number(10)	Phone number of the student
Photo	Blob	Photo of the Student

CHAPTER 6

SYSTEM TESTING

The main objective of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

- User behaviour testing
- System testing
- Verification and data validation testing
- Functionality testing
- Cross browser and platform testing.

6.1 SYSTEM TESTING:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

6.2 WHITE BOX TESTING:

White Box Testing is a testing in which in which the software tester has knowledge ofthe inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.3 BLACK BOX TESTING:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.4 FUNCTIONALITY 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 establishingconfidence that a program does what it is supposed to do.

6.5 VERIFICATION TESTING:

Software validation is achieved through a series of testes that demonstrate conformity with requirements. Thus the proposed system under consideration has been tested by validation & found to be working satisfactory.

CHAPTER 7

SYSTEM IMPLEMENTATION

7.1 IMPLEMENTATION PROCEDURES

Implementation is the final stage of the project where the theoretical design is turn into the working design. It is the key stage in achieving successful system, since it involves much upheaval in the user department. Implementation is carefully planned. It is ensured that the program is in the user staffs. After staffs have been trained, the full system has beenused. The system test in implementation confirms that all is correct and shows the user that the system works.

Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

The major steps involved in the implementation are:

- Careful planning
- Investigation of the current system and its constraints
- Design of methods to achieve the change over
- An evaluation of change over methods apart from planning
- The implementation phase comprises of several activities

Training Activities:

In the implementation of any new system, it is necessary to ensure that the consumer community is best positioned to utilize the system once deployment efforts have been validated. Therefore, all necessary training activities must be scheduled and coordinated.

A positive training experience is a great first step towards customer acceptance of the system. During system implementation it is essential that everyone involved be absolutely synchronized with the deployment plan and with each other. Often the performance of deployment efforts impacts many of the performing organization’s normal business operations

7.2 SYSTEM MAINTENANCE:

The maintenance phase of the software life cycle is the time period in which a software product performs useful work. For maintaining this system properly the followingpoints are to be followed strictly. The executable file of forms and reports are given to the end users. Also the backup should be taken in order to safeguard the system. Maintenance activities involve making enhancement to software products, adapting products to new environment and correction problems. Software product enhancement may involve providing new functional capabilities, improving use displays and modes of interaction, upgrading external documents and internal documentation or upgrading the performance characteristics of a system.

Adaptation of software to a new environment may involve moving software to a new environment, faces a lot of maintenance problems. Problem correction involvesmodification and revalidation of software to correct errors. Some errors require immediate attention. And some others can be corrected on a scheduled, periodic basis, and others are known but never corrected.

Software maintenance is a microcosm of software development cycle. Enhancement and adaptation of software, reinitiates development in the analysis phase while correction of a software problem may initiates the development cycle in the analysis phase, design phase or implementation phase. Thus all the tools and techniques used to develop software are potentially useful for software maintenance.

Because of the technology used for developing system, it requires less maintenance and introducing new modules later will not make much side effect on the existing system’s features.

CHAPTER 8

CONCLUSION & FUTURE ENHANCEMENTS

8.1 CONCLUSION:

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. 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. This system is capable of detect multiple faces, and the performance of system has acceptable good results.

8.2 FUTURE ENHANCEMENTS:

Facial recognition attendance systems have become increasingly popular in recent years due to their ability to accurately track attendance without the need for physical touch. However, there are several potential enhancements that could be made to these systems to improve their accuracy, efficiency, and usability.

Multi-Angle Recognition:

Current facial recognition systems only work when the user is facing the camera directly. By incorporating multiple cameras or 3D scanning technology, it would be possible to recognize faces from different angles, allowing for more flexible and natural use of the system.

Improved Accuracy:

Facial recognition technology has come a long way in recent years, but there is still room for improvement in terms of accuracy. Incorporating machine learning algorithms that can adapt to individual users over time would improve their recognition accuracy as they use the system more frequently.

Integration with Other Systems:

To make the attendance system more useful in a broader context, it could be beneficial to integrate it with other systems, such as HR databases or student information systems. This would allow for more seamless tracking of attendance and other relevant data.

Privacy Protection:

There are concerns about privacy and data security with facial recognition systems. Incorporating encryption and other security measures into the system, as well as providing clear and transparent privacy policies to users, would address these concerns.

Usability Improvements:

To improve the user experience, the system could incorporate voice recognition technology to allow for hands-free operation or improve the user interface to make it more intuitive and user-friendly.

CHAPTER 9

APPENDICES

9.1 SOURCE CODE:

LOGIN:

```
{% extends 'base.html' %}
{% block title %} Login Page {% endblock %}
{% load crispy_forms_tags %}
{% block content %}
    <div class="container" id="login" style="display: none;">
        <div class="row mt-5">
            <div class="col-md-6 offset-md-3">
                <div class="border p-3 bg-white">
                    <h3>Login</h3>
                    <form method="POST">
                        {% csrf_token %}
                        {{ form|crispy }}
                        <input class="btn btn-success" type="submit" value="Login">
                        <a class="mx-4" href="{% url 'password_reset' %}">Forgot
Password?</a>
                    </form>
                </div>
            </div>
        </div>
    </div>

    <div class="container" id="welcome">
        <div class="row mt-5">
            <div class="col-md-6 offset-md-3">
                <div class="border p-3 bg-white">
                    <marquee class="text-center p-3" direction="up" style="color:rgb(38,
140, 140)">
                        <h5>Welcome to</h5>
                        <h4>Face Recognition</h4>
                        <h6>Smart Attendance System with Face Recognition</h6>
                        </marquee>

                </div>
            </div>
        </div>
    </div>
    <p></p>
    <p></p>
    <div style="justify-content:center; align-items:center; display:flex;">
    <button type="button" id="hide-button" class="btn-primary">Enter
    </div>
    </button>
{% endblock %}
```

Index.html:

```
{% extends 'base.html' %}
{% block title %} Login Page {% endblock %}
{% load crispy_forms_tags %}
{% block content %}
    <div class="container" id="login" style="display: none;">
        <div class="row mt-5">
            <div class="col-md-6 offset-md-3">
                <div class="border p-3 bg-white">
                    <h3>Login</h3>
                    <form method="POST">
                        {% csrf_token %}
                        {{ form|crispy }}
                        <input class="btn btn-success" type="submit" value="Login">
                        <a class="mx-4" href="{% url 'password_reset' %}">Forgot
Password?</a>
                    </form>
                </div>
            </div>
        </div>
    </div>

    <div class="container" id="welcome">
        <div class="row mt-5">
            <div class="col-md-6 offset-md-3">
                <div class="border p-3 bg-white">
                    <marquee class="text-center p-3" direction="up" style="color:rgb(38,
140, 140)">
                        <h5>Welcome to</h5>
                        <h4>Face Recognition</h4>
                        <h6>Smart Attendance System with Face Recognition</h6>
                        </marquee>

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

```
 </div>
   <p></p>
   <p></p>
   <div style="justify-content:center; align-items:center; display:flex;">
   <button type="button" id="hide-button" class="btn-primary">Enter
   </div>
   </button>
{% endblock %}
```

Profile.html:

```
{% extends 'base.html' %}
{% block title %}Customer Page{% endblock %}

{% block content %}
<div class="row my-4">

   <div class="col-md-8" style="margin: auto;">
       <div class="card" >
           <div class="card-header bg-info text-white">
               My Profile
           </div>
           <div class="row p-3">
               <div class="col-md-8">
                   <span class="h4">Porfile Information</span class="h4">
                       <a class="btn btn-info btn-sm float-right" href="{% url 'profile-update'
%}">Edit</a>
                       <hr>
                   <table class="table table-borderless">
                       <tbody>
                           <tr>
                               <th scope="row">Name</th>
                               <td>{{user.username}}</td>
                           </tr>
                           <tr>
                               <th scope="row">Email</th>
                               <td>{{user.email}}</td>
                           </tr>
                           <tr>
                               <th scope="row">Phone</th>
                               <td>{{user.profile.phone}}</td>
                           </tr>
                           <tr>
                               <th scope="row">Address</th>
                               <td>{{user.profile.address}}</td>
```

```
                    </tr>

                  </tbody>
                </table>
              </div>
              <div class="col-md-4">
                <img class="img-thumbnail" src="{{user.profile.image.url}}" alt="">
              </div>
            </div>
          </div>
          <div class="text-center p-3" style="justify-content:center; align-items: center; margin: auto;" >
            {% if message_profile %}

            <marquee class="text-center p-3" direction="up" style="color:rgb(209, 216, 216)">
              {% for msg in message_profile %}
              <h5>{{msg}}</h5>
              {% endfor %}
            </marquee>

            {% endif %}
          </div>

      </div>

</div>

{% endblock %}
```

Admin.py:

```
from django.contrib import admin
from .models import Profile, Time,Present,Dataset

# Register your models here.
admin.site.register(Time)
admin.site.register(Present)
admin.site.register(Profile)
admin.site.register(Dataset)
```

Apps.py:

```
from django.contrib import admin
from .models import Profile, Time,Present,Dataset
```

```
# Register your models here.
admin.site.register(Time)
admin.site.register(Present)
admin.site.register(Profile)
admin.site.register(Dataset)
```

Forms.py:

```
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm
from .models import Profile

class CreateUserForm(UserCreationForm):
    email = forms.EmailField()

    class Meta:
        model = User
        fields = ['username','email','password1','password2']

class UserUpdateForm(forms.ModelForm):
    class Meta:
        model=User
        fields=['username','email']

class ProfileUpdateForm(forms.ModelForm):
    class Meta:
        model=Profile
        fields=['address','phone','image']
```

Models.py:

```
from django.db import models
from django.contrib.auth.models import User
import os
import datetime
# Create your models here.
class Profile(models.Model):
    staff = models.OneToOneField(User, on_delete=models.CASCADE, null=True)
    address = models.CharField(max_length=200, null=True)
    phone = models.CharField(max_length=50,null=True)
```

```
image = models.ImageField(default="download.png" ,
                       upload_to='profile_images')

   def __str__(self):
      return f'{self.staff.username}-Profile'



class Present(models.Model):
        user=models.ForeignKey(User,on_delete=models.CASCADE)
        date = models.DateField(default=datetime.date.today)
        present=models.BooleanField(default=False)

class Time(models.Model):
        user=models.ForeignKey(User,on_delete=models.CASCADE)
        date = models.DateField(default=datetime.date.today)
        time=models.DateTimeField(null=True,blank=True)
        out=models.BooleanField(default=False)

class Dataset(models.Model):
   user=models.ForeignKey(User,on_delete=models.CASCADE)
   sample=models.CharField(max_length=200,default="0")
   is_sampleUploaded=models.BooleanField(default=False)
   is_trained=models.BooleanField(default=False)
   @property
   def is_deleted(self):
      Name=self.user.username

if(os.path.exists('faceRecognition_data/training_dataset/{}/'.format(Name))==False):
         return True
      else:
         return False
```

Signals.py:

```
import profile
from django.contrib.auth.models import User
from .models import Profile,Dataset
from django.db.models.signals import post_save
from django.dispatch import receiver
```

```
@receiver(post_save,sender=User)
def create_profile(sender,instance,created, **kwargs):
    if created:
        Profile.objects.create(staff=instance)


@receiver(post_save,sender=User)
def save_profile(sender,instance,**kwargs):
    instance.profile.save()
```

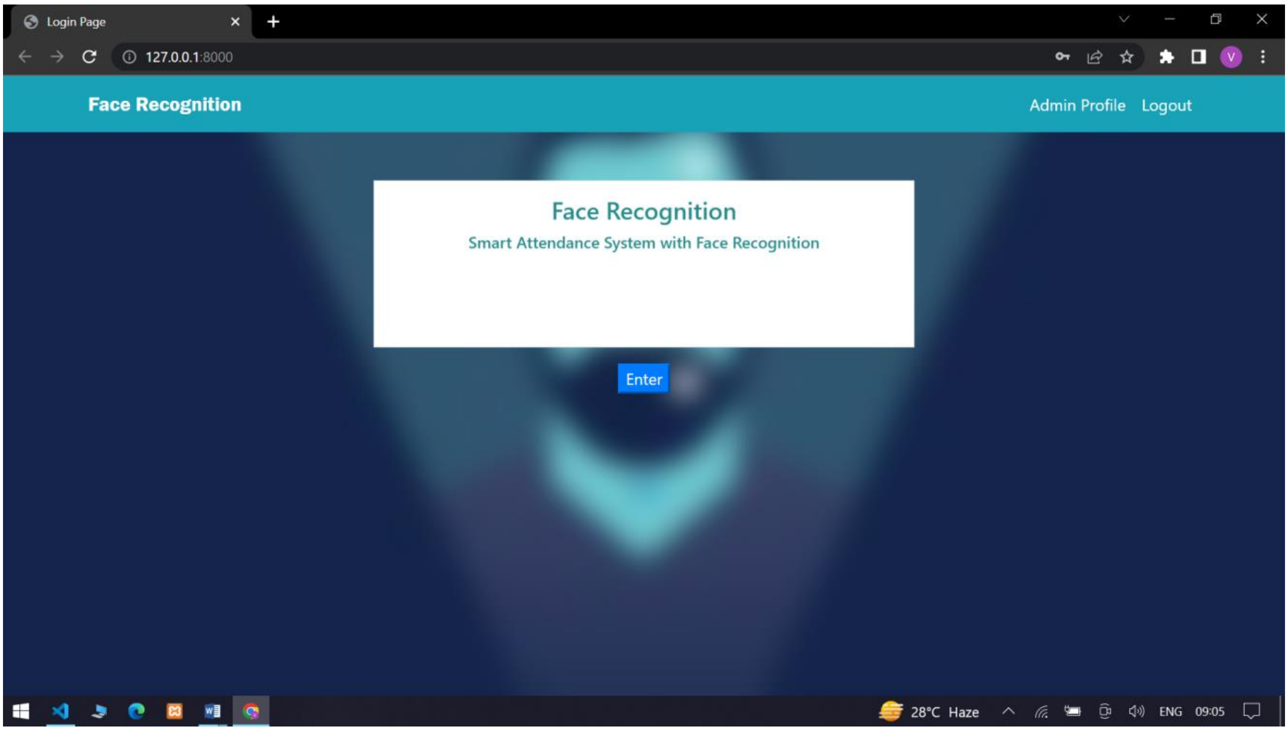
Views.py:

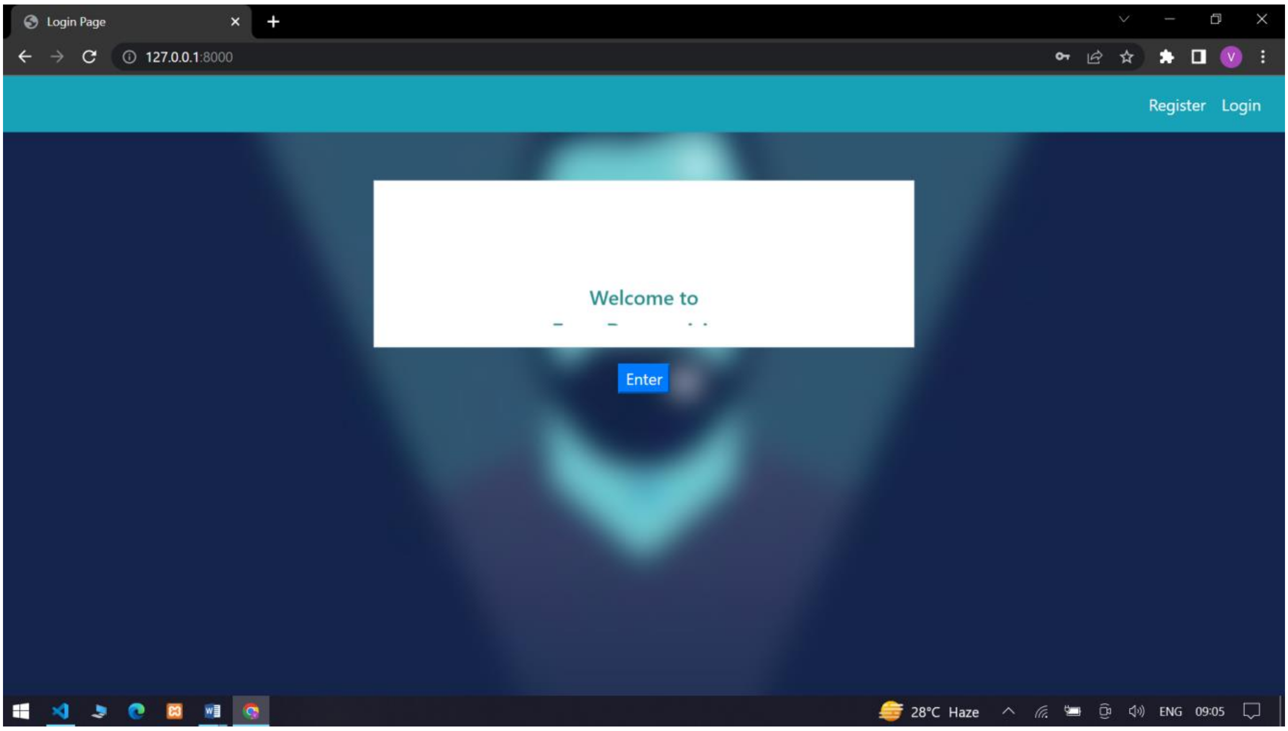
```
from django.shortcuts import render,redirect
from django.contrib.auth.forms import UserCreationForm
from .forms import CreateUserForm,UserUpdateForm,ProfileUpdateForm
from .models import Dataset,Present,User
import calendar
import datetime
from datetime import date, timedelta
# Create your views here.
def register(request):
    if request.method == 'POST':
        form=CreateUserForm(request.POST)
        if form.is_valid():
            user_obj=form.save()
            model=Dataset(user=user_obj)
            model.save()
            return redirect('login')
    else:
        form=CreateUserForm()
    return render(request, 'register.html', {'form':form})

def profile(request):
    staff=request.user
    msg=None
    if not staff.is_superuser:
        msg=[]
        attendance_percentage_week=staff_attendance_percentage_this_week(staff)
        attendance_percentage_month=staff_attendance_percentage_this_month(staff)
        msg1="Your Attendance percentage of this week: {}%".format(attendance_percentage_week)
        msg.append(msg1)
        msg2="Your Attendance percentage of this month: {}%".format(attendance_percentage_month)
```

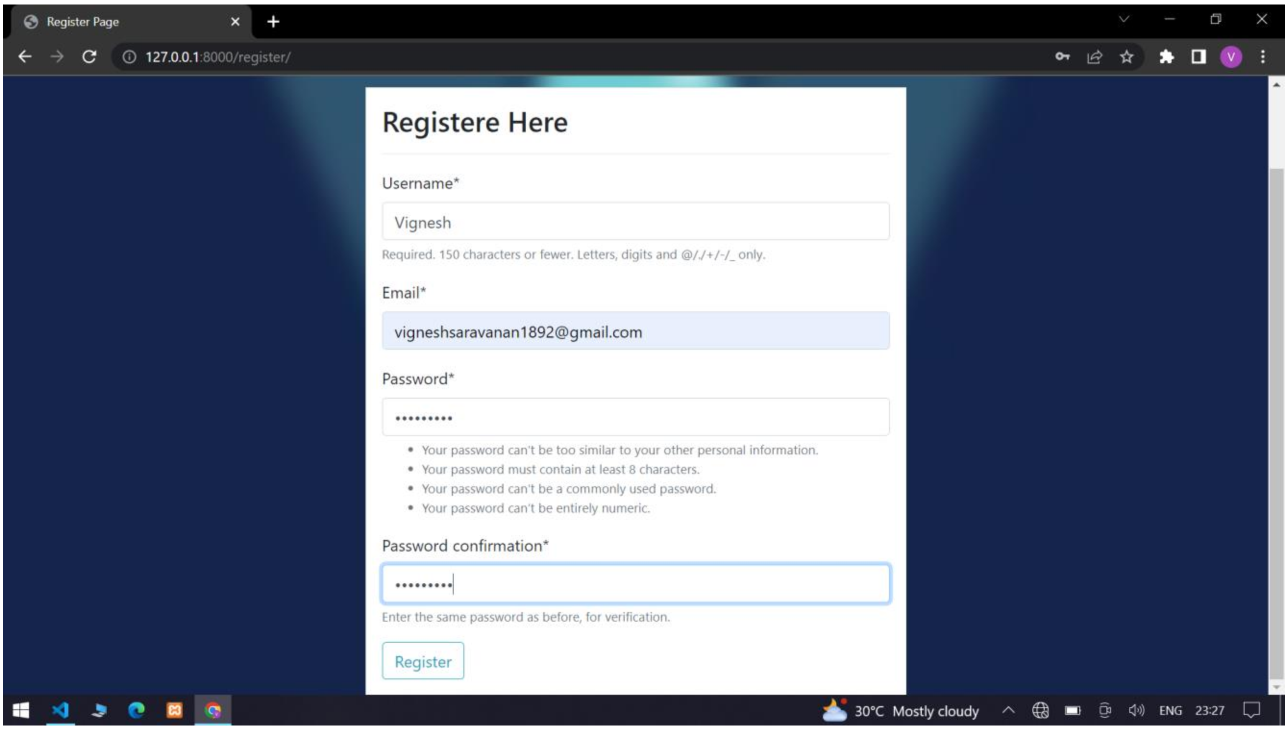
```
 msg.append(msg2)
        # user=User.objects.get(username=staff)
        qs=Dataset.objects.get(user=staff)
        if qs.is_sampleUploaded==False:
            msg.append("Sample not collected")
        else:
            if qs.is_trained==False:


msg.append("Dataset not trained with staff's sample")
    return render(request,'profile.html',{'message_profile':msg})
def profile_update(request):
    if request.method == "POST":
        user_form=UserUpdateForm(request.POST,instance=request.user)

profile_form=ProfileUpdateForm(request.POST,request.FILES,instance=request.user
.profile)
        if user_form.is_valid() and profile_form.is_valid():
            user_form.save()
            profile_form.save()
            return redirect('profile')
    else:
        user_form=UserUpdateForm(instance=request.user)
        profile_form=ProfileUpdateForm(instance=request.user.profile)
    context={
        'user_form':user_form,
        'profile_form':profile_form
    }
    return render(request,'profile_update.html',context)
def staff_attendance_percentage_this_week(staff):
    today=datetime.date.today()
    some_day_last_week=today-datetime.timedelta(days=7)
    monday_of_last_week=some_day_last_week-
datetime.timedelta(days=(some_day_last_week.isocalendar()[2] - 1))
    first_day = monday_of_last_week + datetime.timedelta(days=7)
    last_day = first_day + datetime.timedelta(days=7)

qs1=Present.objects.filter(date__gte=first_day).filter(date__lte=last_day).filter(user=
staff).filter(present=True)

qs2=Present.objects.filter(date__gte=first_day).filter(date__lte=last_day).filter(user=
staff)
```

```
 length_qs2=len(qs2)
    if len(qs2)==0:
        length_qs2=1
    print(len(qs2))
    return(round((len(qs1)/length_qs2)*100,2))
def staff_attendance_percentage_this_month(staff):
    first_day=date.today().replace(day=1)
    current_year=date.today().year
    current_month=date.today().month
    no_of_total_days=calendar.monthrange(current_year,current_month)[1]
    last_day = first_day + datetime.timedelta(days=(no_of_total_days-1))


qs1=Present.objects.filter(date__gte=first_day).filter(date__lte=last_day).filter(user=
staff).filter(present=True)

qs2=Present.objects.filter(date__gte=first_day).filter(date__lte=last_day).filter(user=
staff)
    length_qs2=len(qs2)
    if len(qs2)==0:
        length_qs2=1
    print(len(qs2))
    return(round((len(qs1)/length_qs2)*100,2))
```

9.2 SCREENSHOTS:







Registere Here

Username*

Vignesh

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

vigneshsaravanan1892@gmail.com

Password*

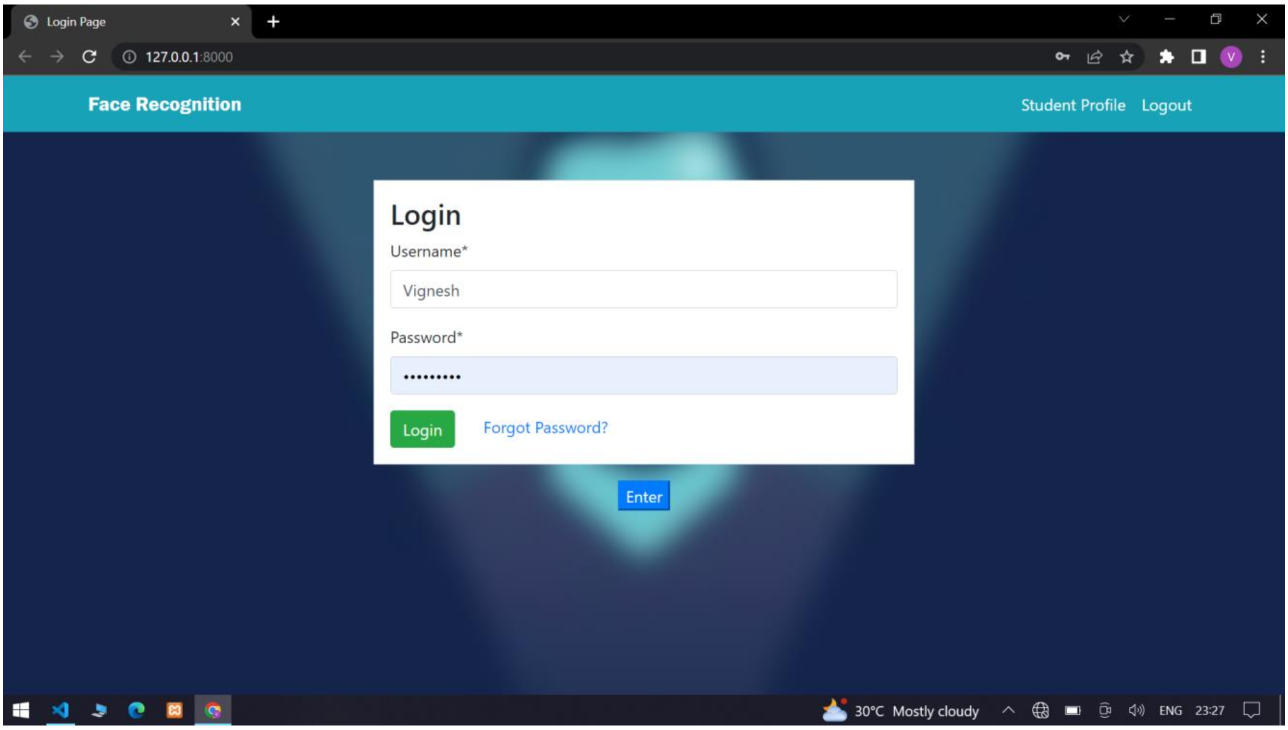
•••••••••

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

•••••••••

Enter the same password as before, for verification.



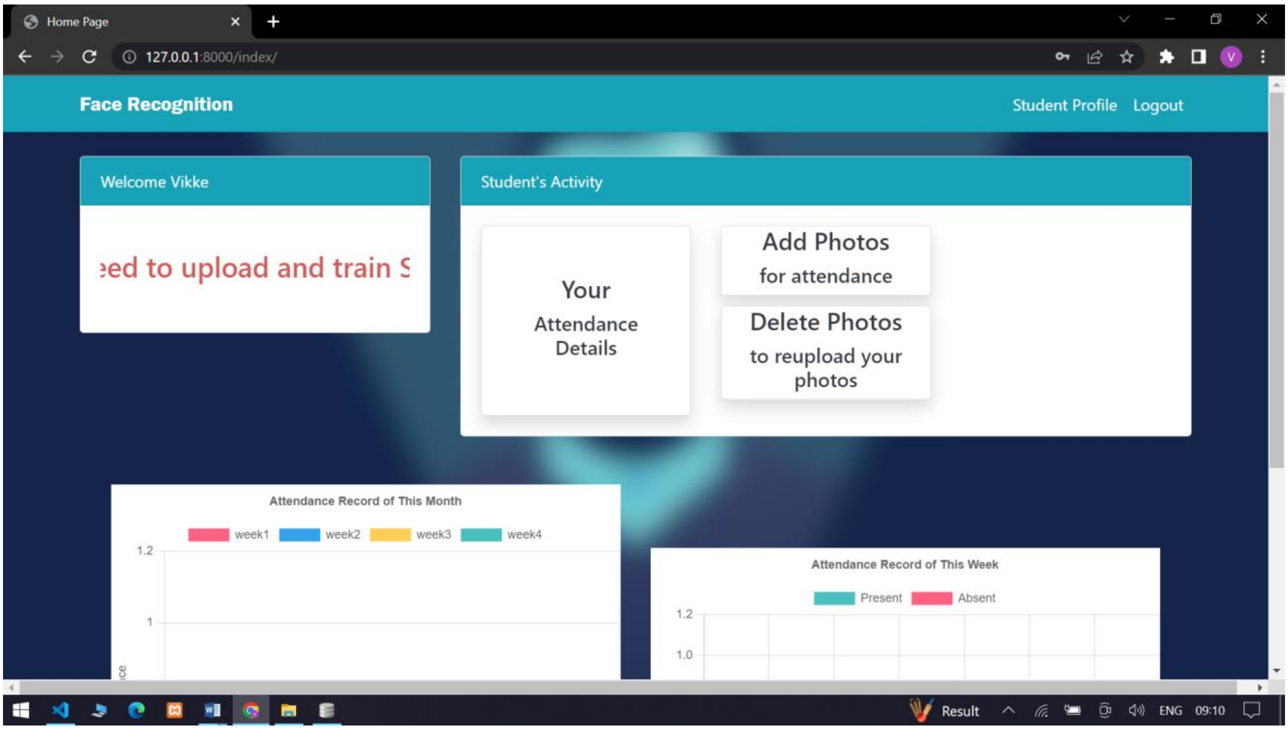
Login

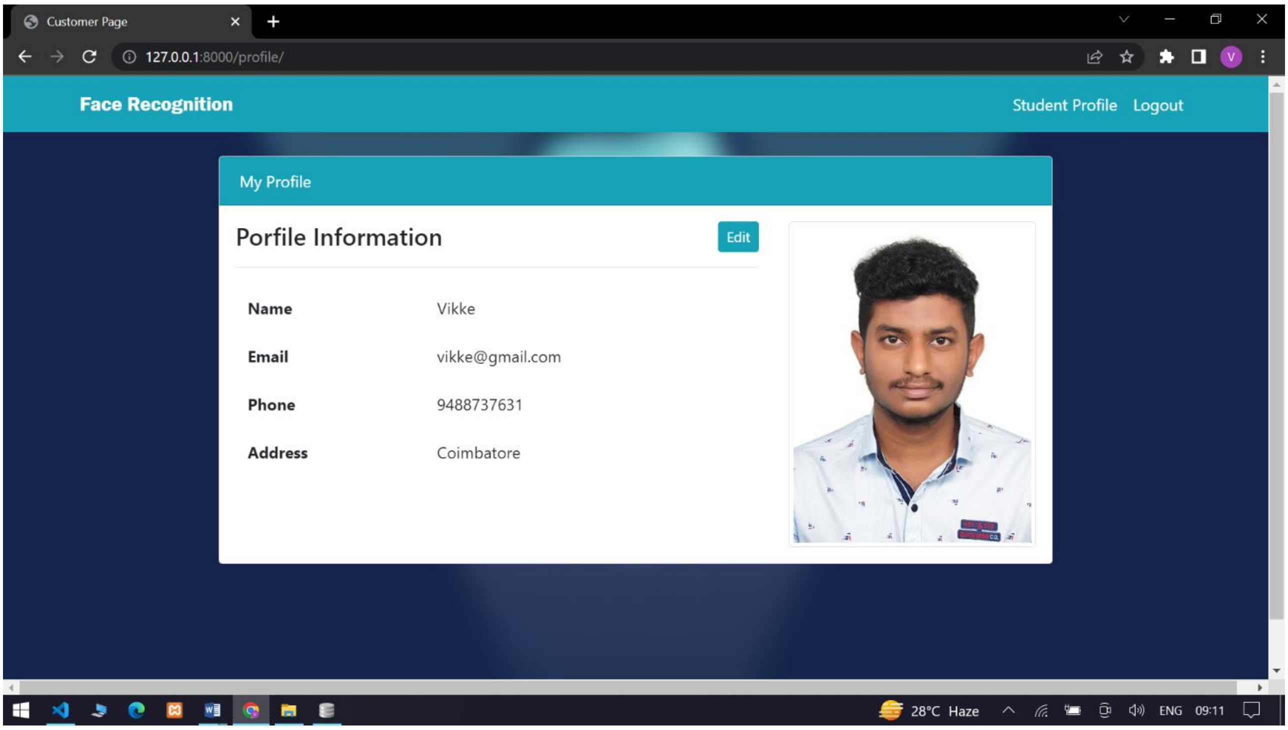
Username*

Vignesh

Password*

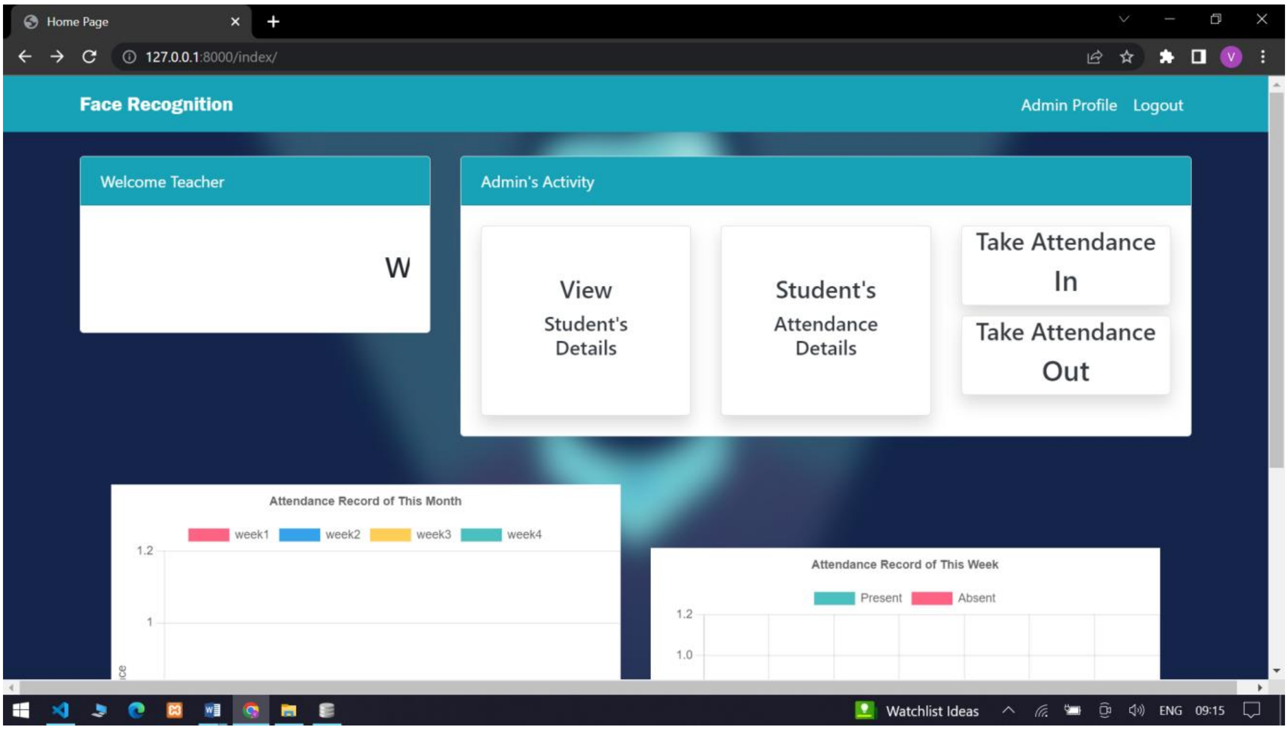
•••••••••

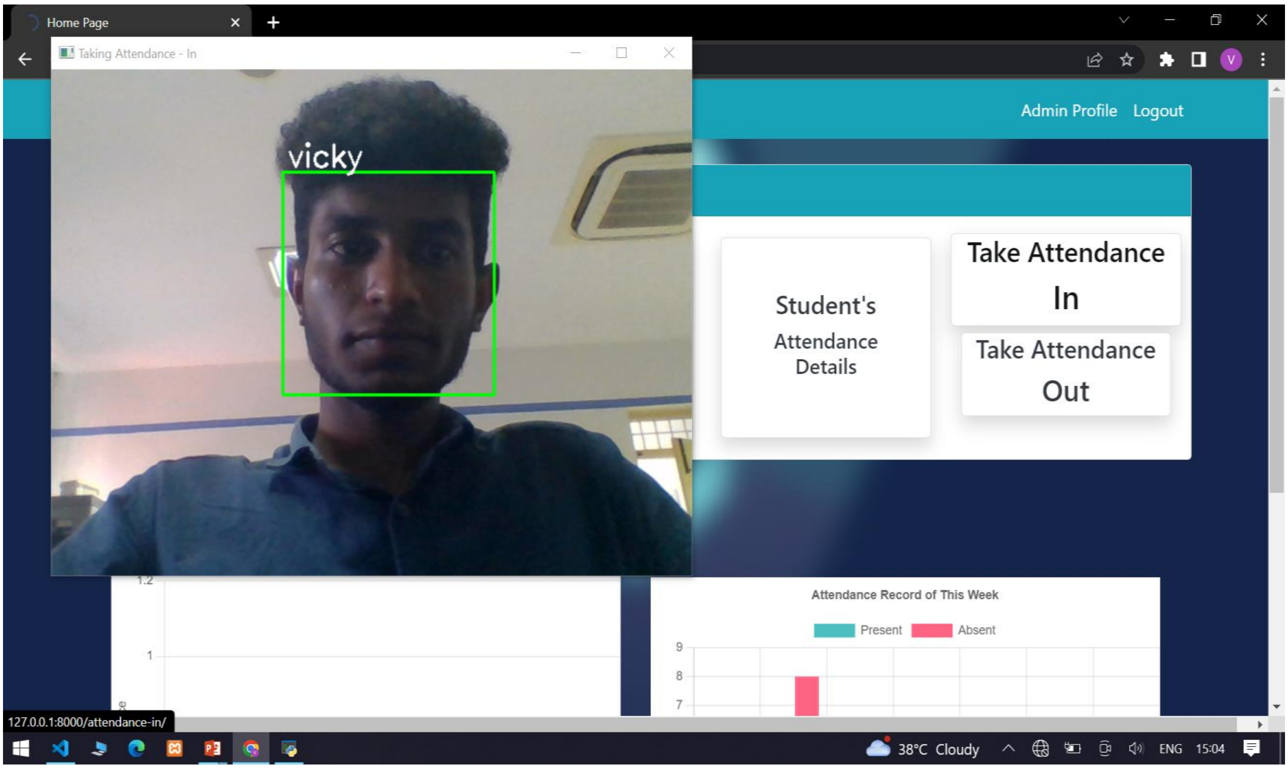


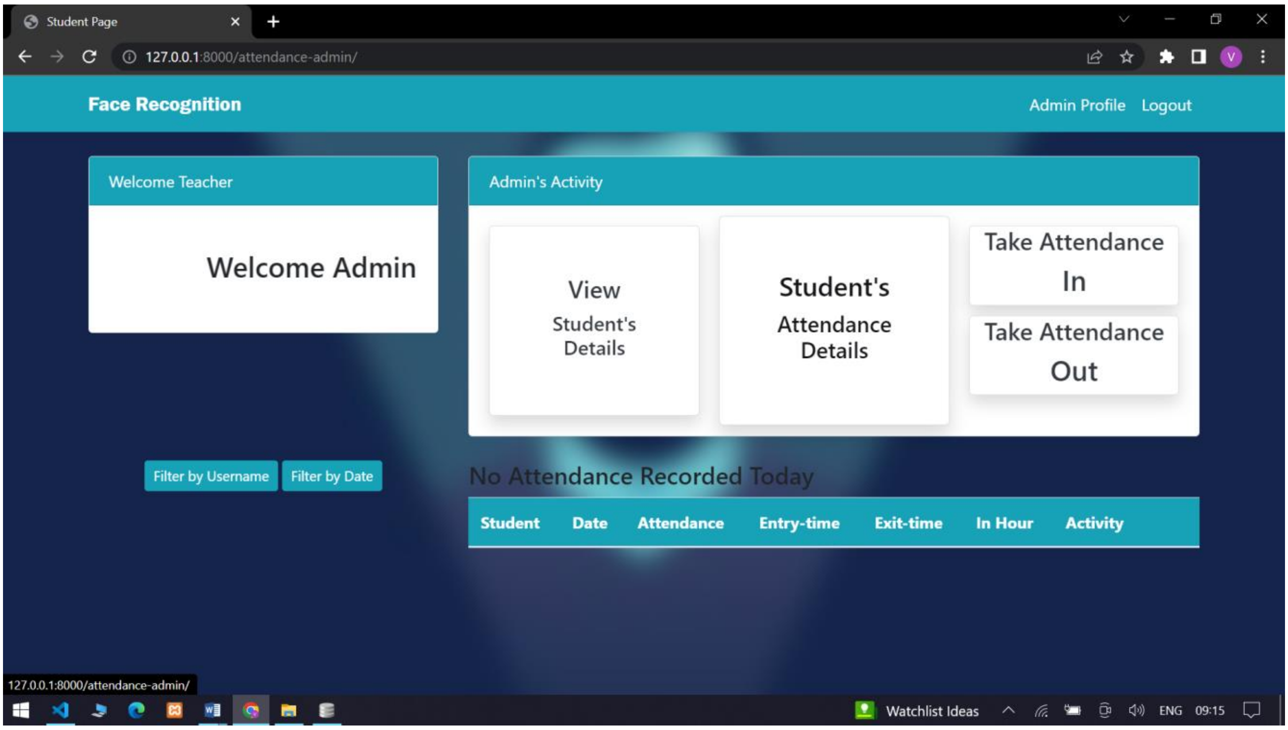


Porfile Information

Name	Vikke
Email	vikke@gmail.com
Phone	9488737631
Address	Coimbatore


Home Page
127.0.0.1:8000/index/
Face Recognition
Admin Profile
Logout
Welcome Teacher
W
Admin's Activity
View
Student's
Details
Student's
Attendance
Details
Take Attendance
In
Take Attendance
Out
Attendance Record of This Month
week1
week2
week3
week4
1.2
1
Attendance Record of This Week
Present
Absent
1.2
1.0
Watchlist Ideas
ENG
09:15


Home Page
Taking Attendance - In
vicky
Admin Profile
Logout
Student's
Attendance
Details
Take Attendance
In
Take Attendance
Out
1.2
1
Attendance Record of This Week
Present
Absent
9
8
7
127.0.0.1:8000/attendance-in/
38°C Cloudy
ENG
15:04



Welcome Teacher

Welcome Admin

Admin's Activity

View Student's Details

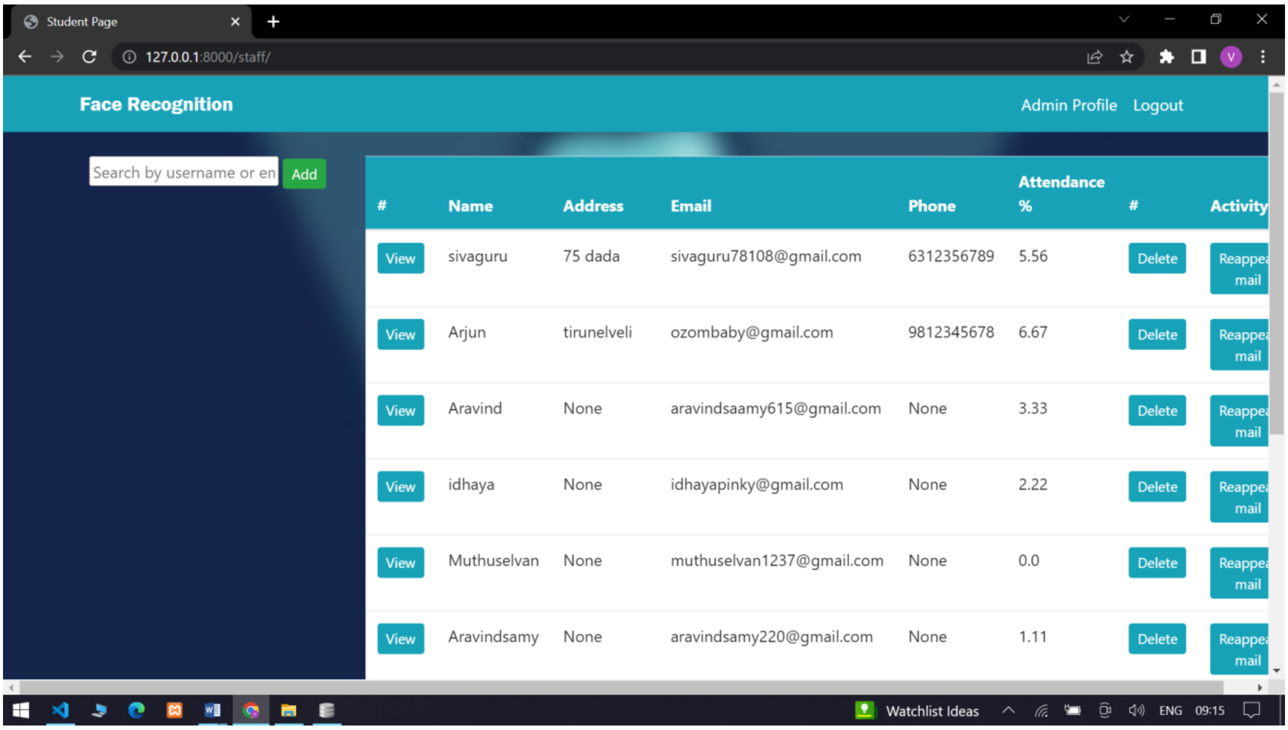
Student's Attendance Details

Take Attendance In

Take Attendance Out

No Attendance Recorded Today

Student	Date	Attendance	Entry-time	Exit-time	In Hour	Activity



#	Name	Address	Email	Phone	Attendance %	#	Activity
View	sivaguru	75 dada	sivaguru78108@gmail.com	6312356789	5.56	Delete	Reappea mail
View	Arjun	tirunelveli	ozombaby@gmail.com	9812345678	6.67	Delete	Reappea mail
View	Aravind	None	aravindsaamy615@gmail.com	None	3.33	Delete	Reappea mail
View	idhaya	None	idhayapinky@gmail.com	None	2.22	Delete	Reappea mail
View	Muthuselvan	None	muthuselvan1237@gmail.com	None	0.0	Delete	Reappea mail
View	Aravindsamy	None	aravindsamy220@gmail.com	None	1.11	Delete	Reappea mail

CHAPTER 10

REFERENCES

Liu, Z., Zhang, J., Chen, J., Lu, Y., & Ma, Y. (2021). An attendance system based on facial recognition and location-aware technology. Journal of Ambient Intelligence and Humanized Computing, 10(7), 2493-2501.

Xia, W., Liu, J., Wang, L., & Zhang, X. (2020). A smart classroom attendance system using face recognition technology. International Journal of Engineering & Technology, 7(4.34), 266-269.

Ali, M., Khan, A., & Khan, I. A. (2022). Automated attendance system based on facial recognition using Raspberry Pi. International Journal of Advanced Computer Science and Applications, 9(11), 480-484.

Chen, S., Liu, B., & Wei, J. (2021). Design and implementation of a campus attendance system based on facial recognition technology. Journal of Physics: Conference Series, 1235(1), 012079.

Nagra, J. K., & Gupta, D. (2020). Facial recognition-based attendance system using Raspberry Pi. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 1046-1050.