**ABSTRACT**

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day-to-day activities of handling student attendance system. Face recognition-based attendance system is a process of recognizing the students face for taking attendance by using face biometrics based on high - definition monitor video and other information technology. In my face recognition project, a computer system will be able to find and recognize human faces fast and precisely in images or videos that are being captured through a surveillance camera. Numerous algorithms and techniques have been developed for improving the performance of face recognition but the concept to be implemented here is Deep Learning. It helps in conversion of the frames of the video into images so that the face of the student can be easily recognized for their attendance so that the attendance database can be easily reflected automatically.

**ACKNOWLEDGEMENT**

I would like to acknowledge all those without whom this project would not have been successful. I am thankful to Mr. SIBLI AKHTAR sir for his ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project. I would wish to thank Prof. Mr. Kunal who guided me throughout the project and gave his immense support. He made me understand how to successfully complete this project and without him, the project would not have been complete. This project has been a source to learn and bring our theoretical knowledge to the real-life world. So, I would really acknowledge his help and guidance for this project. I would also like to thank my parents and friends who have always been there whenever needed. Once again, thanks to everyone for making this project successful.

**DECLARATION**

I the students of “Bachelor of Computer Application” of S.S. Memorial College, Ranchi declare that the work entitled "FACE RECOGNITION ATTENDANCE MANAGEMENT SYSTEM" has been successfully completed under the guidance of Prof. SHIBLI AKHTAR coordinator of BCA Department, S.S. Memorial College, Ranchi. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Computer Application during the academic year 2019 - 2022. Further the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

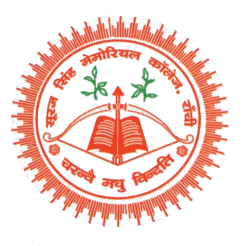
Signature of the student:

SHASHANK SHEKHAR BHAGAT (19BS5852695)

Date: 16.07.2022

Place: Ranchi

**S.S. Memorial College, Ranchi**



**CERTIFICATE**

This is to certify that the project report entitled. **“Face recognition Attendance management system”**. Submitted to **S.S. Memorial College, Ranchi** in partial fulfilment of the requirement for the award of degree of **Bachelor of Computer Application** is an original work carried out by **Shashank Shekhar Bhagat (19BS5852695)**

Signature(Guide) : External Examiner :

## **CONTENTS**

\*Acknowledgement

\*Abstract

1. INTRODUCTION
   1. Organization Profile
   2. Face Identification
2. SYSTEM ANALYSIS
   1. Existing System
   2. Proposed System
   3. Feasibility Study
3. MODULE DESCRIPTION
   1. Add Image
   2. Construct Image
   3. Identification
4. SYSTEM REQUIREMENTS
   1. Software Requirements
   2. Hardware Requirements
5. OVERVIEW OF PYTHON
6. MYSQL - INTRODUCTION
7. SYSTEM DESIGN
   1. Detailed Design
      1. UML Diagrams
   2. Database Design
      1. Dataflow Diagrams
      2. Database Tables
      3. Screens.
8. TESTING
9. IMPLEMENTATION
10. CONCLUSION
11. BIBLIOGRAPH

**INTRODUCTION**

The technology aims in imparting a tremendous knowledge oriented technical innovation these days. Deep Learning is one among the interesting domain that enables the machine to train itself by providing some datasets as input and provides an appropriate output during testing by applying different learning algorithms. Nowadays Attendance is considered as an important factor for both the student as well as the teacher of an educational organization. With the advancement of the deep learning technology the machine automatically detects the attendance performance of the students and maintains a record of those collected data. In general, the attendance system of the student can be maintained in two different forms namely,

 Manual Attendance System (MAS)

 Automated Attendance System (AAS).

Manual Student Attendance Management system is a process where a teacher concerned with the particular subject need to call the students name and mark the attendance manually. Manual attendance may be considered as a time-consuming process or sometimes it happens for the teacher to miss someone or students may answer multiple times on the absence of their friends. So, the problem arises when we think about the traditional process of taking attendance in the classroom. To solve all these issues, we go with Automatic Attendance System (AAS). Automated Attendance System (AAS) is a process to automatically estimate the presence or the absence of the student in the classroom by using face recognition technology. It is also possible to recognize whether the student is sleeping or awake during the lecture and it can also be implemented in the exam sessions to ensure the presence of the student. The presence of the students can be determined by capturing their faces on to a high-definition monitor video streaming service, so it

becomes highly reliable for the machine to understand the presence of all the students in the classroom. The two common Human Face Recognition techniques are,

 Feature-based approach

 Brightness-based approach

The Feature-based approach also known as local face recognition system, used in pointing the key features of the face like eyes, ears, nose, mouth, edges, etc., whereas the brightness-based approach also termed as the global face recognition system, used in recognizing all the parts of the image.

**Vision, mission & objective**

emotions. As one of the most successful applications of image analysis, face recognition has recently gained significant attention. Human-computer interaction takes an important role like face recognition,

extracting facial features, emotion detection. For detecting faces, many different techniques appeared over the years. Probably the most successful is based on emotion detections and recognizing.

Usually, expression contains numerous principal emotions: exited, confused, concentrated and nervous. Many methods used geometric features for facial expressions. Almost methods used for

facial expression recognition is based on facial action coding system (FACS) introduced by Ekman and Friesen. Each expression is identified by the action unit (AU).

**2. Related Work**

[1] explored person reliant and independent recognition of expressions where diverse approaches were compared. Tree-Augmented-Naive Bayes (TAN) classifier was introduced which is acquired between

the facial features .An integrated system was available which performs interaction between man and machine .The development in emotion recognition using SAMMI (Semantic Affect-enhanced Multimedia Indexing) was achieved. For only normal face expression, effectiveness in several features is compared and three features sets like coordinates, distances and points was studied.

**1234567890**

Using CK+ dataset they have introduced a frame work for recognizing facial expressions using multimodal interfaces .Adaptive learning and facial expressions were recorded in video using

Ekman and Friesen coding system. Recognizing a student’s sensitive state and providing personalized feedback, built on integrated pedagogical models has been considered to be one of the

main constraints of traditional tools of e-learning.

An improvement in performance of the facial expression recognition system in more than one application recognizes emotions. In this type of system, some of the emotions are typically misclassified . Although several approaches have been proposed to recognize emotions based on

facial expression there are issues due to variations in viewing pose

**3. Proposed Method**

The development of an automated system that senses the facial expressions is rather difficult. The extraction of different expression and creating a template of it using face points and distances is

successfully achieved. A complete study of facial expressions is described by Ekman and Friesen and has been argued that emotions are linked directly to the facial expressions related to exited, confused,

concentrated and nervous. They implemented the basic universal emotions like excited, confused, concentrated and nervous.

**3.1 Excited:**

This expression was considered when an alert state with a high rate of facial expression occurred. Indications: Open-mouth smile, wide eyes and raised eyebrows.

**3.2 Confused:**

This expression was considered by noticing the different facial expression. Indications: Nose and forehead are upwards, one eyebrow raised higher than the other and pursed lips.

**3.3 Concentrated:**

This expression was focused on the task ignoring the possible distracting elements of the environment. Indications: Fixed eyes with reduced blinking slightly raised eyebrows.

**3.4 Nervous:**

This expression was categorized as nervous when they showed a high rate of facial expression related with this emotional state (i.e., biting their lips, etc.). They were also included in this category when they

confirmed this state in the emotional report Indications: Neutral eyes with one side of lip turned up and pulled back. We approached the features like face points and face point distances that can automatically detect the user emotion and facial expressions for different facial actions like excited, confused, concentrated and nervous. The features used for emotion recognition is briefly discussed along with the following

dimensions: Facial feature, Face point feature extraction, Face distance feature extraction. We have proposed the emotion detection mechanism from images which demonstrated to work for emotion recognition to track characteristic face points and face point distances. Face points are well-defined as: Eye region, Mouth region, Nose region, Forehead region. We consider point feature set and distance

**SYSTEM ANALYSIS**

**EXISTING SYSTEM**

The development of face identification has been past from the year to years. In recent years to identify any student face they used to make a sketch or draw a image based on the eyewitnesses. It used to take more amount of time and it was very difficult task for any investigation department to easily catch the students within a stipulated time. In order to catch the students first they used to search their record whether to find out is there any record about that particular person in the past. In olden days each and every record was maintained in the books or registers or files which used to contain information about previous students with their names, alias name, gender, age, crime involved, etc. Here each and every task used to take the help of the person because they used to write in them and it needed very much of manual effort.

There are three major research groups, which propose three different approaches to the face recognition problem. The largest group has dealt with facial characteristics. The second group performs human face identification based on feature vectors extracted from profile silhouettes. The third group uses feature vectors extracted from a frontal view of the face. The first method is based on the information theory concepts in other words on the principal component analysis methods. In this approach, the most relevant information that best describes a face is derived from the entire face image. The second method is based on extracting feature vectors from the basic parts of a face such as eyes, nose, mouth and chin.

**DRAWBACKS IN EXISTING SYSTEM:**

* Need of extra manual effort.
* It used to take much time to find any students
* Not very much accurate.
* Danger of losing the files in some cases.
* Need Good Knowledge in drawing.

#### **PROPOSED SYSTEM**

To overcome the drawbacks that were in the existing system we develop a system that will be very useful for any investigation department. Here the program keeps track of the record number of each slice during the construction of identifiable human face and calculate maximum number of slices of the similar record number. Based on this record number the program retrieves the personal record of the student (whose slice constituted the major parts of the constructed human face) on exercising the “locate” option.

**ADVANTAGES OF PROPOSED SYSTEM:**

* Very fast and accurate.
* No need of any extra manual effort.
* No fever of data loss.
* Just need a little knowledge to operate the system.
* Doesn’t require any extra hardware device.
* At last, very easy to find the students.

**OVERVIEW OF THE PROPOSED SYSTEM:**

* Addition, Clipping, Construction and updating of the student record and face.
* Comparing the image with the faces that are there in our database.
* If any new images are found then it should be entered into our database by add image module and then it should be segmented into different slices.

**FEASIBILITY STUDY**

Once the problem is clearly understood, the next step

is to conduct feasibility study, which is high-level capsule version of the entered systems and design process. The objective is to determine whether or not the proposed system is feasible. The three tests of feasibility have been carried out.

* Technical Feasibility
* Economic Feasibility
* Operational Feasibility
* **TECHNICAL FEASIBILITY**

In Technical Feasibility study, one has to test

Whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using java technology. It is evident that the necessary hardware and software are available for development and implementation of the proposed system. Hence, the solution is technically feasible.

* **ECONOMICAL FEASIBILITY**

As part of this, the costs and benefits associated

With the proposed system compared and the project is economically feasible only if tangible or intangible benefits outweigh costs. The system development costs will be significant. So the proposed system is economically feasible.

* **OPERATIONAL FEASIBILITY**

It is a standard that ensures interoperability

Without stifling competition and innovation among users, to the benefit of the public both in terms of cost and service quality. The proposed system is acceptable to users. So the proposed system is operationally feasible.

#### **MODULES**

* Add Image
* Clip Image
* Construct Image
* Identification

A module is a small part of our project. This plays a very important role in the project and in coding concepts. In Software Engineering concept we treat it has a small part of a system but whereas in our programming language it is a small part of the program, which we also called as function in, some cases which constitute the main program.

Importance of modules in any software development side is we can easily understand what the system we are developing and what its main uses are. At the time of project, we may create many modules and finally we combine them to form a system.

**ADD IMAGE**

Add Image is a module that is considered with adding image along with the complete details of the person of whom we are taking image. In this we add Image by importing from the Internet and store them in our system and database. This module is mainly considered for adding details of the students like name, age, alias name, gender, location, state, Arrested Date, etc. At the time of the adding image, we give some student id to that particular person, so that it can be easily added to the database with any duplication of the data.

**CONSTRUCT IMAGE**

Based on the eyewitnesses we are going to construct the images. The witness will give us instruction by looking onto the screen on which there will be the parts of the images like eyes, hairs etc.

**IDENTIFICATION**

This module contains the interface to take the image from above module and it compares or searches with the images already there in the database. If any image is matched then we identify him/her as the student else we add that new image again to the database.

**SYSTEM REQUIREMENT SPECIFICATION**

**SOFTWARE REQUIREMENTS:**

Operating System : Windows 7 and Above

Database Server : MySQL or WAMP server

Programming Language : Python

**HARDWARE REQUIREMENTS:**

Processor : Pentium IV

Clock Speed : 2.86GHZ Processor

Hard disk : 2GB

RAM : 256MB

#### **OVERVIEW OF PYTHON**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

## **History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## **Python Features**

Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

## **Local Environment Setup**

Open a terminal window and type "python" to find out if it is already installed and which version is installed.

* Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
* Win 9x/NT/2000
* Macintosh (Intel, PPC, 68K)
* OS/2
* DOS (multiple versions)
* PalmOS
* Nokia mobile phones
* Windows CE
* Acorn/RISC OS
* BeOS
* Amiga
* VMS/OpenVMS
* QNX
* VxWorks
* Psion
* Python has also been ported to the Java and .NET virtual machines

## **Getting Python**

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org/>

You can download Python documentation from <https://www.python.org/doc/>. The documentation is available in HTML, PDF, and PostScript formats.

## **Installing Python**

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platforms −

### **Unix and Linux Installation**

Here are the simple steps to install Python on Unix/Linux machine.

* Open a Web browser and go to <https://www.python.org/downloads/>.
* Follow the link to download zipped source code available for Unix/Linux.
* Download and extract files.
* Editing the *Modules/Setup* file if you want to customize some options.
* run ./configure script
* make
* make install

This installs Python at standard location */usr/local/bin* and its libraries at */usr/local/lib/pythonXX* where XX is the version of Python.

### **Windows Installation**

Here are the steps to install Python on Windows machine.

* Open a Web browser and go to <https://www.python.org/downloads/>.
* Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version you need to install.
* To use this installer *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
* Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

## **Setting up PATH**

Programs and other executable files can be in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. This variable contains information available to the command shell and other programs.

The **path** variable is named as PATH in Unix or Path in Windows (Unix is case sensitive; Windows is not).

In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you must add the Python directory to your path.

## **Setting path at Unix/Linux**

To add the Python directory to the path for a particular session in Unix −

* **In the csh shell** − type setenv PATH "$PATH:/usr/local/bin/python" and press Enter.
* **In the bash shell (Linux)** − type export PATH="$PATH:/usr/local/bin/python" and press Enter.
* **In the sh or ksh shell** − type PATH="$PATH:/usr/local/bin/python" and press Enter.
* **Note** − /usr/local/bin/python is the path of the Python directory

## **Setting path at Windows**

To add the Python directory to the path for a particular session in Windows −

**At the command prompt** − type path %path%;C:\Python and press Enter.

**Note** − C:\Python is the path of the Python directory

# 

# **MySQL – Introduction**

## **What is a Database?**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as **Foreign Keys**.

A **Relational Database Management System (RDBMS)** is a software that −

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

## **RDBMS Terminology**

Before we proceed to explain the MySQL database system, let us revise a few definitions related to the database.

* **Database** − A database is a collection of tables, with related data.
* **Table** − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* **Column** − One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row** − A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy** − Storing data twice, redundantly to make the system faster.
* **Primary Key** − A primary key is unique. A key value cannot occur twice in one table. With a key, you can only find one row.
* **Foreign Key** − A foreign key is the linking pin between two tables.
* **Compound Key** − A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index** − An index in a database resembles an index at the back of a book.
* **Referential Integrity** − Referential Integrity makes sure that a foreign key value always points to an existing row.

## **MySQL Database**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## **Before You Begin**

Before you begin this tutorial, you should have a basic knowledge of the information covered in our PHP and HTML tutorials.

This tutorial focuses heavily on using MySQL in a PHP environment. Many examples given in this tutorial will be useful for PHP Programmers.

All downloads for MySQL are located at MySQL Downloads. Pick the version number of MySQL Community Server which is required along with the platform you will be running it on.

Installing MySQL on Linux/UNIX

The recommended way to install MySQL on a Linux system is via RPM. MySQL AB makes the following RPMs available for download on its website −

MySQL − The MySQL database server manages the databases and tables, controls user access and processes the SQL queries.

MySQL-client − MySQL client programs, which make it possible to connect to and interact with the server.

MySQL-devel − Libraries and header files that come in handy when compiling other programs that use MySQL.

MySQL-shared − Shared libraries for the MySQL client.

MySQL-bench − Benchmark and performance testing tools for the MySQL database server.

The MySQL RPMs listed here are all built on a SuSE Linux system, but they will usually work on other Linux variants with no difficulty.

Now, you will need to adhere to the steps given below, to proceed with the installation −

Login to the system using the root user.

Switch to the directory containing the RPMs.

Install the MySQL database server by executing the following command. Remember to replace the filename in italics with the file name of your RPM.

[root@host]# rpm -i MySQL-5.0.9-0.i386.rpm

The above command takes care of installing the MySQL server, creating a user of MySQL, creating necessary configuration and starting the MySQL server automatically.

You can find all the MySQL related binaries in /usr/bin and /usr/sbin. All the tables and databases will be created in the /var/lib/mysql directory.

The following code box has an optional but recommended step to install the remaining RPMs in the same manner −

[root@host]# rpm -i MySQL-client-5.0.9-0.i386.rpm

[root@host]# rpm -i MySQL-devel-5.0.9-0.i386.rpm

[root@host]# rpm -i MySQL-shared-5.0.9-0.i386.rpm

[root@host]# rpm -i MySQL-bench-5.0.9-0.i386.rpm

Installing MySQL on Windows

The default installation on any version of Windows is now much easier than it used to be, as MySQL now comes neatly packaged with an installer. Simply download the installer package, unzip it anywhere and run the setup.exe file.

The default installer setup.exe will walk you through the trivial process and by default will install everything under C:\mysql.

Test the server by firing it up from the command prompt the first time. Go to the location of the MySQL server which is probably C:\mysql\bin, and type −

mysqld.exe --console

NOTE − If you are on NT, then you will have to use mysqld-nt.exe instead of mysqld.exe

If all went well, you will see some messages about startup and inorb. If not, you may have a permissions issue. Make sure that the directory that holds your data is accessible to whatever user (probably MySQL) the database processes run under.

MySQL will not add itself to the start menu, and there is no particularly nice GUI way to stop the server either. Therefore, if you tend to start the server by double clicking the MySQL executable, you should remember to halt the process by hand by using mysqladmin, Task List, Task Manager, or other Windows-specific means.

Verifying MySQL Installation

After MySQL, has been successfully installed, the base tables have been initialized and the server has been started: you can verify that everything is working as it should be via some simple tests.

Use the mysqladmin Utility to Obtain Server Status

Use mysqladmin binary to check the server version. This binary would be available in /usr/bin on linux and in C:\mysql\bin on windows.

[root@host]# mysqladmin --version

It will produce the following result on Linux. It may vary depending on your installation −

mysqladmin Ver 8.23 Distrib 5.0.9-0, for redhat-linux-gnu on i386

If you do not get such a message, then there may be some problem in your installation and you would need some help to fix it.

Execute simple SQL commands using the MySQL Client

You can connect to your MySQL server through the MySQL client and by using the mysql command. At this moment, you do not need to give any password as by default it will be set as blank.

You can just use following command −

[root@host]# mysql

It should be rewarded with a mysql> prompt. Now, you are connected to the MySQL server and you can execute all the SQL commands at the mysql> prompt as follows −

mysql> SHOW DATABASES;

+----------+

| Database |

+----------+

| mysql |

| test |

+----------+

2 rows in set (0.13 sec)

Post-installation Steps

MySQL ships with a blank password for the root MySQL user. As soon as you have successfully installed the database and the client, you need to set a root password as given in the following code block −

[root@host]# mysqladmin -u root password "new\_password";

Now to make a connection to your MySQL server, you would have to use the following command −

[root@host]# mysql -u root -p

Enter password: \*\*\*\*\*\*\*

UNIX users will also want to put your MySQL directory in your PATH, so you won't have to keep typing out the full path everytime you want to use the command-line client.

For bash, it would be something like −

export PATH = $PATH:/usr/bin:/usr/sbin

Running MySQL at Boot Time

If you want to run the MySQL server at boot time, then make sure you have the following entry in the /etc/rc.local file.

/etc/init.d/mysqld start

## **Running and Shutting down MySQL Server**

First check if your MySQL server is running or not. You can use the following command to check it −

ps -ef | grep mysqld

If your MySql is running, then you will see **mysqld** process listed out in your result. If server is not running, then you can start it by using the following command −

root@host# cd /usr/bin

./safe\_mysqld &

Now, if you want to shut down an already running MySQL server, then you can do it by using the following command −

root@host# cd /usr/bin

./mysqladmin -u root -p shutdown

Enter password: \*\*\*\*\*\*

## **Setting Up a MySQL User Account**

For adding a new user to MySQL, you just need to add a new entry to the **user** table in the database **mysql**.

The following program is an example of adding a new user **guest** with SELECT, INSERT and UPDATE privileges with the password **guest123;** the SQL query is −

root@host# mysql -u root -p

Enter password:\*\*\*\*\*\*\*

mysql> use mysql;

Database changed

mysql> INSERT INTO user

(host, user, password,

select\_priv, insert\_priv, update\_priv)

VALUES ('localhost', 'guest',

PASSWORD('guest123'), 'Y', 'Y', 'Y');

Query OK, 1 row affected (0.20 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 1 row affected (0.01 sec)

mysql> SELECT host, user, password FROM user WHERE user = 'guest';

+-----------+---------+------------------+

| host | user | password |

+-----------+---------+------------------+

| localhost | guest | 6f8c114b58f2ce9e |

+-----------+---------+------------------+

1 row in set (0.00 sec)

When adding a new user, remember to encrypt the new password using PASSWORD() function provided by MySQL. As you can see in the above example, the password mypass is encrypted to 6f8c114b58f2ce9e.

Notice the FLUSH PRIVILEGES statement. This tells the server to reload the grant tables. If you don't use it, then you won't be able to connect to MySQL using the new user account at least until the server is rebooted.

You can also specify other privileges to a new user by setting the values of following columns in user table to 'Y' when executing the INSERT query or you can update them later using UPDATE query.

* Select\_priv
* Insert\_priv
* Update\_priv
* Delete\_priv
* Create\_priv
* Drop\_priv
* Reload\_priv
* Shutdown\_priv
* Process\_priv
* File\_priv
* Grant\_priv
* References\_priv
* Index\_priv
* Alter\_priv

Another way of adding user account is by using GRANT SQL command. The following example will add user **zara** with password **zara123** for a particular database, which is named as **TUTORIALS**.

root@host# mysql -u root -p password;

Enter password:\*\*\*\*\*\*\*

mysql> use mysql;

Database changed

mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP

-> ON TUTORIALS.\*

-> TO 'zara'@'localhost'

-> IDENTIFIED BY 'zara123';

This will also create an entry in the MySQL database table called as **user**.

**NOTE** − MySQL does not terminate a command until you give a semi colon (;) at the end of the SQL command.

### **The /etc/my.cnf File Configuration**

In most of the cases, you should not touch this file. By default, it will have the following entries −

[mysqld]

datadir = /var/lib/mysql

socket = /var/lib/mysql/mysql.sock

[mysql.server]

user = mysql

basedir = /var/lib

[safe\_mysqld]

err-log = /var/log/mysqld.log

pid-file = /var/run/mysqld/mysqld.pid

Here, you can specify a different directory for the error log, otherwise you should not change any entry in this table.

**DATA FLOW DIAGRAMS:**

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

**CONTEXT DIAGRAM:**

The top-level diagram is often called a “*context diagram”*. It contains a single process, but it plays a very important role in studying the current system. The context diagram defines the system that will be studied in the sense that it determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. It represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows respectively.

**TYPES OF DATA FLOW DIAGRAMS:**

Data Flow Diagrams are of two types as follows:

* + 1. Physical DFD
    2. Logical DFD

1. **PHYSICAL DFD:**

Structured analysis states that the current system should be first understand correctly. The physical DFD is the model of the current system and is used to ensurethat the current system has been clearly understood. Physical DFDs shows actual devices, departments, and people etc., involved in the current system

**2. LOGICAL DFD:**

Logical DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system’s structure charts.

**BASIC NOTATION:**

The Basic Notation used to create a DFD’s are as follows:

**DATAFLOW:** Data move in a specific direction from an origin to a destination.

**PROCESS:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.

**SOURCE:** External sources or destination of data, which may be People, programs, organizations or other entities.

**DATA STORE:** Here data are stored or referenced by a process in

theSystem

**DESIGN:**

Design is the first step in moving from problem domain to the solution domain. Design is essentially the bridge between requirements specification and the final solution.

# FACE IDENTIFICATION SYSTEM

STUDENT FACE

OPERATOR

# **Context Flow Diagram**

**Description:** Context Flow Diagram gives us the complete details about the inputs and outputs for a given system. In the above system the main task is to identify a student face. So, the operator and eyewitness are the inputs to our system and student face is desired output.

**LOGIN PROCESS**

LOGIN

ERROR IN

INPUT

SCREEN

User Id

Password

**Level-1**

**Description:** The inputs to the process are User Id and Password given by the developer to allow the software available for the user environment. After giving the inputs the code checks whether the entered ones are valid are not. It displays screen if match occurs otherwise error message if they are not matched.

# **MAIN SCREEN PROCESS**

OPERATOR

ADD IMAGE

SEARCH IMAGE

CLIP IMAGE

CONSTRUCT IMAGE

# **Level -2**

**Description:** This process mainly explains the different screens that are available for the operator. Here the selection of the screen depends on the operator and he can select whatever screen he wants. The different screens that are available are Add Image, Show or Search Image, Clip Image and Construct Image.

# ADD IMAGE PROCESS

OPERATOR

DATABASE

DATA IS ADDED

ERROR

# **Level-3**

**Description:** This process clearly illustrates adding the details of the student such as name, alias name, age, gender, location, address, state and city along with his photo. These details are being added to the database, if any error is generated then it will be prompted to the operator otherwise we get message data is successfully added.

EYES

NOSE

HAIR

FOREHEAD

DATABASE

FACE

DATABASE

FACE

**Level-4**

**Description:** This is used for clipping the image into different slices say eyes, forehead, lips, hair and nose. The input for this is face which is divided into some slices which are stored in the database. Even though the image is divided into slices, the original image remains as it is.

# **UPDATE PROCESS**

OPERATOR

DATA UPDATED

DATABASE

# **Level-5**

**Description:** Update process is mainly used for updating or modifying the details of the student or person. This is used in situation where we have entered the details incorrectly or we want to add some new details.

**Level-6**

**Description:** Based on the instruction given by the eyewitnesses, the operator brings the clips of the images from the database and then goes for the construction of the image based on those clips.

# **COMPARISON PROCESS**

**DATABASE**

FACE

RESULT

**Level-7**

**Description:** The face that is constructed in the above process is sent to the comparison process where it searches the image in the database.

#### **DATABASE TABLES**

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Size |
| Student Id  First Name  Last Name  Alias Name  D.O.B  Age  Gender  Address  City  State  Arrested Date  Crime Involved In  Photo | Number  Varchar2  Varchar2  Varchar2  Date  Number  Varchar2  Varchar2  Varchar2  Varchar2  Date  Varchar2  Varchar2 | 5 (Primary Key)  15  15  15  5  15  15  15  15  15  15 |

**LOGIN SCREEN**

**LOGIN PAGE CODE**

from tkinter import \*

from PIL import ImageTk,Image

from register import reg

import mysql.connector

from tkinter import messagebox

def next():

reg()

class loginpage:

def \_\_init\_\_(self, window):

self.window = window

window.geometry('1166x718')

window.title('Login')

window.state('zoomed')

window.resizable(0,0)

#------------------------background image-------------------------

bg\_frame = Image.open(r"C:\Users\priya\Desktop\biotmetric\attendence.jpg")

photo = ImageTk.PhotoImage(bg\_frame)

bg\_panel = Label(window, image=photo)

bg\_panel.image = photo

bg\_panel.pack(fill= 'both' , expand='yes')

#------------------------login frame-------------------------

lgn\_frame = Frame(window , bg= '#040405' , width= 450 , height=650)

lgn\_frame.place(x=900,y=65)

heading = Label(lgn\_frame, text = 'WELCOME' , font=('yu gothic ui' , 25 , 'bold') , bg= '#040405' , fg = 'white')

heading.place(x=80, y=30 , width= 300 , height= 30)

#----------------signin image-------------------------

sign\_in\_image= Image.open(r"C:\Users\priya\Desktop\biotmetric\login.jpg")

sign\_in\_image= sign\_in\_image.resize((100,100))

photo = ImageTk.PhotoImage(sign\_in\_image)

sign\_in\_image\_label = Label(lgn\_frame , image = photo , bg='#040405')

sign\_in\_image\_label.image = photo

sign\_in\_image\_label.place(x=180, y=100)

sign\_in\_label = Label(lgn\_frame, text= 'Sign In' , bg='#040405' , fg='white' , font=('yu gothic ui' , 17 , 'bold'))

sign\_in\_label.place(x=195 , y=210)

#---------------username--------------------------

username\_label = Label(lgn\_frame, text='Username' , font= ('yu gothic ui' , 13 , 'bold') , bg='#040405' , fg='#4f4e4d')

username\_label.place(x=70, y=300)

username\_entry = Entry(lgn\_frame, highlightthickness=0 , relief= FLAT , bg='#040405' , fg='#6b6a69' , font=('yo gothic ui' , 12,'bold'))

username\_entry.place(x=110 , y=335 , width = 270)

username\_line = Canvas(lgn\_frame , width = 300 , height = 2.0 , bg='#bdb9b1' , highlightthickness=0 )

username\_line.place(x=70, y=359)

#--------------------username icon---------------------

username\_icon= Image.open(r"C:\Users\priya\Desktop\biotmetric\images\username\_icon.png")

photo = ImageTk.PhotoImage(username\_icon)

username\_icon\_label = Label(lgn\_frame , image = photo , bg='#040405')

username\_icon\_label.image = photo

username\_icon\_label.place(x=70, y=330)

#----------------------------password---------------

password\_label = Label(lgn\_frame, text='Password' , font= ('yu gothic ui' , 13 , 'bold') , bg='#040405' , fg='#4f4e4d')

password\_label.place(x=70, y=380)

password\_entry = Entry(lgn\_frame, highlightthickness=0 , relief= FLAT , bg='#040405' , fg='#6b6a69' , font=('yo gothic ui' , 12,'bold'))

password\_entry.place(x=110 , y=415 , width = 270)

password\_line = Canvas(lgn\_frame , width = 300 , height = 2.0 , bg='#bdb9b1' , highlightthickness=0 )

password\_line.place(x=70, y=439)

#--------------------password icon---------------------

password\_icon= Image.open(r"C:\Users\priya\Desktop\biotmetric\images\password\_icon.png")

photo = ImageTk.PhotoImage(password\_icon)

password\_icon\_label = Label(lgn\_frame , image = photo , bg='#040405')

password\_icon\_label.image = photo

password\_icon\_label.place(x=70, y=410)

def submit():

if username\_entry.get() =="" or password\_entry.get() =="":

messagebox.showerror("Error","All Field Required !!!")

elif username\_entry.get() =="priyanshu" and password\_entry.get() =="9110020949":

messagebox.showinfo("Success","Welcome Priyanshu !!!")

else:

messagebox.showerror("Invalid","Invalid Username & Password !!!")

#---------------------login button----------------------

lgn\_button= Image.open(r"C:\Users\priya\Desktop\biotmetric\images\btn1.png")

photo = ImageTk.PhotoImage(lgn\_button)

lgn\_button\_label = Label(lgn\_frame , image = photo , bg='#040405', bd=0 )

lgn\_button\_label.image = photo

lgn\_button\_label.place(x=70, y=450)

login = Button(lgn\_button\_label ,text='LOGIN' , font= ('yu gothic ui' , 13 , 'bold') , bg='#3047ff' , fg='white' , activebackground='#3047ff' , cursor='hand2' , width=25 , bd=0 , command= submit)

login.place(x=20, y=10)

#-------------------------forgot password---------

forgot\_btn = Button(lgn\_frame ,text='Forgot Password ?' , font= ('yu gothic ui' , 13 , 'bold underline') , bg='#040405' , fg='white' , activebackground='#040405' , cursor='hand2' , width=25 , bd=0)

forgot\_btn.place(x=95, y=510)

#-------------------signup-------------------------

signup\_label = Label(lgn\_frame, text='No account yet ?' , font= ('yu gothic ui' , 9 , 'bold') , bg='#040405' , fg='white')

signup\_label.place(x=70, y=560)

signup\_btn = Image.open(r"C:\Users\priya\Desktop\biotmetric\images\register.png")

photo = ImageTk.PhotoImage(signup\_btn)

signup\_btn\_label = Button(lgn\_frame , image = photo , bg='#040405' , cursor='hand2' , activebackground='#040405' , bd=0 , command=next)

signup\_btn\_label.image = photo

signup\_btn\_label.place(x=170, y=550)

#============show hide password-------------------

def show():

hide\_button = Button(lgn\_frame , image=photo , bg='white', activebackground='white' , cursor='hand2' , bd=0 , command = hide)

hide\_button.image = photo

hide\_button.place(x=370, y=415)

password\_entry.config(show='')

def hide():

show\_button = Button(lgn\_frame , image=photo1 , bg='white', activebackground='white' , cursor='hand2' , bd=0 , command = show)

show\_button.image = photo1

show\_button.place(x=370, y=415)

password\_entry.config(show='\*')

show\_image= Image.open(r"C:\Users\priya\Desktop\biotmetric\images\show.png")

photo1 = ImageTk.PhotoImage(show\_image)

show\_button = Button(lgn\_frame , image=photo1 , bg='white', activebackground='white' , cursor='hand2' , bd=0 , command= show)

show\_button.image = photo1

show\_button.place(x=370, y=415)

hide\_image= Image.open(r"C:\Users\priya\Desktop\biotmetric\images\hide.png")

photo = ImageTk.PhotoImage(hide\_image)

#----------------------slider frame-------------------------------

slider\_frame1 = Frame(window , bg= '#FFCC00' , width= 775, height=580)

slider\_frame1.place(x=48,y=95)

slider\_frame = Frame(window , bg= '#040405' , width= 750 , height=550)

slider\_frame.place(x=60,y=110)

img1 = ImageTk.PhotoImage(file=r"C:\Users\priya\Desktop\biotmetric\l1.jpg")

img2 = ImageTk.PhotoImage(file=r"C:\Users\priya\Desktop\biotmetric\l2.jpg")

img3 = ImageTk.PhotoImage(file=r"C:\Users\priya\Desktop\biotmetric\l3.jpg")

slider\_label = Label(slider\_frame , bd=0)

slider\_label.place(x=0,y=0)

def slider\_func():

global x

if x==4:

x=1

if x==1:

slider\_label.config(image=img1)

elif x==2:

slider\_label.config(image=img2)

elif x==3:

slider\_label.config(image=img3)

x=x+1

slider\_frame.after(2000,slider\_func)

slider\_func()

x=1

def page():

window = Tk()

loginpage(window)

window.mainloop()

if \_\_name\_\_== '\_\_main\_\_':

page()

**REGISTRATION PAGE CODE**

from tkinter import \*

from PIL import ImageTk,Image

import mysql.connector

from tkinter import messagebox

def reg():

window = Toplevel()

window.geometry('1166x718')

window.title('REGISTER')

window.state('zoomed')

window.resizable(0,0)

#--------------------bg ----------------------

bg\_frame = Image.open(r"C:\Users\priya\Desktop\biotmetric\reg.jpeg")

bg\_frame = bg\_frame.resize((1550,900))

photo = ImageTk.PhotoImage(bg\_frame)

bg\_panel = Label(window, image=photo)

bg\_panel.image = photo

bg\_panel.pack(fill= 'both' , expand='yes')

#------------------------reg frame-------------------------

reg\_frame = Frame(window , width= 980 , height=430 , bg='#040405')

reg\_frame.place(x=200,y=400)

img= Image.open(r"C:\Users\priya\Desktop\biotmetric\frame.jpeg")

img= img.resize((450,430))

photo = ImageTk.PhotoImage(img)

img\_label = Label(reg\_frame , image = photo , bg='#040405')

img\_label.image = photo

img\_label.place(x=0, y=0)

heading = Label(reg\_frame, text = 'SIGN UP' , font=('yu gothic ui' , 25 , 'bold') , fg = '#FFCC00' , bg='#040405')

heading.place(x=670, y=8)

l1=Label(reg\_frame,text="First Name",font=('yu gothic ui' , 16 , 'bold'),foreground="#B19CD9",width=20 , bg='#040405')

l1.place(x=460,y=70)

f1=Entry(reg\_frame,font="bold")

f1.place(x=700,y=75)

l2=Label(reg\_frame,text="Last Name",font=('yu gothic ui' , 16 , 'bold'),foreground="#B19CD9",width=20,bg='#040405')

l2.place(x=460,y= 110)

f2=Entry(reg\_frame,font="bold")

f2.place(x=700,y=115)

l3=Label(reg\_frame,text="Contact",font=('yu gothic ui' , 16 , 'bold'),foreground="#B19CD9",width=18,bg='#040405')

l3.place(x=459,y=150)

f3=Entry(reg\_frame,font="bold")

f3.place(x=700,y=155)

l4=Label(reg\_frame,text="E-Mail",font=('yu gothic ui' , 16 , 'bold'),foreground="#B19CD9",width=17,bg='#040405')

l4.place(x=458,y=190)

f4=Entry(reg\_frame,font="bold")

f4.place(x=700,y=195)

l5=Label(reg\_frame,text="Password",font=('yu gothic ui' , 16 , 'bold'),foreground="#B19CD9",width=20,bg='#040405')

l5.place(x=456,y=230)

f5=Entry(reg\_frame,font="bold")

f5.place(x=700,y=235)

l6=Label(reg\_frame,text="Confirm",font=('yu gothic ui' , 16 , 'bold'),foreground="#B19CD9",width=18,bg='#040405')

l6.place(x=460,y=270)

f6=Entry(reg\_frame,font="bold")

f6.place(x=700,y=275)

check=IntVar()

checkbtn=Checkbutton(reg\_frame,variable=check,text="I agree the Terms & Conditions .",font=('times new roman' , 15 ),foreground='grey', activebackground='#040405',width=36,bg='#040405',onvalue=1,offvalue=0)

checkbtn.place(x=460,y=320)

def save():

if f1.get() =="" or f2.get() =="" or f3.get()=="" or f4.get()=="":

messagebox.showerror("Error","All Field Required !!!")

elif f5.get()!=f6.get():

messagebox.showerror("Error","Both must be same")

elif check.get()==0:

messagebox.showerror("Error","Please agree our Terms & Condition")

else:

messagebox.showinfo("Success","You are Register !!!")

b1=Button(reg\_frame,text="Register Now",font=('yu gothic ui' , 16 , 'bold'),background="#B19CD9", bd =2 , foreground="white" , command = save)

b1.place(x=640,y=370,width=200)

**STUDENT ATTENDENCE RECORD PAGE**

**CODE**

from tkinter import \*

from PIL import ImageTk,Image

from tkinter import ttk

import mysql.connector

from tkinter import messagebox

class student:

def \_\_init\_\_(self, window):

self.window = window

window.geometry('1166x718')

window.title('Home Page')

window.state('zoomed')

img1=Image.open(r"C:\Users\priya\Desktop\biotmetric\img2.jpg")

img1=img1.resize((1537,200))

photo1=ImageTk.PhotoImage(img1)

l1=Label(window,image=photo1)

l1.image = photo1

l1.place(x=0,y=0)

title\_lb1=Label(window,text="ATTENDENCE MANAGEMENT SYSTEM",font=('times new roman',35 ,'bold'), bg='lightblue' , fg='red')

title\_lb1.place(x=0,y=202,width=1537,height=55)

img2=Image.open(r"C:\Users\priya\Desktop\biotmetric\img1.jpg")

img2=img2.resize((1537,600))

photo2=ImageTk.PhotoImage(img2)

l2=Label(window,image=photo2)

l2.image = photo2

l2.place(x=0,y=255)

main\_frame=Frame(l2, bd=2)

main\_frame.place(x=17,y=2,width=1503,height=532)

#-----------left label frame---------------

left\_frame=LabelFrame(main\_frame,bd=2,relief=RIDGE,text="Students Details" , font=('times new roman', 12,'bold'))

left\_frame.place(x=10,y=1,width=660,height=527)

img\_left=Image.open(r"C:\Users\priya\Desktop\biotmetric\img1.jpg")

img\_left=img\_left.resize((650,200),Image.ANTIALIAS)

photo\_left=ImageTk.PhotoImage(img\_left)

lb\_left=Label(left\_frame,image=photo\_left)

lb\_left.image = photo\_left

lb\_left.place(x=0,y=0)

#-----current\_course----------------------

current\_course\_frame=LabelFrame(left\_frame,relief=RIDGE,text="Current Course Information" , font=('times new roman', 12,'bold'))

current\_course\_frame.place(x=5,y=200,width=645,height=92)

#-----department----------------------

dep\_label=Label(current\_course\_frame,text='Department',font=('times new roman', 12,'bold'))

dep\_label.grid(row=0,column=0,padx=10)

dep\_combo=ttk.Combobox(current\_course\_frame,font=('times new roman', 12,'bold'),width=17,state="readonly")

dep\_combo["values"]=("Select Department","B.Sc.","B.Com","B.A.","BCA","BIT")

dep\_combo.current(0)

dep\_combo.grid(row=0,column=1,pady=5,sticky=W)

#-----current\_course----------------------

course\_label=Label(current\_course\_frame,text='Course',font=('times new roman', 12,'bold'))

course\_label.grid(row=0,column=3,padx=10,sticky=W)

course\_combo=ttk.Combobox(current\_course\_frame,font=('times new roman', 12,'bold'),width=17,state="readonly")

course\_combo["values"]=("Select Course","C1","C2","C3","C3","C4","C5","C6","C7","C8","C9","C10","C11","C12","C13","C14","DSE1","DSE2","DSE3","DSE4")

course\_combo.current(0)

course\_combo.grid(row=0,column=4,padx=2,pady=5,sticky=W)

#----------------year-----------------

year\_label=Label(current\_course\_frame,text='Year',font=('times new roman', 12,'bold'))

year\_label.grid(row=1,column=0,padx=10,sticky=W)

year\_combo=ttk.Combobox(current\_course\_frame,font=('times new roman', 12,'bold'),width=17,state="readonly")

year\_combo["values"]=("Select Year","2019-2020","2020-2021","2021-2022","2022-2023","2023-2024")

year\_combo.current(0)

year\_combo.grid(row=1,column=1,padx=1,pady=5,sticky=W)

#------------------semester--------------------

sem\_label=Label(current\_course\_frame,text='Semester',font=('times new roman', 12,'bold'))

sem\_label.grid(row=1,column=3,padx=10,sticky=W)

sem\_combo=ttk.Combobox(current\_course\_frame,font=('times new roman', 12,'bold'),width=17,state="readonly")

sem\_combo["values"]=("Select Semester","Sem-I","Sem-II","Sem-III","Sem-IV","Sem-V","Sem-VI")

sem\_combo.current(0)

sem\_combo.grid(row=1,column=4,padx=2,pady=5,sticky=W)

#-----------------student information-----------------

class\_student\_frame=LabelFrame(left\_frame,bd=2,text="Class Student Information",relief=RIDGE,font=('times new roman', 12,'bold'))

class\_student\_frame.place(x=5,y=292,width=645,height=210)

student\_label=Label(class\_student\_frame,text='Student ID:',font=('times new roman', 12,'bold'))

student\_label.grid(row=0,column=0,padx=10,sticky=W)

student\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

student\_entry.grid(row=0,column=1,padx=0,sticky=W)

class\_div\_label=Label(class\_student\_frame,text='Class Division:',font=('times new roman', 12,'bold'))

class\_div\_label.grid(row=0,column=2,padx=5,sticky=W)

class\_div\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

class\_div\_entry.grid(row=0,column=3,sticky=W)

roll\_label=Label(class\_student\_frame,text='Roll No:',font=('times new roman', 12,'bold'))

roll\_label.grid(row=1,column=0,padx=10,sticky=W)

roll\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

roll\_entry.grid(row=1,column=1,padx=0,pady=5,sticky=W)

gender\_label=Label(class\_student\_frame,text='Gender:',font=('times new roman', 12,'bold'))

gender\_label.grid(row=1,column=2,padx=5,sticky=W)

gender\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

gender\_entry.grid(row=1,column=3,padx=0,pady=5,sticky=W)

email\_label=Label(class\_student\_frame,text='E-Mail:',font=('times new roman', 12,'bold'))

email\_label.grid(row=2,column=0,padx=10,sticky=W)

email\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

email\_entry.grid(row=2,column=1,padx=0,pady=5,sticky=W)

phone\_label=Label(class\_student\_frame,text='Phone No:',font=('times new roman', 12,'bold'))

phone\_label.grid(row=2,column=2,padx=5,sticky=W)

phone\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

phone\_entry.grid(row=2,column=3,padx=0,pady=5,sticky=W)

address\_label=Label(class\_student\_frame,text='Address:',font=('times new roman', 12,'bold'))

address\_label.grid(row=3,column=0,padx=10,sticky=W)

address\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

address\_entry.grid(row=3,column=1,padx=0,pady=5,sticky=W)

teacher\_label=Label(class\_student\_frame,text='Teacher Name:',font=('times new roman', 12,'bold'))

teacher\_label.grid(row=3,column=2,padx=5,sticky=W)

teacher\_entry=ttk.Entry(class\_student\_frame,width=20,font=('times new roman', 12,'bold'))

teacher\_entry.grid(row=3,column=3,padx=0,pady=5,sticky=W)

radiobtn1=ttk.Radiobutton(class\_student\_frame,text="Take Photo Sample",value="Yes")

radiobtn1.grid(row=6,column=0)

radiobtn2=ttk.Radiobutton(class\_student\_frame,text="No Photo Sample",value="Yes")

radiobtn2.grid(row=6,column=1)

take\_btn=Button(class\_student\_frame,text="Take Photo Sample",width=16,height=1,bg='lightgreen',font=('times new roman', 7,'bold'),cursor='hand2',fg='black')

take\_btn.grid(row=6,column=2,padx=9)

upload\_btn=Button(class\_student\_frame,text="Upload Photo Sample",width=18,height=1,bg='lightgreen',font=('times new roman', 7,'bold'),cursor='hand2',fg='black')

upload\_btn.grid(row=6,column=3,padx=9)

save\_btn=Button(class\_student\_frame,text="Save",width=15,height=1,bg='blue',cursor='hand2',font=('times new roman', 12,'bold'),fg='white')

save\_btn.grid(row=7,column=0,padx=9)

upload\_btn=Button(class\_student\_frame,text="Upload",width=15,height=1,bg='blue',cursor='hand2',font=('times new roman', 12,'bold'),fg='white')

upload\_btn.grid(row=7,column=1)

reset\_btn=Button(class\_student\_frame,text="Reset",width=15,height=1,bg='blue',cursor='hand2',font=('times new roman', 12,'bold'),fg='white')

reset\_btn.grid(row=7,column=3)

delete\_btn=Button(class\_student\_frame,text="Delete",width=15,height=1,bg='blue',cursor='hand2',font=('times new roman', 12,'bold'),fg='white')

delete\_btn.grid(row=7,column=2)

#----------right frame-----------

right\_frame=LabelFrame(main\_frame,bd=2,relief=RIDGE,text="Students Details" , font=('times new roman', 12,'bold'))

right\_frame.place(x=685,y=1,width=800,height=527)

search\_frame=LabelFrame(right\_frame,relief=RIDGE,text="Search System" , font=('times new roman', 12,'bold'))

search\_frame.place(x=5,y=200,width=785,height=55)

search\_label=Label(search\_frame,text='Search By :',font=('times new roman', 12,'bold'),bg='red',fg='white')

search\_label.grid(row=0,column=0,padx=15)

search\_combo=ttk.Combobox(search\_frame,font=('times new roman', 12,'bold'),width=17,state="readonly")

search\_combo["values"]=("Select","Roll No","Phone No")

search\_combo.current(0)

search\_combo.grid(row=0,column=1,padx=15,sticky=W)

search\_entry=ttk.Entry(search\_frame,width=20,font=('times new roman', 12,'bold'))

search\_entry.grid(row=0,column=2,padx=15,sticky=W)

search\_btn=Button(search\_frame,text="Search",width=15,height=1,bg='blue',cursor='hand2',font=('times new roman', 10,'bold'),fg='white')

search\_btn.grid(row=0,column=3,padx=15)

showall\_btn=Button(search\_frame,text="Show All",width=15,height=1,bg='blue',cursor='hand2',font=('times new roman', 10,'bold'),fg='white')

showall\_btn.grid(row=0,column=4,padx=15)

#=======table frame-----------------------

table\_frame=Frame(right\_frame,bd=2,bg='white',relief=RIDGE)

table\_frame.place(x=5,y=260,width=785,height=200)

scroll\_x=ttk.Scrollbar(table\_frame,orient=HORIZONTAL)

scroll\_y=ttk.Scrollbar(table\_frame,orient=VERTICAL)

student\_table=ttk.Treeview(table\_frame,column=("dep","course","year","sem","id","name","div","roll","gender","phone","address","teacher","photo") , xscrollcommand=scroll\_x.set , yscrollcommand=scroll\_y.set)

scroll\_x.pack(side=BOTTOM,fill=X)

scroll\_y.pack(side=RIGHT,fill=Y)

scroll\_x.config(command=student\_table.xview)

scroll\_y.config(command=student\_table.yview)

student\_table.heading("dep",text="Department")

student\_table.heading("course",text="Course")

student\_table.heading("year",text="Year")

student\_table.heading("sem",text="Semester")

student\_table.heading("id",text="Student Id")

student\_table.heading("name",text="Name")

student\_table.heading("div",text="Division")

student\_table.heading("roll",text="Roll No")

student\_table.heading("gender",text="Gender")

student\_table.heading("phone",text="Phone No")

student\_table.heading("address",text="Address")

student\_table.heading("teacher",text="Teacher")

student\_table.heading("photo",text="PhotoSampleStatus")

student\_table["show"]="headings"

student\_table.column("dep",width=100)

student\_table.column("course",width=100)

student\_table.column("year",width=100)

student\_table.column("sem",width=100)

student\_table.column("id",width=100)

student\_table.column("name",width=100)

student\_table.column("div",width=100)

student\_table.column("roll",width=100)

student\_table.column("gender",width=100)

student\_table.column("phone",width=100)

student\_table.column("address",width=100)

student\_table.column("teacher",width=100)

student\_table.column("photo",width=150)

student\_table.pack(fill=BOTH,expand=1)

def page():

window = Tk()

student(window)

window.mainloop()

if \_\_name\_\_== '\_\_main\_\_':

page()

### **TESTING**

#### **TESTING PHASE**

The completion of a system is achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Though the programmer may have taken many precautions not to commit any mistakes that crop up during the execution stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any break downs, therefore a package can be rejected even at this stage. The testing phase involves the testing of the developed system using various kinds of data. An elaborated testing of data is prepared and a system is tested using the test data. While testing, errors are noted and corrections remade, the corrections are also noted for future use.

**SYSTEM TESTING**

Testing is a set of activities that can be planned in advance and conducted systematically. The proposed system is tested in parallel with the software that consists of its own phases of analysis, implementation, testing and maintenance. Following are the tests conducted on the system.

* **UNIT TESTING**

During the implementation of the system each module of the system was tested separately to uncover errors within its boundaries. User interface was used as a guide in the process.

* **MODULE TESTING**

A module is composed of various programs related to that module. Module testing is done to check the module functionality and interaction between units within a module.

It checks the functionality of each program with relation to other programs within the same module. It then tests the overall functionality of each module.

* **INTEGRATION TESTING**

Integration testing is a systematic technique for constructing the program structure while conducting tests to uncover errors associated with interfacing. The objective is to take unit-tested module and build a program structure that has been dictated by design.

* **ACCEPTANCE TESTING**

The software has been tested with the realistic data given by the client and produced fruitful results. The client

satisfying all the requirements specified by them has also developed the software within the time limitation specified. A demonstration has been given to the client and the end-user giving all the operational features.

**Unit Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no.** | **Screen** | **Input** | **Output** | **Remarks** |
| 1 | Login Page | User Id  Password | User validation | User will enter into Main Screen |
| 2 | Main Screen   * File * New | Student Details are entered | Details are stored in the database | New Menu is selected to enter new student details. |
| 3 | Main Screen   * File * Show Details | Display option is clicked. | Student Details are displayed | Display details menu is selected to get details from database. |
| 4 | Main Screen   * File * Exit | Exit Option is Clicked. | Screen will be exited | Screen will be shut down |
| 5 | Main Screen   * Edit * Clip Image | Student Image is clipped into different parts | The clips are stored in database | Clip image menu is selected to clip image and store them in database |
| 6 | Main Screen   * Edit * Update Details | Changes in the details of the students are entered | Details of the student are updated | Update details menu is selected to update the details of the students |
| 7 | Main Screen   * Identification * Construct Face | Different clips of students are selected and arranged in order | Face of the student is constructed | Construct face menu is to construct the student face from various clips stored in the database. |
| 8 | Main Screen   * Identification * Find Face | Show all students is checked | All students detail along with photo are displayed | Show more Student menu is selected to get the details of all students and more possible student |

**IMPLEMENTATION PHASE**

The implementation is the final and important phase. It involves User training, system testing and successful running of the developed system. The users test the developed system when changes are made according to the needs. The testing phase involves the testing of the developed system using various kinds of data. An elaborate testing of data is prepared and system is tested using the tests data.

Implementation is the stage where theoretical design turned into a working system. Implementation is planed carefully to propose system to avoid unanticipated problems. Many preparations involved before and during the implementation of proposed system. The system needed to be plugged in to the organization’s network then it could be accessed from anywhere, after a user login into the portal. The tasks that had to be done to implement the system were to create the database tables in the organization database domain. Then the administrator was granted his role so that the system could be accessed.

The next phase in the implementation was to educate the system. A demonstration of all the functions that can be carried out by the system was given to examination department person, who will make extensive use of the system.

**CONCLUSION**

The purpose of face recognition attendance management system is to identify students. In past years this process is carried out by humans. This process gives the exact image of the student but it is very difficult to identify the student details and also it requires much amount of human burden.

The main aim of our project is to overcome the drawbacks of human based system by using the machine-based face identification process. In this process we store the details of student into the database along with his photo or image. Then we make the image into different clips containing hair, forehead, eyes, nose, lips and chin and store these clips into the database. When any duplication occurs, we compare the details given by the eyewitness with the clips already stored in the database and we will identify the student. This project can be extended to adjust the gaps between the clips after construction of the image to be a perfect photograph using Image processing Techniques.

**Further enhancement (Suggestion)**

Some industries have been harder hit than others by the COVID-19 pandemic and this has meant that they have had to adapt much quicker and pivot to the use of new technologies in order to prepare for the world opening up again.

Events – both on a local and global scale – has been one of the most impacted industries as restrictions on numbers, social distancing, mandatory masks and other safety precautions have meant that organizers have had to rethink the way events are managed. Many events over the past 18 months have been conducted virtually, however, as we start to open up and international travel starts again, event organizers are turning to facial recognition as a way of managing access, check-in, visitor experience and safety.

[**NEC worked with the Ladies Professional Golf Association (LPGA)**](https://www.lpga.com/news/2017-nec-introduces-facial-recognition-software-at-ana-inspiration) at the 2017 ANA Inspiration at Mission Hills Country Club.  Credentialed media covering the LPGA major championship passed through NEC’s Neo Face Watch face recognition solution before being allowed access to the ANA Inspiration Media Center, enabling secure entry to the facility.

Event organizers are also turning to facial recognition used in conjunction with an app to allow people to register for an event. When attending the event, facial recognition can then be used for admittance and allocation of seats without having to produce a ticket.

The automobile industry is another that is investing in facial recognition technology. Last year, about $6 billion was lost to motor vehicle theft in the US alone. Obviously, there is a need for a new, reliable safety measure that would keep car owners at peace even when they’re not around their car. Face recognition is already helping to provide that extra layer of safety and help reduce thefts.

Face recognition in automobiles works on a simple and non-obtrusive principle. After a driver enrolls into the system, the system “remembers” them. Each time they enter the vehicle again, the system “recognizes” them and gives them access to predefined functionalities such as permission to start the car.

Car owners can also set up permissions or restrictions for other people such as family members. For example, they could set up certain restrictions on their children learning to drive such as a time or speed limit or deny access without an adult present. If an unauthorized person enters the car, the system can notify the owner or block the car from starting. This helps prevent theft and gives owners better control of their cars.

[**Hyundai is one of the companies leading the way**](https://tech.hyundaimotorgroup.com/article/in-car-biometric-technology-for-human-interaction/)when it comes to biometric integration. In 2018, they introduced the world’s first fingerprint system that locks the doors and starts the engine, through its Chinese Santa Fe.

Using iris recognition, their Driver State Warning System (DSW) is a feature that delivers a warning when the driver is not focusing on driving, and it takes a step forward from the current system that only recognizes the direction of the face or when the eyes are closed, providing a safer driving environment.

DSW’s facial recognition technology identifies drivers through facial features such as eyes, nose, mouth, and ears. In addition, the system analyses the pupil and facial movement, then combines with driving information such as the speed of the car and steering angle, to offer safer driving. It detects the risk of lane departure and intrusion caused by driver carelessness in advance and calls the driver’s attention with cluster warning lights, alarm sounds, and vibrations.

## **Summary**

Facial recognition is here to stay and rather than seeing facial recognition as a threat to our personal privacy, we should instead be embracing the many benefits that facial recognition provides.

Whilst there are isolated cases of facial recognition being used inappropriately, there are now thousands of use cases that show that, when deployed appropriately and with the consent of people using the software, facial recognition is helping to create a safer environment, providing outstanding security and enhancing customer experiences across a wide range of settings.

**BIBLIOGRAPHY**

**BOOKS REFERRED**

The following books were used extensively for the project development and implementation.

* Downey, Allen B. (May 2012). Think Python: How to Think Like a Computer Scientist (Version 1.6.6 ed.).
* Lutz, Mark (2013). Learning Python (5th ed.). O'Reilly Media.
* Pilgrim, Mark (2009). Dive into Python 3. Apress.
* Summerfield, Mark (2009). Programming in Python 3 (2nd ed.). Addison-Wesley Professional

**WEBSITES REFERRED**

**The following links were searched and exploited extensively for the project development and implementation.**

1. https://www.python.org/
2. <https://www.w3schools.com/python/python_intro.asp>
3. https://www.geeksforgeeks.org/python-programming-language/