

# Face Recognition Attendance System Using Python

## A Project Report

Submitted in partial fulfillment of the  
Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

By

Mrs. PAYAL JAYANT JADHAV

Seat Number:

Under the esteemed guidance of  
**Mrs. Neha R. Petkar**



**DEPARTMENT OF INFORMATION TECHNOLOGY  
I.C.S. COLLEGE OF ARTS, COMMERCES AND  
SCIENCE KHED**

(Affiliated to University of Mumbai)

**KHED, PIN CODE: 415 708**

**MAHARASHTRA**

**YEAR (2023-2024)**



## CERTIFICATE

This is to certify that the project entitled "**Face Recognition Attendance System Using Python**" By Miss Payal Jadhav, Exam Seat No: \_\_\_\_\_ In the partial fulfillment of the requirement of the award of degree **Bachelor of Science (Information Technology)** from University of Mumbai.

**Internal Guide**

**Coordinator**

**External Examiner**

**Date:**

**College Seal**

## **ABSTRACT**

The paper argues that maintaining regular attendance is crucial for student success, and traditional attendance management methods can be inefficient and time-consuming for teachers and administrators. For example, calling out student names or taking manual attendance on paper can take up valuable classroom time and can be prone to errors or manipulation .To address these issues, the paper suggests that a computer-based attendance management system using Computer Vision technology can be an effective solution.

Computer Vision involves the use of cameras, sensors, and algorithms to identify and analyze visual data, including images of individuals. In the context of attendance management, Computer Vision can be used to capture images of students during class and automatically recognize and mark their attendance using facial recognition technology. This approach can offer several advantages over traditional attendance methods.

Firstly, it can be faster and more accurate, reducing the time and effort needed to manage attendance manually. Secondly, it can provide real-time updates on attendance status, allowing teachers to track students who arrive late or leave early. Finally, it can generate reports on attendance patterns, allowing administrators to identify and address issues related to student attendance and engagement.

Overall, the paper highlights the potential benefits of using a computer-based attendance management system using Computer Vision, emphasizing its ability to streamline attendance management and improve student outcomes.

## ACKNOWLEDGEMENT

It's my great pleasure to take this opportunity and sincerely thanks all those, who have showed me the way to successful project and helped me a lot during the completion of my project. I greatly thank my Project Guide **Ms. Neha R. Petkar Mam** without the completion of this project couldn't have been possible.

My Sincere thanks to respect IC. Principal **Dr. A. J. Awati Mam** and Head of Computer Science Department **Dr. Sachin Bhosale Sir** for providing all the facilities including availability of Computer Lab. I am greatly thanks teaching and Non-teaching staff of Computer Science Department of **ICS College of Arts, Commerce and Science.**

Finally, I am thanking to my all friends for their encouragement and support throughout period of completion.

Yours sincerely

PAYAL J. JADHAV (TYIT)

## DECLARATION

I hereby declare that the project entitled, “**Face Recognition Attendance System Using Python**” done at **ICS College, Khed**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirement for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Miss. Payal J. Jadhav

# TABLE OF CONTENTS

<b>SR. NO.</b>	<b>TOPIC</b>	<b>Page No.</b>
	Abstract.....	I
	Acknowledgements.....	II
	Declaration.....	III
	Table of Contents.....	IV
	List of Figures.....	V
	List of Table.....	VI
1.	<b>Introduction.....</b>	<b>1</b>
	1.1 Face Recognition Attendance System .....	1
	1.2 Background .....	1
	1.3 Objectives.....	2
	1.4 Features.....	2
	1.5 Purpose and Scope .....	3
	1.5.1 Purpose .....	3
	1.5.2 Scope.....	4
2.	<b>System Analysis.....</b>	<b>5</b>
	2.1 Existing System.....	5
	2.2 Proposed System .....	6
	2.3 Requirement Analysis.....	7
	2.4 Hardware Requirements.....	8
	2.5 Software Requirements .....	8
	2.6 Justification of selection of Technology .....	8
3.	<b>System Design.....</b>	<b>9</b>
	3.1 Module Division.....	9
	3.2 Data Dictionary.....	10
	3.3 ER Diagrams.....	13
	3.4 DFD/UML Diagrams.....	14

<b>4.</b>	<b>Implementation and Testing .....</b>	<b>16</b>
4.1	Code .....	16
4.1.1	Login Page .....	16
4.1.2	Main Page .....	26
4.1.3	Face Recognition Page .....	33
4.1.4	Attendance Page .....	38
4.2	Testing Approach .....	47
4.2.1	Unit Testing .....	47
4.2.2	Integration Testing .....	48
<b>5.</b>	<b>Results and Discussions .....</b>	<b>49</b>
5.1	Login Page .....	49
5.2	Forget Password .....	49
5.3	Register Page .....	50
5.4	Main Page .....	50
5.5	Student Details .....	51
5.6	Train Data Set .....	52
5.7	Images Data Folder .....	52
5.8	Face Recognition .....	53
5.9	Attendance Page .....	53
5.10	Developer Page .....	54
5.11	ChatBot Page .....	54
5.12	Exit .....	55
5.13	Database .....	55
5.13.1	Register .....	55
5.13.2	Students Details .....	56
<b>6.</b>	<b>Conclusion and Future Work .....</b>	<b>57</b>
6.1	Conclusion .....	57
6.2	Future Work .....	57
<b>7.</b>	<b>Reference .....</b>	<b>58</b>

# List of Figures

<b>Fig. No.</b>	<b>Figures Name</b>	<b>Page No.</b>
1.3.1	Block Diagram of the General Framework.....	2
2.2.1	Project Outline.....	6
2.3.1	Haar Features.....	7
2.3.2	LBPH Algorithm.....	7
3.3.1	ER Diagram.....	13
3.4.1	Data Flow Diagram.....	14
3.4.2	Use Case Diagram.....	15
5.1	Login Page .....	49
5.2	Forget Password .....	49
5.3	Register Page .....	50
5.4	Main Page .....	50
5.5	Student Details .....	51
5.6	Train Data Set .....	52
5.7	Images Data Folder .....	52
5.8	Face Recognition .....	53
5.9	Attendance Page .....	53
5.10	Developer Page .....	54
5.11	ChatBot Page .....	54
5.12	Exit .....	55
5.13.1	Register .....	55
5.13.2	Students Details .....	56

# Introduction

## 1.1 Face Recognition Attendance System:

To maintain the attendance record with day-to-day activities is a challenging task. The conventional method of calling name of each student is time consuming and there is always a chance of proxy attendance. The following system is based on face recognition to maintain the attendance record of students. The daily attendance of students is recorded subject wise which is stored already by the administrator. As the time for corresponding subject arrives the system automatically starts taking snaps and then apply face detection and recognition technique to the given image and the recognize students are marked as present and their attendance update with corresponding time and subject id. We have used Python techniques to develop this system, histogram of oriented gradient method is used to detect faces in images and Python method is used to compute and compare feature facial of students to recognize them. Our system is capable to identify multiple faces in real time. The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

## 1.2 Background:

Traditional attendance tracking methods involving paper registers, ID cards, or biometric systems have limitations that hinder productivity and data accuracy. FRAS emerges as a solution designed to leverage the cutting-edge technology of facial recognition to revolutionize attendance management.

In today's fast-paced world, managing attendance records manually can be a cumbersome and error-prone task, especially in institutions, organizations, and businesses where large numbers of people need to be monitored. Traditional methods of attendance tracking using paper registers or card swiping systems can be inefficient and prone to fraud. To address these challenges, the Face Recognition Attendance System (FRAS) project has been developed. FRAS leverages cutting-edge

facial recognition technology to automate the attendance tracking process, making it more efficient, accurate, and secure.

### 1.3 Objectives:

To identify the student faces accurately. To mark the attendance automatically. To reduce the time and the efforts required for manual attendance to provide a valuable attentive system for both teacher and students. It provides flexibility and reduces the time loss. There will be no chance for a proxy.

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.



Fig 1.3.1 (Block Diagram of the General Framework)

### 1.4 Features:

Face Recognition Attendance System offers a range of powerful features to achieve these objectives:

- **Facial Recognition:** Utilizes state-of-the-art facial recognition technology to identify individuals accurately and swiftly.
- **Real-time Tracking:** Records attendance in real-time, providing immediate access to attendance data.

- **Biometric Verification:** Relies on unique facial features for identity verification, reducing the risk of impersonation.
- **User-Friendly Interface:** Boasts an intuitive and user-friendly interface for administrators and end-users alike.
- **Data Storage:** Securely stores attendance records in a protected database for future reference and analysis.
- **Reporting:** Generates comprehensive attendance reports for various time frames and user groups.
- **Notifications:** Sends notifications to relevant parties (e.g., students) regarding their attendance status.
- **Integration:** Can be seamlessly integrated with access control systems for enhanced security and streamlined operations.

## 1.4 Purpose and Scope:

### 1. Purpose:

The Face Recognition Attendance System (FRAS) project is initiated with the primary objective of revolutionizing traditional attendance management methods by harnessing the power of facial recognition technology. The key purposes of this project are as follows:

- **Efficiency Enhancement:** To streamline and automate the attendance tracking process in various sectors, reducing the administrative burden associated with manual attendance recording.
- **Accuracy Improvement:** To ensure the accuracy and reliability of attendance records by utilizing advanced facial recognition algorithms, thereby eliminating errors and fraudulent practices like buddy punching.
- **Security Enhancement:** To enhance security by storing biometric data in the form of facial features, making it difficult for unauthorized individuals to record attendance on behalf of others.
- **Resource Optimization:** To save valuable time and resources for both institutions and individuals by simplifying and expediting the attendance tracking process.
- **Accessibility:** To provide easy access to attendance data for authorized personnel through a user-friendly interface, facilitating efficient data management and analysis.

## 2. Scope:

The scope of the Face Recognition Attendance System project encompasses various sectors and applications, offering a versatile solution to modernize attendance management. The key areas of scope are as follows:

- **Education Sector:** FRAS can be implemented in schools, colleges, and universities to automate the attendance tracking of students and faculty, ensuring accurate and efficient record-keeping.
- **Corporate Environment:** Businesses can adopt FRAS to monitor employee attendance, enhance security, and optimize workforce management, particularly in large organizations.
- **Government Institutions:** Government agencies across different departments can benefit from FRAS by implementing efficient attendance recording methods, thereby improving overall resource utilization.
- **Event Management:** FRAS can simplify registration and attendance tracking at events, conferences, seminars, and workshops, ensuring precise participant records.
- **Access Control Integration:** Beyond attendance, FRAS can seamlessly integrate with access control systems, regulating entry and exit based on attendance status, further enhancing security.

The project's scope encompasses the entire lifecycle, including:

- **Design and Development:** Creating a robust and scalable Face Recognition Attendance System tailored to the needs of the specific sector or organization.
- **Deployment:** Implementing the system in the chosen environment, ensuring compatibility and usability.
- **Maintenance and Support:** Providing ongoing maintenance and technical support to ensure the system's continued reliability and effectiveness.

The Face Recognition Attendance System project aims to bring about a transformation in attendance management by harnessing the capabilities of facial recognition technology. It promises increased efficiency, accuracy, and security while reducing the administrative burden associated with traditional attendance tracking methods across a wide range of applications and sectors.

# System Analysis

## 2.1 Existing System

Before delving into the proposed Face Recognition Attendance System (FRAS) project, it's crucial to understand the limitations and drawbacks of the existing manual attendance tracking systems commonly in use. The existing system typically involves traditional methods such as paper registers, ID card swiping, or manual entry into attendance software. Here, we'll discuss the shortcomings of these conventional systems:

1. **Manual and Time-Consuming:** In the existing manual systems, attendance is recorded through handwritten registers or card swiping machines. This process is labor-intensive and time-consuming, especially in organizations or institutions with a large number of attendees.
2. **Prone to Errors:** Human errors, intentional or unintentional, can lead to inaccuracies in attendance records. Illegible handwriting, data entry mistakes, and accidental omissions can all contribute to unreliable attendance data.
3. **Fraud Vulnerabilities:** Manual systems are susceptible to fraudulent activities, such as buddy punching, where one person records attendance on behalf of another. This can lead to inaccurate attendance records and financial losses for organizations.
4. **Limited Accessibility:** Access to attendance records is often restricted, making it challenging for relevant stakeholders (e.g., students) to access and verify their attendance data in real-time.
5. **Lack of Security:** Conventional methods do not offer robust security measures to prevent unauthorized access or tampering with attendance records. ID cards can be shared, and handwritten registers can be manipulated.
6. **Inefficiency and Paper Dependency:** Paper-based registers require storage space and are vulnerable to damage or loss. Card swiping systems may malfunction, leading to disruptions in attendance tracking.
7. **Inflexibility:** Traditional systems lack flexibility in terms of adapting to different attendance tracking requirements. They often cannot accommodate variations in attendance policies or capture additional data beyond basic check-in/check-out times.

## 2.2 Proposed System

All the students of the class must register themselves by entering the required details and then their images will be captured and stored in the dataset. During each session, faces will be detected from live streaming video of classroom. The faces detected will be compared with images present in the dataset. If match found, attendance will be marked for the respective student. The task of the proposed system is to capture the face of each student and to store it in the database for their attendance. The face of the student needs to be captured in such a manner that all the feature of the students' face needs to be detected, even the seating and the posture of the student need to be recognized. There is no need for the teacher to manually take attendance in the class because the system records a video and through further processing steps the face is being recognized and the attendance database is updated.

- **Flow chart:**

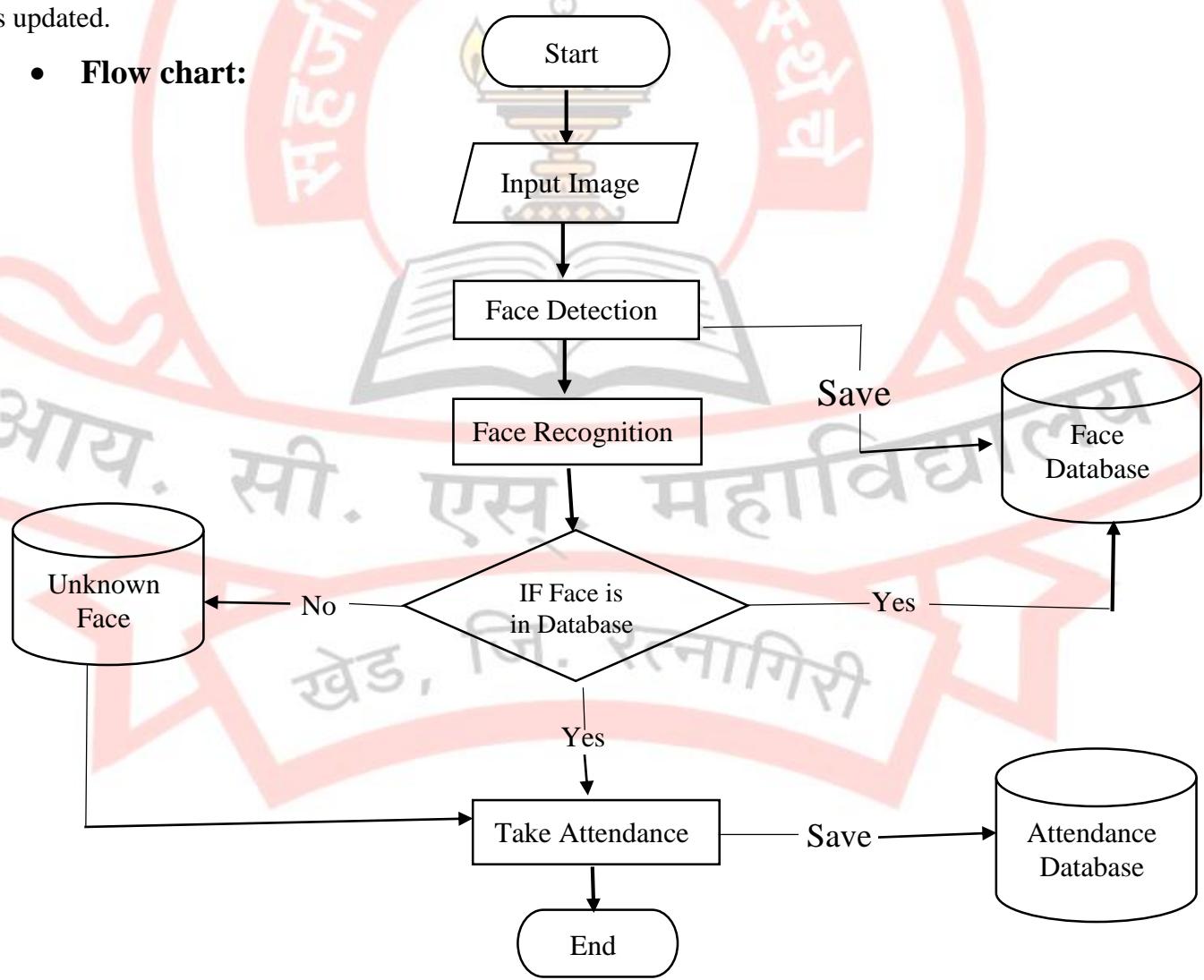


Fig 2.2.1(Project Outline)

## 2.3 Requirement Analysis:

- **Haar cascade Algorithm:**

It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images (where positive images are those where the object to be detected is present, negative are those where it is not). It is then used to detect objects in other images. Luckily, OpenCV offers pre-trained Haar cascade algorithms, organized into categories (faces, eyes and so forth), depending on the images they have been trained on.

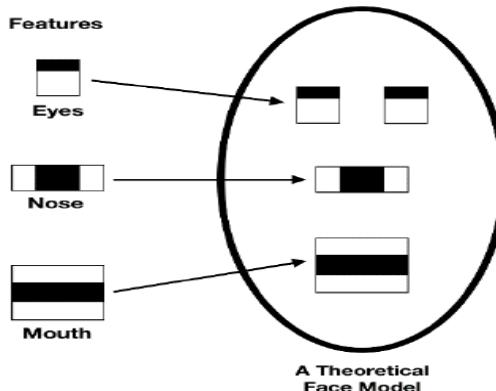


Fig 2.3.1(Haar Features)

- **LBPH Algorithm:**

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

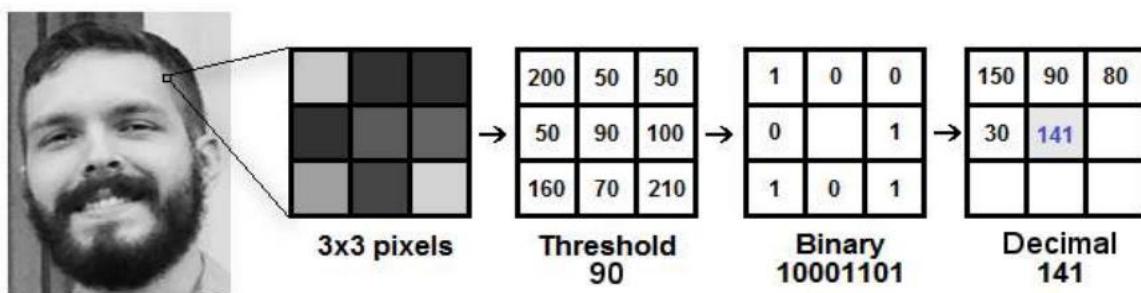


Fig 2.3.2 (LBPH Algorithm)

## **2.4 Hardware Requirements:**

The hardware components required for the FRAS project may include:

- Laptop with 8 GB RAM or above
- Camera 720p or above

## **2.5 Software Requirements:**

The software components required for the FRAS project may include:

- Visual Studio Code
- MySQL Workbench 8.0 CE
- Tkinter

## **2.6 Justification of Selection of Technology:**

The selection of facial recognition technology for FRAS is justified for several reasons:

- **Accuracy:** Facial recognition technology has advanced significantly and can achieve high levels of accuracy in identifying individuals based on unique facial features.
- **Security:** Biometric data (facial features) is difficult to forge, providing robust security against unauthorized access and fraudulent attendance recording.
- **Efficiency:** Facial recognition allows for real-time attendance tracking, reducing administrative overhead and processing time.
- **User-Friendly:** Modern facial recognition systems can be designed with intuitive user interfaces, making them accessible to a wide range of users.
- **Compatibility:** Facial recognition technology can be integrated with existing hardware and software systems, allowing for seamless adoption in various environments.
- **Scalability:** It is adaptable to different scales, making it suitable for both small institutions and large enterprises.
- **Future-Proofing:** As facial recognition technology continues to evolve, the system can be updated to benefit from advancements in accuracy and security.

# System Design

## 3.1 Module Division:

- **OpenCV Library:** OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.
- **NumPy package:** NumPy is a Python package which stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object, provide tools for integrating C, C++ etc. It is also useful in linear algebra, random number capability etc.
- **Pandas Library:** Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the NumPy package and its key data structure is called the Data Frame. Data Frames allow you to store and manipulate tabular data in rows of observations and columns of variables.
- **Tkinter Module:** Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit
- **Time Module:** Python has a module named time to handle time related task. To use functions defined in the module, we need to import the module first.
- **Date Time Module:** A date in python is not a date type of its own, but we can import a module named date time work with dates as a date objects.

### **3.2 Data Dictionary:**

Creating a data dictionary for a Face Recognition Attendance System project helps define the data used in the system, its structure, and its relationships. Below is a simplified data dictionary for such a project:

- **Entities and Their Attributes:**

1. **User Data:**

- **First Name:** User's first name.
- **Last Name:** User's last name.
- **Contact No:** User's latest contact number
- **Email:** Using email as a login identifier on a login page
- **Select Security Questions:** Selecting security questions for email use is an important step in enhancing the security of user email account.
- **Security Answer:** Selecting security answers for security questions is a crucial step in enhancing the security of user email account.
- **Password:** A unique identifier for students.

2. **Student Data:**

- **Department:** Select Student Department (IT/CS/BMS)
- **Course:** Select Student Course (AWP/Java/Python/....)
- **Year:** Select Student Year (2020-21/2021-22/.....)
- **Semester:** Select Semester (Semester I/II/III/....)
- **Student ID (Primary Key):** Unique identifier for each student Information.
- **Student Name:** Student Name
- **Roll No:** Unique identifier for each student Roll No
- **DOB (Date of Birth):** Student Birth Date
- **Gender:** Select Gender (Male /Female / Others)
- **Phone No:** Student phone number
- **Email ID:** Student Email ID
- **Address:** Student Address
- **Photo Sample:** Data representing facial features extracted during registration.

### **3. Attendance Records:**

- **AttendanceID (Primary Key):** Unique identifier for each attendance record.
- **Roll No:** Unique identifier for each student Roll No
- **Name:** Student Name
- **Date:** Date when the attendance was marked.
- **Times:** Time when the attendance was marked.
- **Attendance Status:** Student Present or Absent

- **Relationships:**

Each user can have multiple face images (one-to-many relationship between User Data and Face Data).

Each user can have multiple attendance records (one-to-many relationship between User Data and Attendance Records).

- **Data Types:**

Contact No: Varchar (45)

Email ID: Varchar (45)

AttendanceID: Varchar (45)

First Name: Varchar (45)

Last Name: Varchar (45)

Roll No: Varchar (45)

Select Security Questions: Varchar (45)

Security Answer: Varchar (45)

Password: Varchar (45)

Department: Varchar (45)

Course: Varchar (45)

Year: Varchar (45)

Semester: Varchar (45)

Student ID (Primary Key): Varchar (45)

Student Name: Varchar (45)

DOB (Date of Birth): Varchar (45)

Gender: Varchar (45)

Phone No: Varchar (45)

Address: Varchar (45)

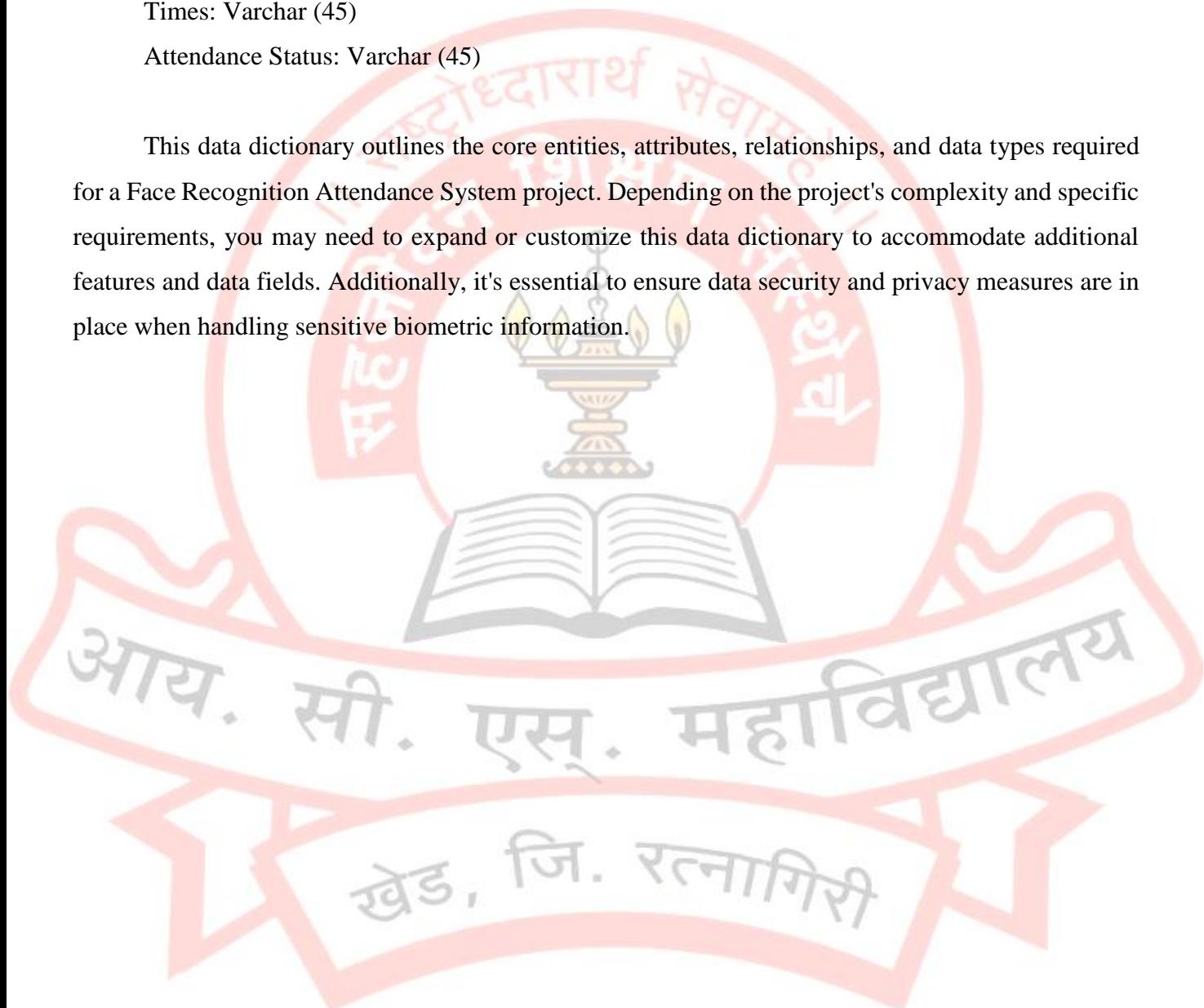
Photo Sample: Varchar (45) / Photo Sample Save in Folder

Date: Varchar (45)

Times: Varchar (45)

Attendance Status: Varchar (45)

This data dictionary outlines the core entities, attributes, relationships, and data types required for a Face Recognition Attendance System project. Depending on the project's complexity and specific requirements, you may need to expand or customize this data dictionary to accommodate additional features and data fields. Additionally, it's essential to ensure data security and privacy measures are in place when handling sensitive biometric information.



### 3.3 ER Diagrams:

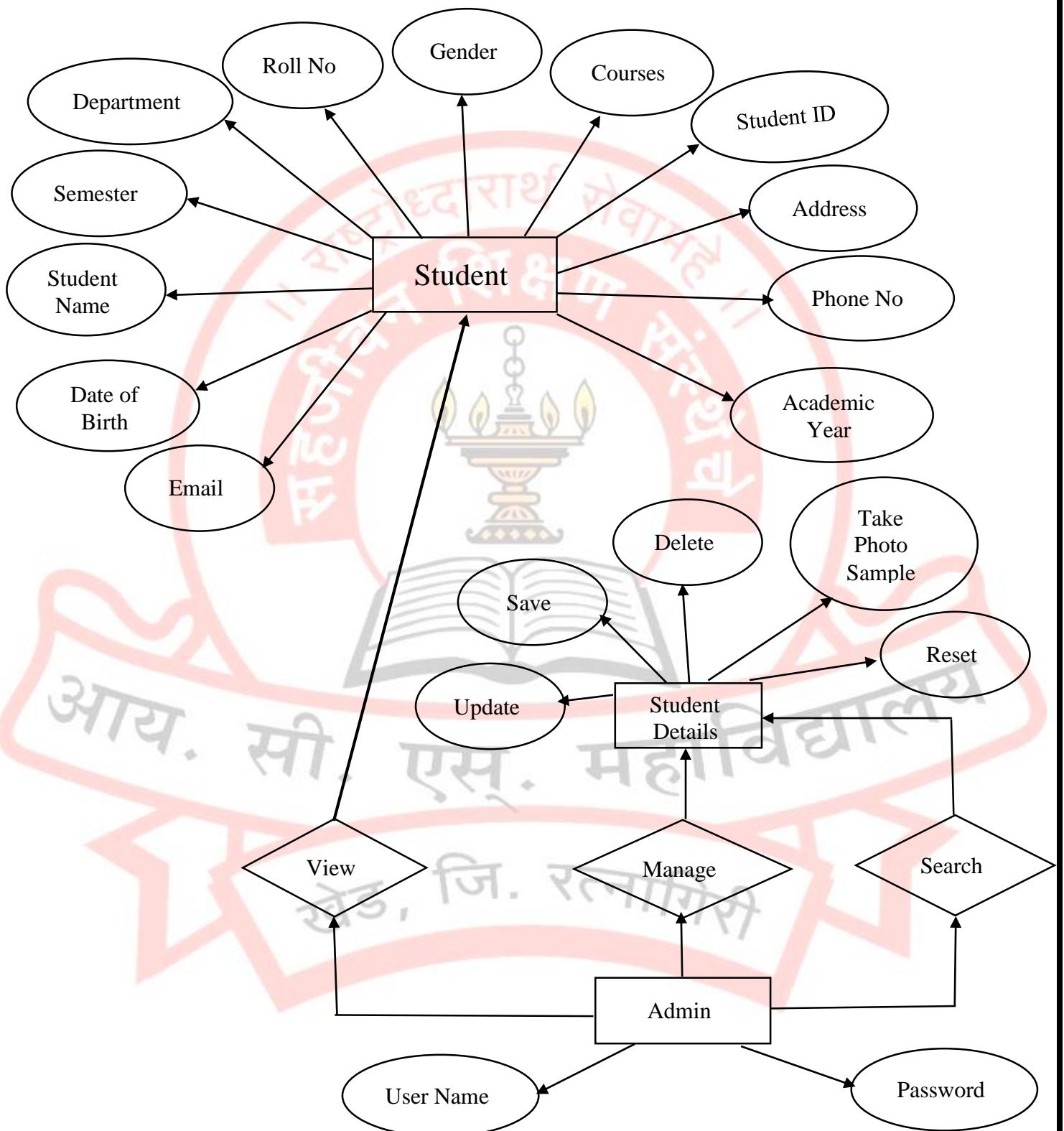


Fig 3.3.1 (ER Diagrams)

### 3.4 DFD/UML Diagrams:

- Data Flow Diagram:

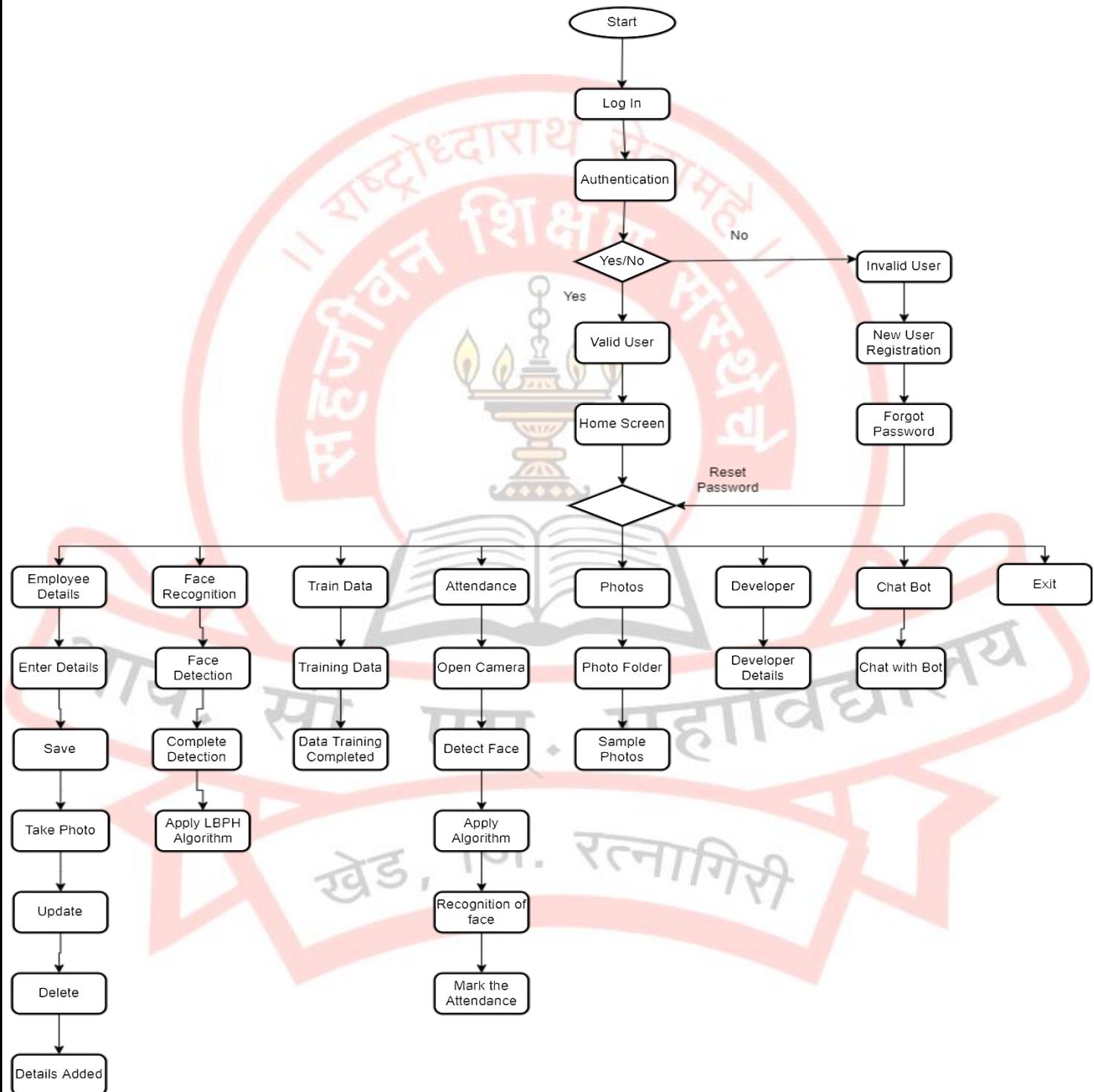


Fig 3.4.1 (Data Flow Diagram)

- Use Case Diagram:

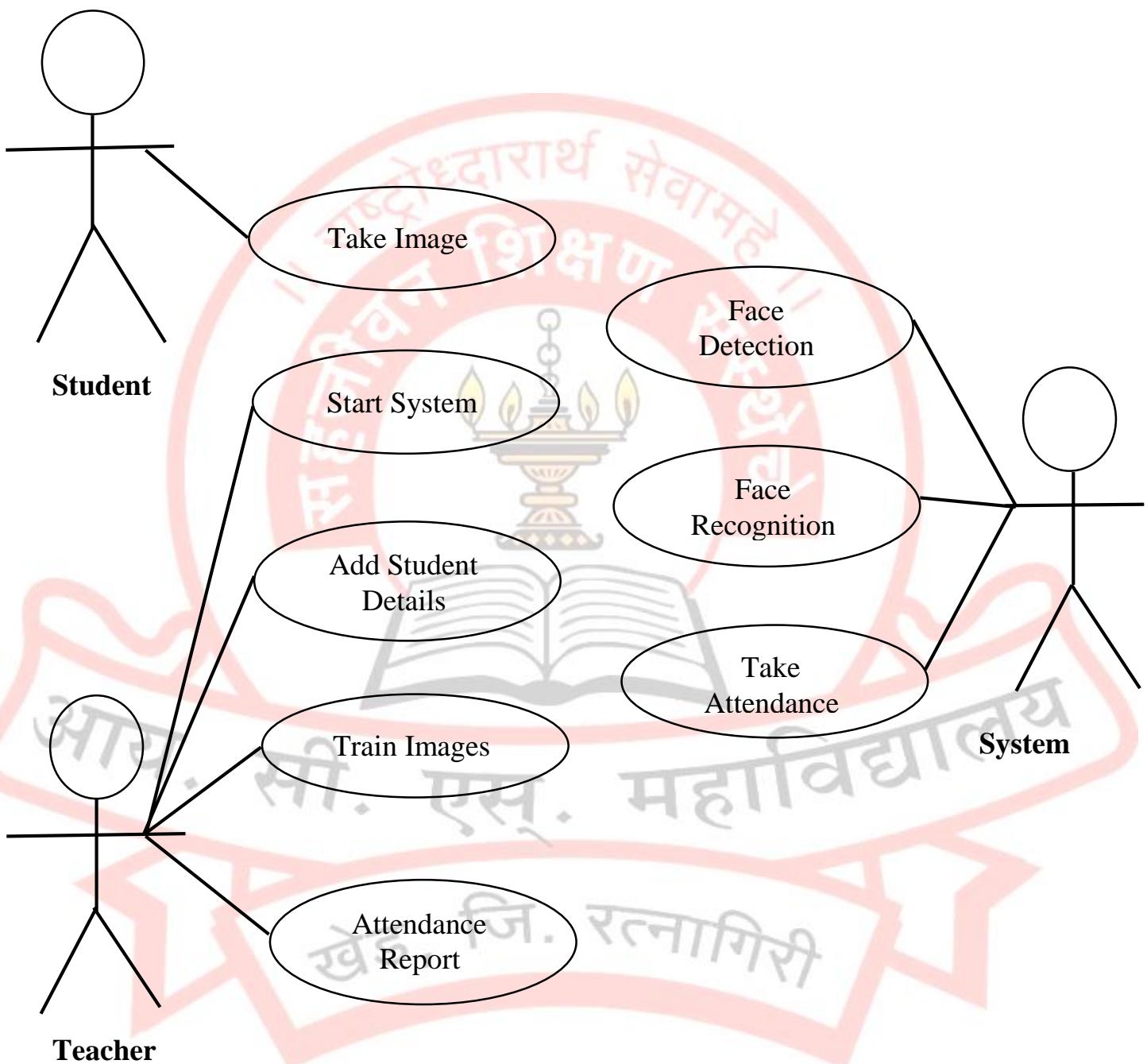


Fig 3.4.2 (Use Case Diagram)

# Implementation and Testing

## 4.1 Code

### 4.1.1 Login Page :

```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk      # pip install pillow
from tkinter import messagebox
import os
import tkinter
from time import strftime
from datetime import datetime
from student import Student
import mysql.connector
```

```
def main():
    win=Tk()
    app=Login_Window(win)
    win.mainloop()
```

```
class Login_Window:
    def __init__(self,root):
        self.root=root
        self.root.title("Login")
        self.root.geometry("1350x720+0+0")
```

```
        # Variables
        self.var_email=StringVar()
        self.var_pass=StringVar()
```

```
# bg Image
```

```
img=Image.open(r"Login_img\images1.jpg")
img=img.resize((1350,720),Image.ANTIALIAS)
```

```
self.photoimg=ImageTk.PhotoImage(img)
```

```
bg_lbl=Label(self.root,image=self.photoimg)
bg_lbl.place(x=0,y=0,width=1350,height=720)
```

```
# first img
img1=Image.open(r"Login_img\images2.jpg")
img1=img1.resize((450,130),Image.ANTIALIAS)
self.photoimg1=ImageTk.PhotoImage(img1)
```

```
f_lbl=Label(bg_lbl,image=self.photoimg1)
f_lbl.place(x=0,y=0,width=450,height=130)
```

```
# second img
img2=Image.open(r"project_img\images22.jpg")
img2=img2.resize((450,130),Image.ANTIALIAS)
```

```
self.photoimg2=ImageTk.PhotoImage(img2)
```

```
f_lbl=Label(bg_lbl,image=self.photoimg2)
f_lbl.place(x=450,y=0,width=450,height=130)
```

```
# third img
img3=Image.open(r"project_img\images19.jpg")
img3=img3.resize((450,130),Image.ANTIALIAS)
self.photoimg3=ImageTk.PhotoImage(img3)
```

```
f_lbl=Label(bg_lbl,image=self.photoimg3)
```

```

f_lbl.place(x=900,y=0,width=450,height=130)

title_lbl=Label(bg_lbl,text="FACE RECOGNITION ATTENDANCE SYSTEM
",font=("times new roman",35,"bold"),bg="white",fg="darkblue")
title_lbl.place(x=0,y=130,width=1350,height=45)

title_lbl=Label(self.root,text="Note : Enter Valid Username and valid Password
",font=("times new roman",20,"bold"),bg="white",fg="darkblue")
title_lbl.place(x=0,y=660,width=1350,height=40)

frame=Frame(self.root,bd=2,bg="black")
frame.place(x=520,y=200,width=340,height=450)

img4=Image.open(r"Login_img\images6.png")
img4=img4.resize((100,100),Image.ANTIALIAS)
self.photoimg4=ImageTk.PhotoImage(img4)

f_lbl=Label(image=self.photoimg4,bg="black",borderwidth=0)
f_lbl.place(x=645,y=210,width=100,height=100)

get_str=Label(frame,text="GetStarted",font=("timesnewroman",20,"bold"),fg="white",bg="black")
get_str.place(x=95,y=105)

# Labels
# User Name

username_label=Label(frame,text="Username:",font=("timesnewroman",15,"bold"),fg="white",bg="black")
username_label.place(x=70,y=155)

self.txtuser=ttk.Entry(frame,font=("times new roman",15,"bold"))
self.txtuser.place(x=40,y=180,width=270)

```

```

# Password

password_label=Label(frame,text="Password:",font=("timesnewroman",15,"bold"),fg="white",bg="black")
password_label.place(x=70,y=225)

self.txtpass=ttk.Entry(frame,font=("times new roman",15,"bold"),show="*")
self.txtpass.place(x=40,y=250,width=270)

# Show Password img
show_img=Image.open(r"Login_img\images7.png")
show_img=show_img.resize((30,25),Image.ANTIALIAS)
self.photoimg_s=ImageTk.PhotoImage(show_img)

# Hide Password img
hide_img=Image.open(r"Login_img\images8.png")
hide_img=hide_img.resize((30,25),Image.ANTIALIAS)
self.photoimg_h=ImageTk.PhotoImage(hide_img)

# Show passWord Button
show_btn=Button(frame,image=self.photoimg_s,cursor="hand2",command=self.show,borderwidth=0
, bg="white")
show_btn.place(x=280,y=250,width=30)

# ***** Icon Images *****
img5=Image.open(r"Login_img\images6.png")
img5=img5.resize((30,30),Image.ANTIALIAS)
self.photoimg5=ImageTk.PhotoImage(img5)

```

```
f_lbl=Label(image=self.photoimg5,bg="black",borderwidth=0)
f_lbl.place(x=560,y=350,width=30,height=30)
```

```
img6=Image.open(r"Login_img\images2.png")
img6=img6.resize((30,30),Image.ANTIALIAS)
self.photoimg6=ImageTk.PhotoImage(img6)
```

```
f_lbl=Label(image=self.photoimg6,bg="black",borderwidth=0)
f_lbl.place(x=560,y=420,width=30,height=30)
```

```
#Login button
```

```
login_btn=Button(frame,text="Login",command=self.login,cursor="hand2",font=("times
new
roman",15,"bold"),bd=2,relief=RIDGE,fg="white",bg="red",activeforeground="white",activebackground="red")
login_btn.place(x=110,y=300,width=120,height=35)
```

```
# register button
```

```
register_btn=Button(frame,text="NewUserRegister",command=self.register_window,cursor="hand2"
,font=("timesnewroman",10,"bold"),borderwidth=0,fg="white",bg="black",activeforeground="white"
,activebackground="black")
```

```
register_btn.place(x=15,y=340,width=160)
```

```
# forget password button
```

```
forgetpass_btn=Button(frame,text="Forgot
Password",command=self.forgot_password_window,cursor="hand2",font=("timesnewroman",10,"bol
d"),borderwidth=0,fg="white",bg="black",activeforeground="white",activebackground="black")
forgetpass_btn.place(x=10,y=370,width=160)
```

```

# Show password
def show(self):

hide_btn=Button(image=self.photoimg_h,cursor="hand2",command=self.hide,borderwidth=0,bg="white")
hide_btn.place(x=800,y=455,width=30)
self.txtpass.config(show="")

# Hide password
def hide(self):

show_btn=Button(image=self.photoimg_s,cursor="hand2",command=self.show,borderwidth=0,bg="white")
show_btn.place(x=800,y=455,width=30)
self.txtpass.config(show='*')

def register_window(self):
    self.new_window=Toplevel(self.root)
    self.app=Register(self.new_window)

def login(self):
    if self.txtuser.get()=="" or self.txtpass.get()=="":

        messagebox.showerror("Error","All field required")

    elif self.txtuser.get()=="Payal" and self.txtpass.get()=="prem":

        messagebox.showinfo("Success","Welcome to the Project")

    else:


```

```

conn=mysql.connector.connect(host="localhost",user="root",password="p123@g",database="mydata")
")
my_cursor=conn.cursor()
my_cursor.execute("select * from register where email=%s and password=%s",(
self.var_email.get(),
self.var_pass.get()
))
row=my_cursor.fetchone()
if row != None:
    messagebox.showerror("Error","Inavalid Username & Password")
else:
    open_main=messagebox.askyesno("YesNo","Access only admin")
    if open_main >0:
        self.new_window=Toplevel(self.root)
        self.app=Face_Recognition_System(self.new_window)
    else:
        if not open_main:
            return
    conn.commit()
    conn.close()

# ***** Reset password *****
def reset_pass(self):
    if self.combo_security.get()=="Select":
        messagebox.showerror("Error","Select the Security Quetion",parent=self.root2)
    elif self.txt_security.get()=="":
        messagebox.showerror("Error","Plaese Enter the Answer ",parent=self.root2)
    elif self.txt_newpass.get()=="":

```

```

        messagebox.showerror("Error","Plaese Enter the New Password",parent=self.root2)

    else:

conn=mysql.connector.connect(host="localhost",user="root",password="p123@g",database="mydata")
my_cursor=conn.cursor()
qry=("select * from register where email=%s and securityQ=%s and securityA=%s")
value=(self.txtuser.get(),self.combo_security.get(),self.txt_security.get())
my_cursor.execute(qry,value)
row=my_cursor.fetchone()
if row ==None:
    messagebox.showerror("Error","Plaese Enter correct Answer",parent=self.root2)
else:
    query=("update register set password=%s where email=%s")
    value=(self.txt_newpass.get(),self.txtuser.get())
    my_cursor.execute(query,value)
    conn.commit()
    conn.close()
    messagebox.showinfo("Info","Your Password has been reset, Plaese login new password",parent=self.root2)
    self.root2.destroy()

# *****Forgot password window *****
def forgot_password_window(self):
    if self.txtuser.get()=="":
        messagebox.showerror("Error","Plaese Enter the Email address to reset password")
    else:

```

```

conn=mysql.connector.connect(host="localhost",user="root",password="p123@g",database="mydata
")

my_cursor=conn.cursor()
query=("select * from register where email=%s")
value=(self.txtuser.get(),)
my_cursor.execute(query,value)
row=my_cursor.fetchone()
#print(row)

if row== None:
    messagebox.showerror("Error","Plaese enter the valid username")
else:
    conn.close()
    self.root2=Toplevel()
    self.root2.title("Forget Password")
    self.root2.geometry("340x450+520+200")

l=Label(self.root2,text="ForgetPassword",font=("timesnewroman",20,"bold"),bg="white",fg="red")
l.place(x=0,y=10,relwidth=1)

security_Q=Label(self.root2,text="Select Security Quetions ",font=("times new
roman",15,"bold"),bg="white")
security_Q.place(x=50,y=80)

self.combo_security=ttk.Combobox(self.root2,font=("timesnewroman",15,"bold"),state="readonly")
self.combo_security["values"]=("Select ","Your Birth Place","Your Friend
Name","Your Pet Name")
self.combo_security.place(x=50,y=110,width=250)
self.combo_security.current(0)

```

```
security_A=Label(self.root2,text="SecurityAnswer:",font=("timesnewroman",15,"bold"),bg="white")
    security_A.place(x=50,y=150)
```

```
self.txt_security=ttk.Entry(self.root2,font=("times new roman",15,"bold"))
    self.txt_security.place(x=50,y=180,width=250)
```

```
new_password=Label(self.root2,text="NewPassword",font=("timesnewroman",15,"bold"),bg="white")
")
```

```
    new_password.place(x=50,y=220)
```

```
self.txt_newpass=ttk.Entry(self.root2,font=("times new roman",15,"bold"))
    self.txt_newpass.place(x=50,y=250,width=250)
```

```
btn=Button(self.root2,text="Reset",command=self.reset_pass,font=("times
    new
roman",15,"bold"),fg="white",bg="green")
```

```
    btn.place(x=120,y=290,width=100)
```

```
if __name__=="__main__":
    main()
```

#### **4.1.2 Main Page :**

```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
import os
import tkinter
from time import strftime
from datetime import datetime
from student import Student
from train import Train
from face_recognition import Face_Recognition
from attendance import Attendance
from developer import Developer
from chatbot import ChatBot

class Face_Recognition_System:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1350x720+0+0")
        self.root.title("Face Recognition System")

    # first img
    img=Image.open(r"project_img\images1.jpg")
    img=img.resize((500,130),Image.ANTIALIAS)
    self.photoimg=ImageTk.PhotoImage(img)

    f_lbl=Label(self.root,image=self.photoimg)
    f_lbl.place(x=0,y=0,width=500,height=130)

    # second img
    img1=Image.open(r"project_img\images22.jpg")
```

```

img1=img1.resize((500,130),Image.ANTIALIAS)
self.photoimg1=ImageTk.PhotoImage(img1)

f_lbl=Label(self.root,image=self.photoimg1)
f_lbl.place(x=500,y=0,width=500,height=130)

# third img
img2=Image.open(r"project_img\images2.jpg")
img2=img2.resize((500,130),Image.ANTIALIAS)
self.photoimg2=ImageTk.PhotoImage(img2)

f_lbl=Label(self.root,image=self.photoimg2)
f_lbl.place(x=1000,y=0,width=400,height=130)

# bg img
img3=Image.open(r"project_img\images4.jpg")
img3=img3.resize((1350,790),Image.ANTIALIAS)
self.photoimg3=ImageTk.PhotoImage(img3)

bg_img=Label(self.root,image=self.photoimg3)
bg_img.place(x=0,y=130,width=1350,height=660)

title_lbl=Label(bg_img,text="FACE RECOGNITION ATTENDANCE SYSTEM ",font=("times new roman",35,"bold"),bg="white",fg="darkgreen")
title_lbl.place(x=0,y=0,width=1350,height=45)

***** Time *****
def time():
    string = strftime("%H:%M:%S %p")
    lbl.config(text= string)
    lbl.after(1000, time)

```

```

lbl=Label(title_lbl,font=("times new roman",15,"bold"),bg="white",fg="blue")
lbl.place(x=0,y=0,width=110,height=50)
time()

# student button
img4=Image.open(r"project_img\images10.jpg")
img4=img4.resize((220,220),Image.ANTIALIAS)
self.photoimg4=ImageTk.PhotoImage(img4)

b1=Button(bg_img,image=self.photoimg4,command=self.student_details,cursor="hand2")
b1.place(x=100,y=70,width=220,height=220)

b1_1=Button(bg_img,text="StudentDetails",command=self.student_details,cursor="hand2",font=("times new roman",15,"bold"),bg="darkblue",fg="white")
b1_1.place(x=100,y=260,width=220,height=40)

# Detect face button
img5=Image.open(r"project_img\images5.jpg")
img5=img5.resize((220,220),Image.ANTIALIAS)
self.photoimg5=ImageTk.PhotoImage(img5)

b1=Button(bg_img,image=self.photoimg5,cursor="hand2",command=self.face_data)
b1.place(x=400,y=70,width=220,height=220)

b1_1=Button(bg_img,text="FaceDetector",cursor="hand2",command=self.face_data,font=("times new roman",15,"bold"),bg="darkblue",fg="white")
b1_1.place(x=400,y=260,width=220,height=40)

# Attendance button
img6=Image.open(r"project_img\images6.jpg")

```

```

        img6=img6.resize((220,220),Image.ANTIALIAS)
        self.photoimg6=ImageTk.PhotoImage(img6)

b1=Button(bg_img,image=self.photoimg6,cursor="hand2",command=self.attendance_data)
b1.place(x=700,y=70,width=220,height=220)

b1_1=Button(bg_img,text="Attendance",cursor="hand2",command=self.attendance_data,font=("timesnewroman",15,"bold"),bg="darkblue",fg="white")
b1_1.place(x=700,y=260,width=220,height=40)

# ChatBot button
img7=Image.open(r"project_img\images12.jpg")
img7=img7.resize((220,220),Image.ANTIALIAS)
self.photoimg7=ImageTk.PhotoImage(img7)

b1=Button(bg_img,image=self.photoimg7,cursor="hand2",command=self.chatbot_data)
b1.place(x=1000,y=70,width=220,height=220)

b1_1=Button(bg_img,text="ChatBot",cursor="hand2",command=self.chatbot_data,font=("times new roman",15,"bold"),bg="darkblue",fg="white")
b1_1.place(x=1000,y=260,width=220,height=40)

# Train Data button
img8=Image.open(r"project_img\images7.jpg")
img8=img8.resize((220,220),Image.ANTIALIAS)
self.photoimg8=ImageTk.PhotoImage(img8)

b1=Button(bg_img,image=self.photoimg8,cursor="hand2",command=self.train_data)
b1.place(x=100,y=330,width=220,height=220)

```

```
b1_1=Button(bg_img,text="TrainData",command=self.train_data,cursor="hand2",font=("times new roman",15,"bold"),bg="darkblue",fg="white")  
b1_1.place(x=100,y=530,width=220,height=40)
```

```
# Photos button  
img9=Image.open(r"project_img\images11.jpg")  
img9=img9.resize((220,220),Image.ANTIALIAS)  
self.photoimg9=ImageTk.PhotoImage(img9)  
  
b1=Button(bg_img,image=self.photoimg9,cursor="hand2",command=self.open_img)  
b1.place(x=400,y=330,width=220,height=220)
```

```
b1_1=Button(bg_img,text="Photos",cursor="hand2",command=self.open_img,font=("timesnew roman",15,"bold"),bg="darkblue",fg="white")  
b1_1.place(x=400,y=530,width=220,height=40)
```

```
# Developer button  
img10=Image.open(r"project_img\images9.jpg")  
img10=img10.resize((220,220),Image.ANTIALIAS)  
self.photoimg10=ImageTk.PhotoImage(img10)
```

```
b1=Button(bg_img,image=self.photoimg10,cursor="hand2",command=self.developer_data)  
b1.place(x=700,y=330,width=220,height=220)
```

```
b1_1=Button(bg_img,text="Developer",cursor="hand2",command=self.developer_data,font=("times new roman",15,"bold"),bg="darkblue",fg="white")  
b1_1.place(x=700,y=530,width=220,height=40)
```

```
# Exit button
```

```

img11=Image.open(r"project_img\images14.jpg")
img11=img11.resize((220,220),Image.ANTIALIAS)
self.photoimg11=ImageTk.PhotoImage(img11)

b1=Button(bg_img,image=self.photoimg11,cursor="hand2",command=self.iExit)
b1.place(x=1000,y=330,width=220,height=220)

b1_1=Button(bg_img,text=" Exit ",cursor="hand2",command=self.iExit,font=("times new roman",15,"bold"),bg="darkblue",fg="white")
b1_1.place(x=1000,y=530,width=220,height=40)

def open_img(self):
    os.startfile("data")

def iExit(self):
    self.iExit=tkinter.messagebox.askyesno("Face Recognition","Are you sure exit this project",parent=self.root)
    if self.iExit >0:
        self.root.destroy()
    else :
        return

# ***** Functions buttons *****
def student_details(self):
    self.new_window=Toplevel(self.root)
    self.app=Student(self.new_window)

def train_data(self):
    self.new_window=Toplevel(self.root)

```

```
self.app=Train(self.new_window)

def face_data(self):
    self.new_window=Toplevel(self.root)
    self.app=Face_Recognition(self.new_window)

def attendance_data(self):
    self.new_window=Toplevel(self.root)
    self.app=Attendance(self.new_window)

def developer_data(self):
    self.new_window=Toplevel(self.root)
    self.app=Developer(self.new_window)

def chatbot_data(self):
    self.new_window=Toplevel(self.root)
    self.app=ChatBot(self.new_window)

if __name__=="__main__":
    root=Tk()
    obj= Face_Recognition_System(root)
    root.mainloop()
```

#### **4.1.3 Face Recognition Page :**

```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
import os
import numpy as np
from time import strftime
from datetime import datetime

class Face_Recognition:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1350x720+0+0")
        self.root.title("Face Recognition System")

        title_lbl=Label(self.root,text="FACERECOGNITION",font=("timesnewroman",35,"bold"),bg="white",fg="green")
        title_lbl.place(x=0,y=0,width=1350,height=45)

        # 1st image
        img_top=Image.open(r"project_img\images5.jpg")
        img_top=img_top.resize((600,670),Image.ANTIALIAS)
        self.photoimg_top=ImageTk.PhotoImage(img_top)

        f_lbl=Label(self.root,image=self.photoimg_top)
        f_lbl.place(x=0,y=55,width=600,height=670)

        # 2nd image
```

```

img_bottom=Image.open(r"project_img\images1.webp")
img_bottom=img_bottom.resize((950,670),Image.ANTIALIAS)
self.photoimg_bottom=ImageTk.PhotoImage(img_bottom)

f_lbl=Label(self.root,image=self.photoimg_bottom)
f_lbl.place(x=600,y=55,width=950,height=670)

# button

b1_1=Button(f_lbl,text="FaceRecognition",command=self.face_recog,cursor="hand2",font=("times new roman",15,"bold"),bg="darkblue",fg="white")
b1_1.place(x=370,y=590,width=200,height=40)

# ***** Attendance *****
def mark_attendance(self,i,r,n,d):
    with open("payal.csv","r+",newline="\n") as f:
        myDataList=f.readlines()
        name_list=[] # del self.bill_list[:]
        for line in myDataList:
            entry=line.split(",")
            name_list.append(entry[0])
        if((i not in name_list) and (r not in name_list) and (n not in name_list) and (d not in name_list)):
            now=datetime.now()
            d1=now.strftime("%d/%m/%Y")
            dtString=now.strftime("%H:%M:%S")
            f.writelines(f"\n{i},{r},{n},{d},{dtString},{d1},Preset")
# ***** face recognition *****

```

```

def face_recog(self):
    def draw_boundray(img,classifier,scaleFactor,minNeighbors,color,text,clf):
        gray_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
        features=classifier.detectMultiScale(gray_image,scaleFactor,minNeighbors)

        coord=[]

        for (x,y,w,h) in features:
            cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),3)
            id,predict=clf.predict(gray_image[y:y+h,x:x+w])
            confidence=int((100*(1-predict/300)))

```

**आय. सी. एस. महाविद्यालय**  
**खेड, जि. रत्नागिरी**

```

conn=mysql.connector.connect(host="localhost",username="root",password="p123@g",database="face_recognizer")
my_cursor=conn.cursor()

my_cursor.execute("select Name from student where Student_id="+str(id))
n=my_cursor.fetchone()
n)+"."+join(n)

my_cursor.execute("select Roll from student where Student_id="+str(id))
r=my_cursor.fetchone()
r)+"."+join(r)

my_cursor.execute("select Dep from student where Student_id="+str(id))
d=my_cursor.fetchone()
d)+"."+join(d)

my_cursor.execute("select Student_id from student where Student_id="+str(id))
i=my_cursor.fetchone()

```

```

i= "+".join(i)
if confidence > 77:

cv2.putText(img,f"ID:{i}",(x,y75),cv2.FONT_HERSHEY_COMPLEX,0.7,(0,0,0),2)

cv2.putText(img,f"Roll:{r}",(x,y55),cv2.FONT_HERSHEY_COMPLEX,0.7,(0,0,0),2)

cv2.putText(img,f"Name:{n}",(x,y30),cv2.FONT_HERSHEY_COMPLEX,0.7,(0,0,0),2)

cv2.putText(img,f"Department:{d}",(x,y5),cv2.FONT_HERSHEY_COMPLEX,0.7,(0,0,0),2)

    self.mark_attendance(i,r,n,d)

else:
    cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3)

cv2.putText(img,"UnknownFace",(x,y5),cv2.FONT_HERSHEY_COMPLEX,0.8,(0,0,0),3)

coord=[x,y,w,h]

return coord

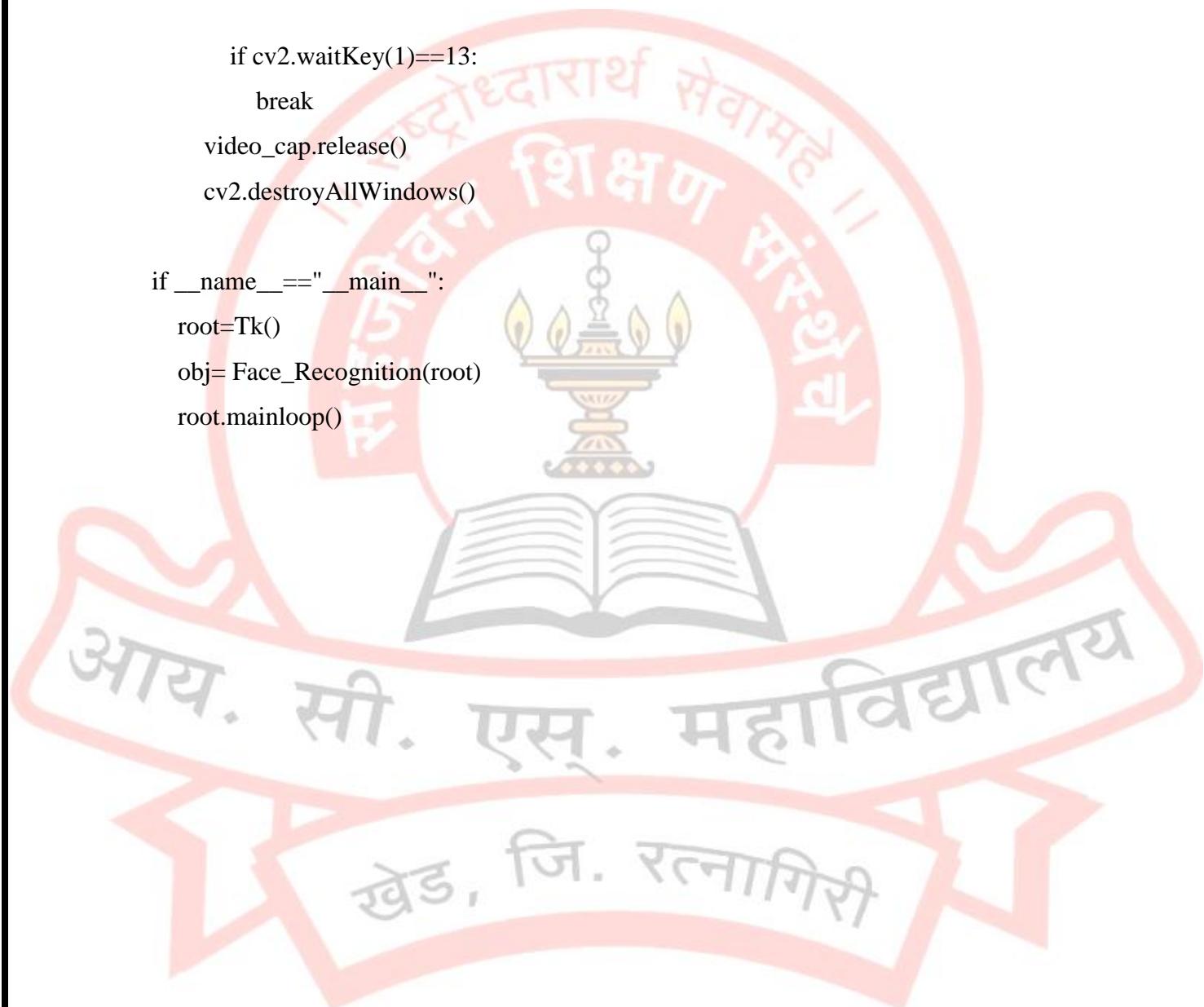
def recognize(img,clf,faceCascade):
    coord=draw_boundray(img,faceCascade,1.1,10,(255,25,255),"Face",clf)
    return img

faceCascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
clf=cv2.face.LBPHFaceRecognizer_create()
clf.read("classifier.xml")

video_cap=cv2.VideoCapture(0)

```

```
while True:  
    ret,img=video_cap.read()  
    img=recognize(img,clf,faceCascade)  
    cv2.imshow("Welcome To Face Recognition",img)  
  
    if cv2.waitKey(1)==13:  
        break  
    video_cap.release()  
    cv2.destroyAllWindows()  
  
if __name__=="__main__":  
    root=Tk()  
    obj= Face_Recognition(root)  
    root.mainloop()
```



#### 4.1.4 Attendance Page :

```
from tkinter import*
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
import os
import csv
from tkinter import filedialog
from tkcalendar import DateEntry

mydata=[]

class Attendance:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1350x720+0+0")
        self.root.title("Face Recognition System")

# ***** Varaibles *****
        self.var_atten_id=StringVar()
        self.var_atten_roll=StringVar()
        self.var_atten_name=StringVar()
        self.var_atten_dep=StringVar()
        self.var_atten_time=StringVar()
        self.var_atten_date=StringVar()
        self.var_atten_attendance=StringVar()

# first img
img=Image.open(r"project_img\images19.jpg")
img=img.resize((675,200),Image.ANTIALIAS)
```

```

self.photoimg=ImageTk.PhotoImage(img)

f_lbl=Label(self.root,image=self.photoimg)
f_lbl.place(x=0,y=0,width=675,height=200)

# second img
img1=Image.open(r"project_img\images16.jpg")
img1=img1.resize((675,200),Image.ANTIALIAS)

self.photoimg1=ImageTk.PhotoImage(img1)

f_lbl=Label(self.root,image=self.photoimg1)
f_lbl.place(x=675,y=0,width=675,height=200)

# bg img
img3=Image.open(r"project_img\images4.jpg")
img3=img3.resize((1350,720),Image.ANTIALIAS)
self.photoimg3=ImageTk.PhotoImage(img3)

bg_img=Label(self.root,image=self.photoimg3)
bg_img.place(x=0,y=200,width=1350,height=550)

title_lbl=Label(bg_img,text="ATTENDANCE MANAGEMENT SYSTEM ",  

font=("times new roman",35,"bold"),bg="white",fg="purple")
title_lbl.place(x=0,y=0,width=1350,height=45)

main_frame=Frame(bg_img,bd=2,bg="white")
main_frame.place(x=10,y=50,width=1330,height=480)

#left lable frame

```

```

Left_frame=LabelFrame(main_frame,bd=2,bg="white",relief=RIDGE,text="Student
Attendance Details",font=("times new roman",12,"bold"))

Left_frame.place(x=5,y=8,width=660,height=440)

img_left=Image.open(r"project_img\images15.jpg")
img_left=img_left.resize((640,130),Image.ANTIALIAS)
self.photoimg_left=ImageTk.PhotoImage(img_left)

f_lbl=Label(Left_frame,image=self.photoimg_left)
f_lbl.place(x=5,y=0,width=640,height=130)

left_inside_frame=Frame(Left_frame,bd=2,relief=RIDGE,bg="white")
left_inside_frame.place(x=5,y=135,width=640,height=275)

# Label and entry
# Attendance ID
attendanceID_label=Label(left_inside_frame,text="AttendanceID : ",font=("times new
roman",13,"bold"),bg="white")
attendanceID_label.grid(row=0,column=0,padx=10,pady=5,sticky=W)

attendanceID_entry=ttk.Entry(left_inside_frame,width=20,textvariable=self.var_atten_id,font=("time
s new roman",13,"bold"))
attendanceID_entry.grid(row=0,column=1,padx=10,pady=5,sticky=W)

# Roll
rollLabel=Label(left_inside_frame,text="RollNo:",font=("comicsansns11bold"),bg="white")
rollLabel.grid(row=0,column=2,padx=4,pady=8)

```

```
atten_roll=ttk.Entry(left_inside_frame,width=22,textvariable=self.var_atten_roll,font=("comicsansns  
11 bold"))

atten_roll.grid(row=0,column=3,pady=8)
```

#Name

```
nameLabel=Label(left_inside_frame,text="Name:",font=("comicsansns11bold"),bg="white")
nameLabel.grid(row=1,column=0)
```

```
atten_name=ttk.Entry(left_inside_frame,width=22,textvariable=self.var_atten_name,font=("comicsan  
sns 11 bold"))

atten_name.grid(row=1,column=1,pady=8)
```

#Department

```
depLabel=Label(left_inside_frame,text="Department:",font=("comicsansns,11,bold"),bg="white")
depLabel.grid(row=2,column=2)
```

```
atten_dep=ttk.Entry(left_inside_frame,width=22,textvariable=self.var_atten_dep,font=("comicsansns  
11 bold"))

atten_dep.grid(row=2,column=3,pady=8)
```

#Time

```
timeLabel=Label(left_inside_frame,text="Time:",font=("comicsansns11bold"),bg="white")
timeLabel.grid(row=2,column=0)
```

```
atten_time=ttk.Entry(left_inside_frame,width=22,textvariable=self.var_atten_time,font=("co  
micsansns 11 bold"))

atten_time.grid(row=2,column=1,pady=8)
```

```

# Date
dateLabel=Label(left_inside_frame,text="Date:",font=("comicsansns11bold"),bg="white")
dateLabel.grid(row=1,column=2)

#atten_date=ttk.Entry(left_inside_frame,width=22,textvariable=self.var_atten_date,font=("comicsansns 11 bold"))
#atten_date.grid(row=2,column=3,pady=8)

cal=DateEntry(left_inside_frame,selectmode="day",width=18,textvariable=self.var_atten_date,font=("times new roman",13,"bold"))
cal.grid(row=1,column=3,pady=13,sticky=W)

# Attendance
attendanceLabel=Label(left_inside_frame,text="AttendanceStatus : ",font=("comicsansns11 bold"),bg="white")
attendanceLabel.grid(row=3,column=0)

self.attn_status=ttk.Combobox(left_inside_frame,textvariable=self.var_atten_attendance,font=("comicsansns 11 bold"),state="readonly",width=20)
self.attn_status["values"]=("Status","Present","Absent")
self.attn_status.grid(row=3,column=1,pady=8)
self.attn_status.current(0)

# buttons frame
btn_frame=Frame(left_inside_frame,bd=2,relief=RIDGE)
btn_frame.place(x=0,y=200,width=645,height=35)

import_btn=Button(btn_frame,text="Import
csv",command=self.importCsv,width=20,font=("times new roman",13,"bold"),bg="blue",fg="white")
import_btn.grid(row=0,column=0)

```

```

        export_btn=Button(btn_frame,text="Export
csv",command=self.exportCsv,width=20,font=("times new roman",13,"bold"),bg="blue",fg="white")
        export_btn.grid(row=0,column=1)

#update_btn=Button(btn_frame,text="Update",width=15,font=("timesnewroman",13,"bold"),bg="blue",fg="white")
#update_btn.grid(row=0,column=2)

reset_btn=Button(btn_frame,text="Reset",command=self.reset_data,width=22,font=("timesnewromana",13,"bold"),bg="blue",fg="white")
reset_btn.grid(row=0,column=3)

#right lable frame

Right_frame=LabelFrame(main_frame,bd=2,bg="white",relief=RIDGE,text="Attendance
Details",font=("times new roman",12,"bold"))

Right_frame.place(x=675,y=8,width=645,height=440)
table_frame=Frame(Right_frame,bd=2,relief=RIDGE)
table_frame.place(x=5,y=5,width=630,height=405)

# *****scroll bar table*****
# Scrollbar
scroll_x=ttk.Scrollbar(table_frame,orient=HORIZONTAL)
scroll_y=ttk.Scrollbar(table_frame,orient=VERTICAL)

self.AttendanceReportTable=ttk.Treeview(table_frame,columns=("id","roll","name","department","ti
me","date","attendance"),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=self.AttendanceReportTable.xview)
scroll_y.config(command=self.AttendanceReportTable.yview)

# Headings
self.AttendanceReportTable.heading("id",text="Attendance ID")
self.AttendanceReportTable.heading("roll",text="Roll No")
self.AttendanceReportTable.heading("name",text="Name")
self.AttendanceReportTable.heading("department",text="Department")
self.AttendanceReportTable.heading("time",text="Time")
self.AttendanceReportTable.heading("date",text="Date")
self.AttendanceReportTable.heading("attendance",text="AttendanceStatus")

self.AttendanceReportTable["show"]="headings"

self.AttendanceReportTable.column("id",width=100)
self.AttendanceReportTable.column("roll",width=100)
self.AttendanceReportTable.column("name",width=100)
self.AttendanceReportTable.column("department",width=100)
self.AttendanceReportTable.column("time",width=100)
self.AttendanceReportTable.column("date",width=100)
self.AttendanceReportTable.column("attendance",width=100)

self.AttendanceReportTable.pack(fill=BOTH,expand=1)

self.AttendanceReportTable.bind("<ButtonRelease>",self.get_cursor)

*****fetch data *****
def fetchData(self,rows):
    self.AttendanceReportTable.delete(*self.AttendanceReportTable.get_children())
    for i in rows:

```

```

        self.AttendanceReportTable.insert("",END,values=i)

# import csv
def importCsv(self):
    global mydata
    mydata.clear()
    fln=filedialog.askopenfilename(initialdir=os.getcwd(),title="Open
CSV",filetypes=((("CSV File",".csv"),("All File","*.*"))),parent=self.root)
    with open(fln) as myfile :
        csvread=csv.reader(myfile,delimiter=",")
        for i in csvread:
            mydata.append(i)
        self.fetchData(mydata)

# Export csv
def exportCsv(self):
    try:
        if len(mydata)<1:
            messagebox.showerror("No Data","No Data fount to export",parent=self.root)
            return False
        fln=filedialog.asksaveasfilename(initialdir=os.getcwd(),title="Open
CSV",filetypes=((("CSV File",".csv"),("All File","*.*"))),parent=self.root)
        with open(fln,mode="w",newline="") as myfile:
            exp_write=csv.writer(myfile,delimiter=",")
            for i in mydata:
                exp_write.writerow(i)
            messagebox.showinfo("Data      Export","Your      data      exported      to
"+os.path.basename(fln)+"successfully")
    except Exception as es :

```

```
messagebox.showerror("Error",f"Due To :{str(es)}",parent=self.root)

def get_cursor(self,event=""):
    cursor_row=self.AttendanceReportTable.focus()
    content=self.AttendanceReportTable.item(cursor_row)
    rows=content["values"]
    self.var_atten_id.set(rows[0])
    self.var_atten_roll.set(rows[1])
    self.var_atten_name.set(rows[2])
    self.var_atten_dep.set(rows[3])
    self.var_atten_time.set(rows[4])
    self.var_atten_date.set(rows[5])
    self.var_atten_attendance.set(rows[6])

def reset_data(self):
    self.var_atten_id.set("")
    self.var_atten_roll.set("")
    self.var_atten_name.set("")
    self.var_atten_dep.set("")
    self.var_atten_time.set("")
    self.var_atten_date.set("")
    self.var_atten_attendance.set("")

if __name__=="__main__":
    root=Tk()
    obj= Attendance(root)
    root.mainloop()
```

## 4.2 Testing Approach

### 4.2.1 Unit Testing

To perform unit testing for a face recognition attendance system using Python, you would focus on testing the individual components and their integration, such as face detection, face recognition, and database management. Here are some suggestions for unit testing based on the search results:

- **Face detection:** Test the face detection algorithm using various images, including those with different lighting conditions, angles, and resolutions. Ensure that the algorithm can accurately detect faces in these scenarios.
- **Face recognition:** Test the face recognition algorithm using a dataset of known faces. Ensure that the algorithm can accurately recognize faces with varying levels of similarity.
- **Database management:** Test the database management component to ensure that it can efficiently store and retrieve face data. Test the database's ability to handle large volumes of data and to maintain data integrity.
- **Integration:** Test the integration of the face detection and recognition algorithms with the database management component. Ensure that the system can accurately identify and record the attendance of individuals.
- **User interface:** Test the user interface to ensure that it is intuitive and user-friendly. Ensure that the system can easily add or remove students from the database and that it can generate accurate attendance reports.
- **Security:** Test the system's security features to ensure that it can protect user data and prevent unauthorized access.
- **Performance:** Test the system's performance to ensure that it can handle large volumes of data and that it can process images quickly.
- **Error handling:** Test the system's error handling capabilities to ensure that it can gracefully handle errors and provide meaningful feedback to users.
- **Scalability:** Test the system's scalability to ensure that it can handle an increasing number of users and images.

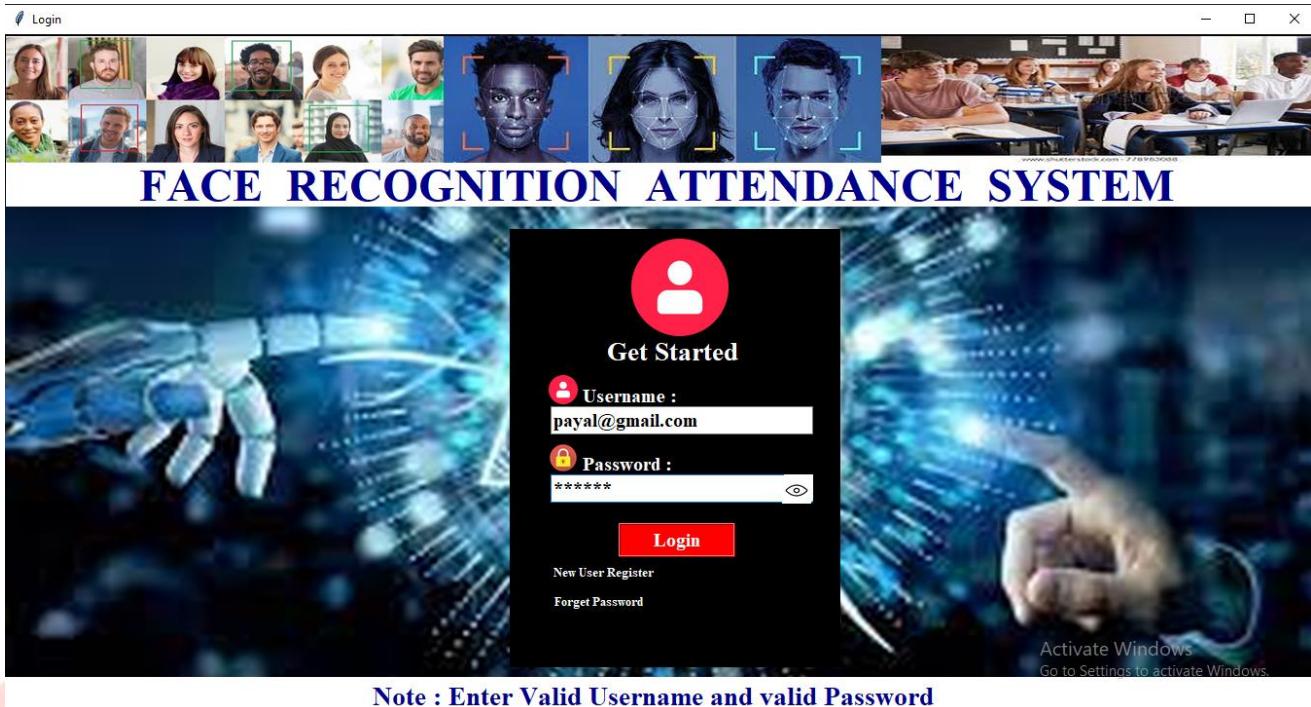
#### **4.2.2 Integration Testing**

Integration testing for a face recognition attendance system using Python can be performed by testing the system's ability to connect and communicate with the various components that make up the system, such as the face detection and recognition algorithms, the database, and the user interface.

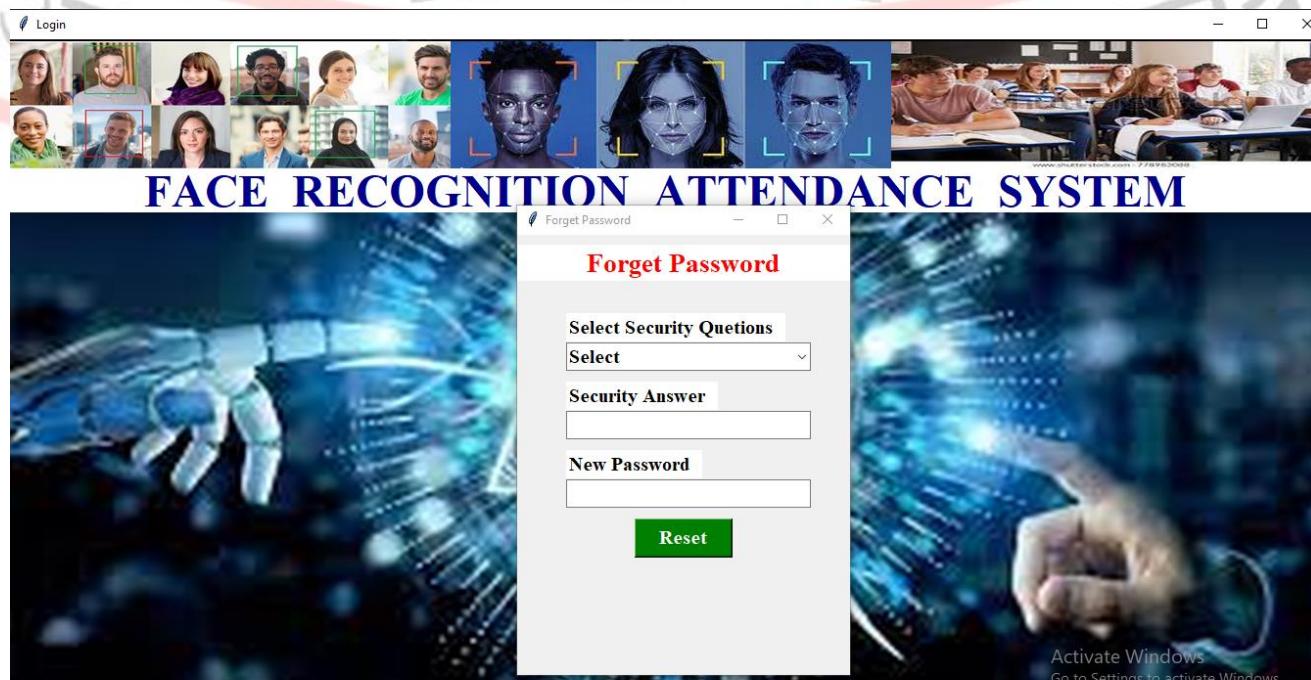
- To perform integration testing, follow these steps:
  1. Set up the system's environment, including the face detection and recognition algorithms, the database, and the user interface.
  2. Test the system's ability to connect to the database and retrieve or update data.
  3. Test the system's ability to detect and recognize faces using the face detection and recognition algorithms.
  4. Test the system's ability to mark attendance based on the recognized faces.
  5. Test the system's ability to generate and send attendance reports.
  6. Test the system's ability to handle errors and provide meaningful feedback to users.
  7. Test the system's ability to handle large volumes of data and maintain data integrity.
  8. Test the system's ability to handle various input scenarios, such as multiple faces in a single frame, low-quality images, and different lighting conditions.
  9. Test the system's ability to handle proxy attendance and ensure that it can accurately identify and mark attendance for authorized individuals only.
  10. Test the system's ability to handle different user roles and permissions, ensuring that only authorized users can access and modify the system's data.

# Results and Discussions

## 5.1 Login Page



## 5.2 Forget Password



### 5.3 Register Page

The screenshot shows a registration form titled "REGISTER HERE". On the left, there is a decorative image of a coffee cup with a "Good Morning" quote: "I like my coffee black and my mornings bright." Below the quote, it says "Your name here". The registration fields include:

- First Name: Input field
- Last Name: Input field
- Contact No: Input field
- Email: Input field
- Select Security Questions: A dropdown menu set to "Select".
- Security Answer: Input field
- Password: Input field
- Confirm Password: Input field

Below the fields are two buttons: "REGISTER NOW" (red) and "Login Now" (blue). There is also a checkbox labeled "I Agree The Terms & Conditions". At the bottom right, there is a watermark for "Activate Windows" with the text "Go to Settings to activate Windows".

### 5.4 Main Page

The main page title is "FACE RECOGNITION ATTENDANCE SYSTEM". It features a header banner with the text "BEST FACIAL RECOGNITION SOFTWARE" and "Analytics Insight". The main interface is divided into several sections:

- Student Details:** Shows a group of people.
- Face Detector:** Shows a close-up of a face with a circular overlay.
- Attendance:** Shows a person standing next to a screen displaying a welcome message.
- ChatBot:** Shows a cartoon robot with a speech bubble saying "HI!"
- Train Data:** Shows a 3D wireframe model of a face.
- Photos:** Shows a grid of many small faces.
- Developer:** Shows a person working at a desk with a laptop.
- Exit:** A large red button with the word "EXIT".

At the bottom right, there is a watermark for "Activate Windows" with the text "Go to Settings to activate Windows".

## 5.5 Student Details



### STUDENT MANAGEMENT SYSTEM

**Student Details**



**Current Course Information**

Department	IT	* Course	Python
Year	2023-24	* Semester	Semester VI

**Class Student Information**

Student ID	1	* Student Name	Payal
Roll NO	617	* DOB	5/28/03
Gender	Female	* Phone No	7264964145
Email ID	payal@gmail.com	* Address	Satara

Take Photo Sample     No Photo Sample

**Action Buttons:** Save | Update | Delete | Reset

**Photo Options:** Take Photo Sample | Update Photo Sample

**Student Details**



**Search System**

Search By :	Select	Search	Show All
-------------	--------	--------	----------

Department	Course	Year	Semester	Student ID	Name
IT	Python	2023-24	Semester VI	1	Payal

Activate Windows  
Go to Settings to activate Windows.



### MANAGEMENT SYSTEM

**Student Details**



**Current Course Information**

Department	IT	* Course	Python
Year	2023-24	* Semester	Semester VI

**Class Student Information**

Student ID	1	* Student Name	Payal
Roll NO	617	* DOB	5/28/03
Gender	Female	* Phone No	7264964145
Email ID	payal@gmail.com	* Address	Satara

Take Photo Sample     No Photo Sample

**Action Buttons:** Save | Update | Delete | Reset

**Photo Options:** Take Photo Sample | Update Photo Sample

**Student Details**



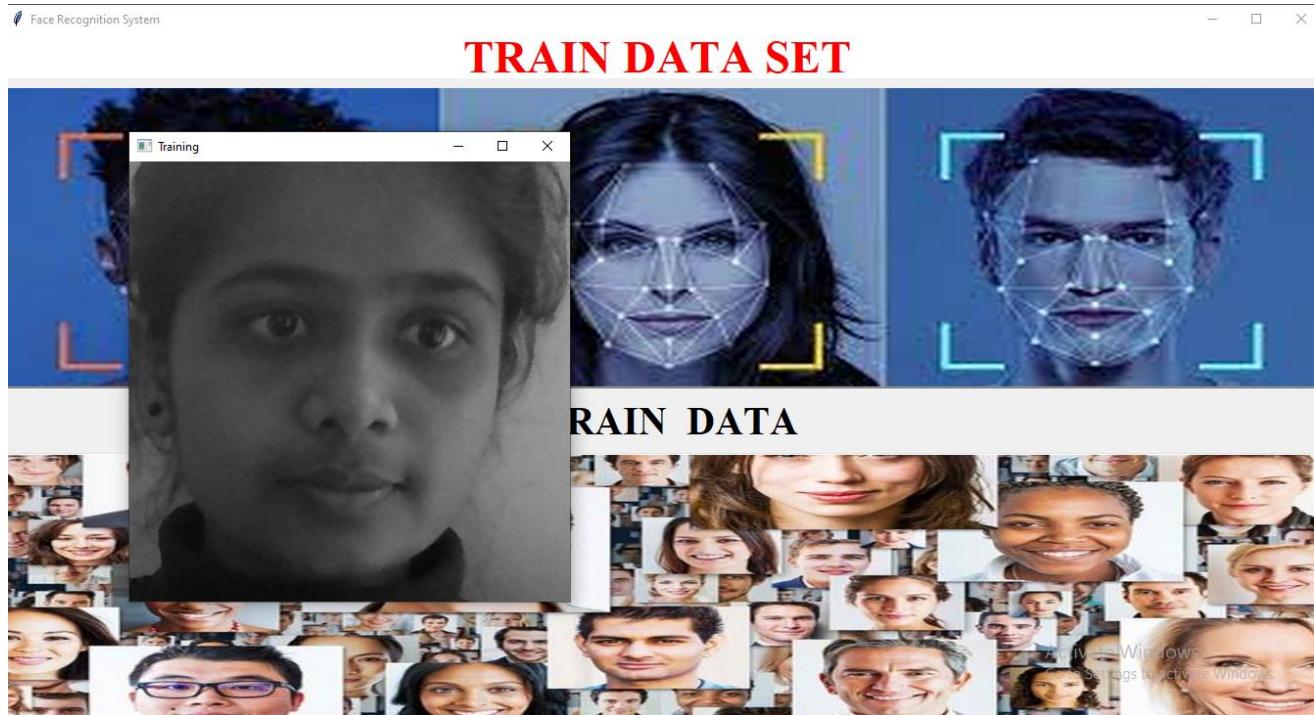
**Search System**

Search By :	Select	Search	Show All
-------------	--------	--------	----------

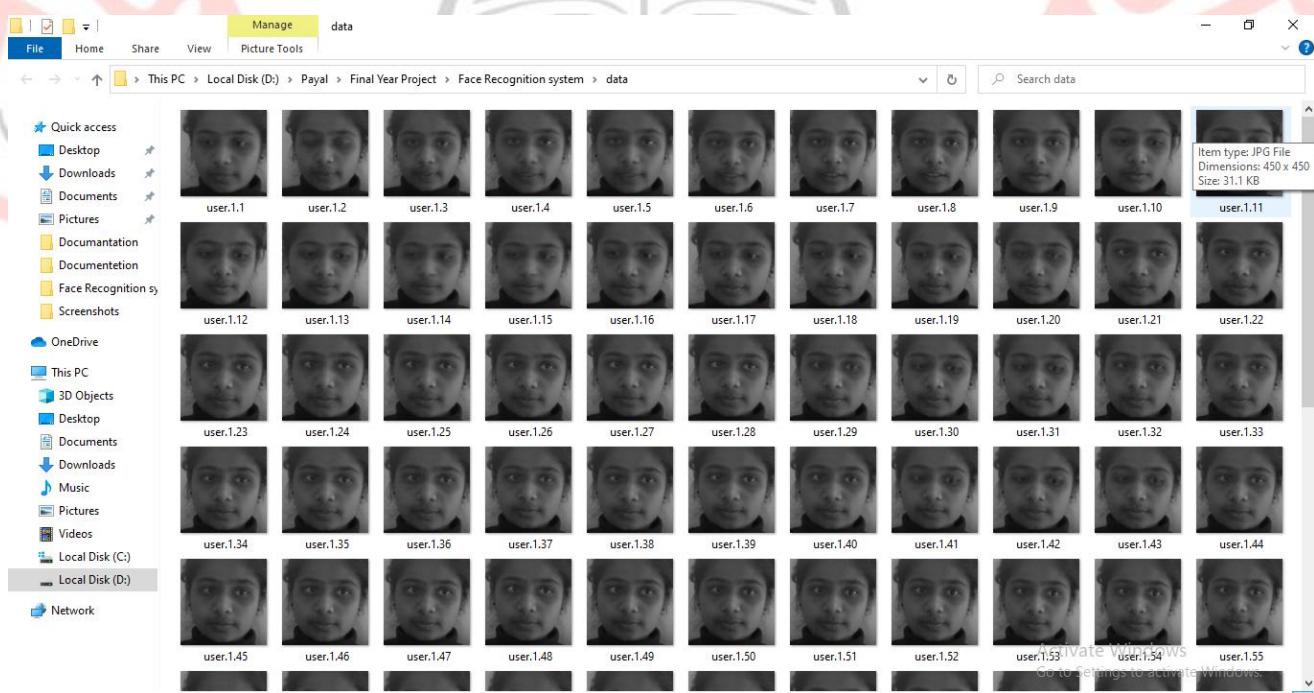
Department	Course	Year	Semester	Student ID	Name
IT	Python	2023-24	Semester VI	1	Payal

Activate Windows  
Go to Settings to activate Windows.

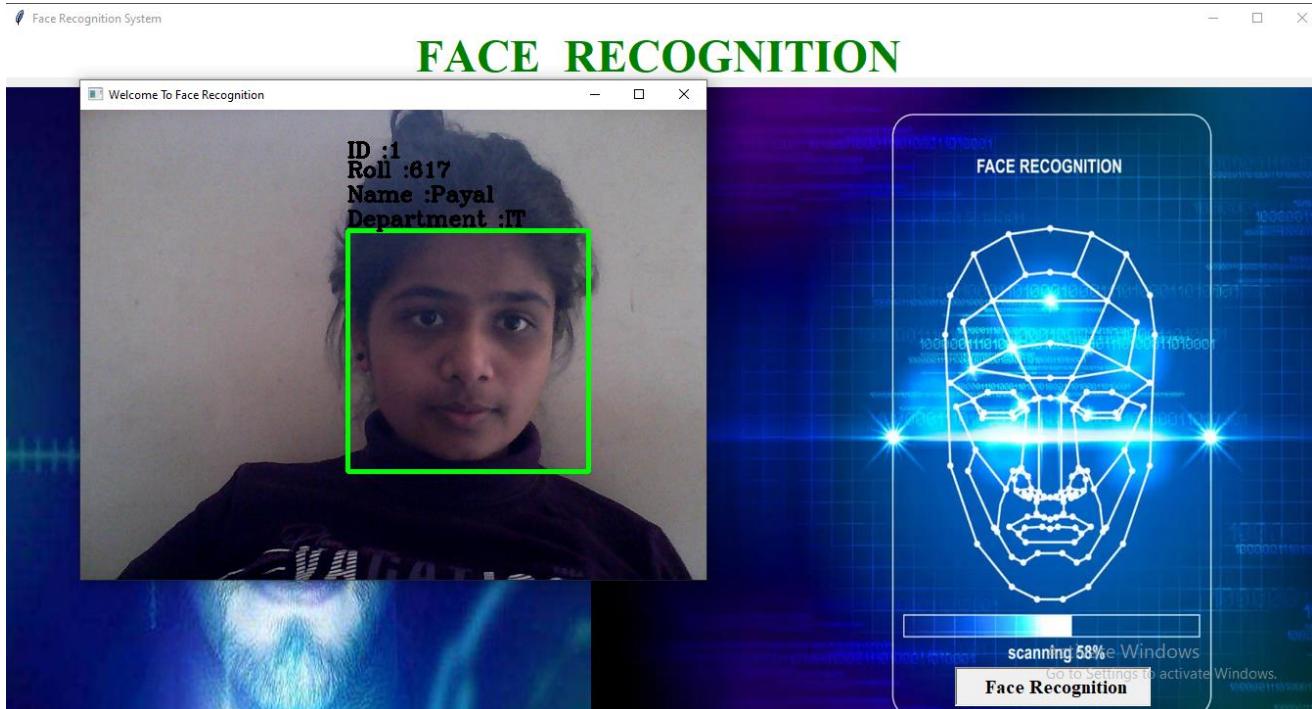
## 5.6 Train Data Set



## 5.7 Images Data Folder



## 5.8 Face Recognition



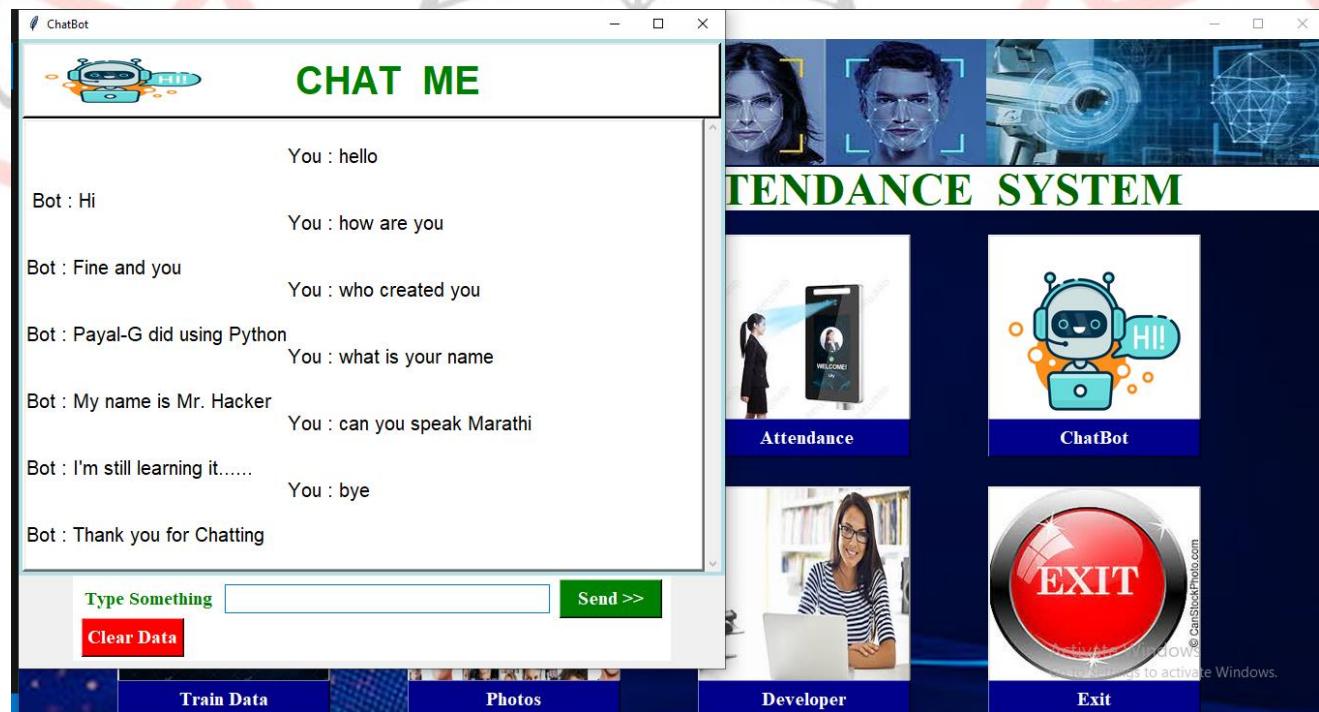
## 5.9 Attendance Page



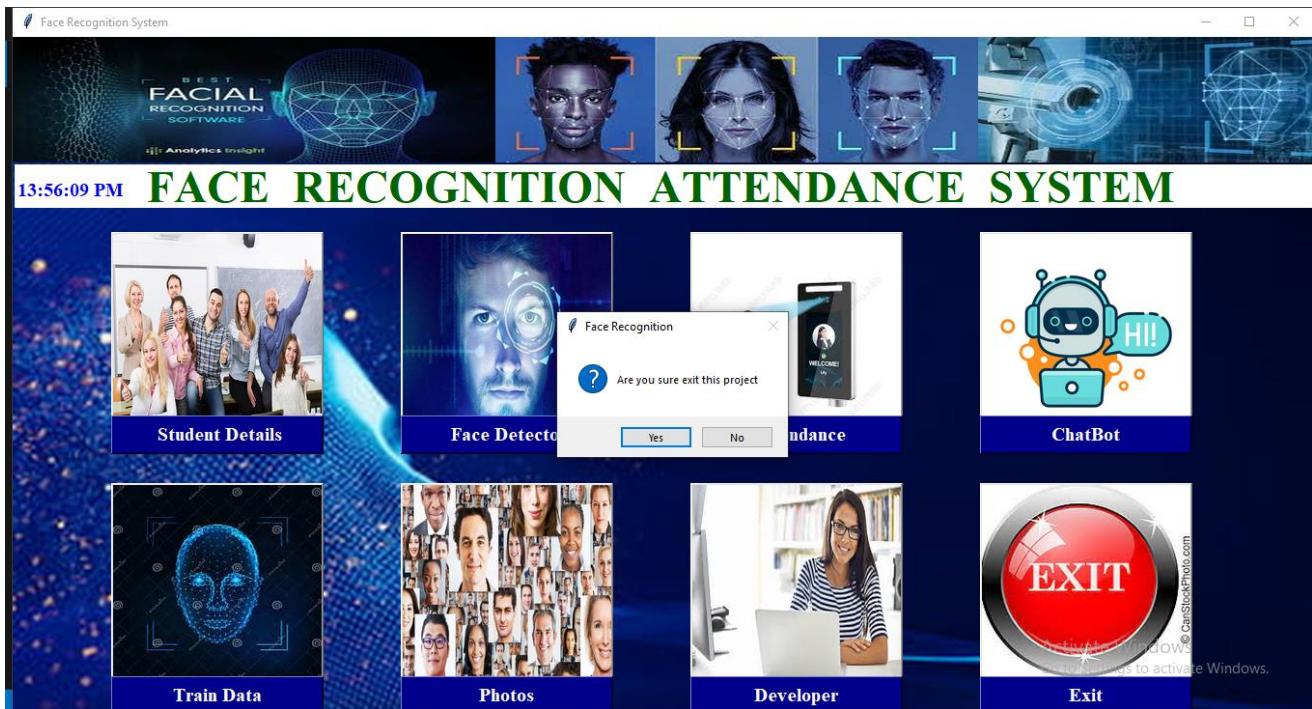
## 5.10 Developer Page



## 5.11 ChatBot Page



## 5.12 Exit Page



## 5.13 Database

### 5.13.1 Register

A screenshot of MySQL Workbench. The left sidebar shows the schema "mydata" with a single table "register". The main area shows the results of a query: "SELECT \* FROM mydata.register;". The result grid displays two rows of data:

faname	Iname	contact	email	securityQ	securityA	password
Payal	Jadhav	9975686768	payal@gmail.com	Your Birth Place	Satara	payalg
Riya	Jadhav	9964678424	riya@gmail.com	Your Birth Place	Khed	123

A message in the status bar says "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help." Another message at the bottom right says "Activate Windows" and "Skip Help".

## 5.13.2 Student Details

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas Tables Views Stored Procedures Functions mydata sys

Query 1 student register register student

1 • `SELECT * FROM face_recognizer.student;`

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types

Dep	Course	Year	Semester	Student_Id	Name	Roll	Gender	Dob	Email	Phone	Address
IT	Python	2023-24	Semester VI	1	Peyal	617	Female	5/28/03	peyal@gmail.com	7264964145	Satara
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Information Schema: mydata

Object Info Session Output

Activate Windows Context Help Snippets Go to settings to activate Windows.

# Conclusion and Future Work

## 6.1 Conclusion:

Capturing the images from camera or cc camera and applying techniques face detection and recognition can decrease the manual work from human and increase the security safety, taking the decision from this recognition result. Based on this face detection and recognition can used in implement so many application like automatic attendances system based on face recognition, worker attendances, security, safety, police application like finding thief in image that help to catching thief. In this system we have implemented an attendance system for a lecture, section or laboratory by which lecturer or teaching assistant a record student's attendance. It saves time and effort, especially if it is a lecture with huge number of students. This attendance system shows the use of facial recognition techniques for the purpose of student attendance and for the further process this record of student can be used in exam related issues.

## 6.2 Future Work:

Moving forward, there are several areas for potential future work and improvement in face recognition attendance systems using Python. These include:

- **Real-time Performance:** Optimization of the system to achieve real-time performance, allowing for seamless and instantaneous attendance marking.
- **User Experience:** Further refinement of the user interface to enhance user experience and accessibility, potentially incorporating features such as mobile applications for on-the-go attendance management.
- **Integration with Educational Systems:** Integration of the attendance system with educational management systems to streamline administrative processes and provide comprehensive attendance reporting.
- **Scalability:** Ensuring that the system can scale effectively to accommodate a growing number of users and locations.
- **Ethical Considerations:** Continued attention to ethical considerations surrounding facial recognition technology, including privacy protection and bias mitigation.

## References

1. <https://www.codewithharry.com/tutorial/python/>
2. <https://stackoverflow.com>
3. <https://www.studocu.com>
4. <https://opencv.org/opencv-face-recognition/>

