

Project Report
On
Face Recognition Attendance System
Submitted in partial fulfilment of the requirements for the award of the
degree of
Bachelor of Technology
in
Computer Science & Engineering
Session 2019-2023 program under the Maulana Abul Kalam Azad University
of Technology, West Bengal



Submitted By:

Priyanshu Jha (26000119032)
Nandalal Gorai (26000119016)
Mayank Singh (26000119034)
Kunal Kr. Roy (26000119027)

Department of Computer Science and Engineering

Bengal College of Engineering

Shilpo Kanan Road, Bidhannagar, Durgapur-713212

www.bcedgp.ac.in

DECLARATION

We, **PRIYANSHU JHA, NANDALAL GORAI, KUNAL KR. ROY and MAYANK SINGH**, hereby declare the work, which is being presented to the **BENGAL COLLEGE OF ENGINEERING** in the project, entitled “**FACIAL RECOGNITION ATTENDANCE SYSTEM**” in partial fulfillment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING**, is an authentic record of my own work carried out under the supervision of **MR. BIPLAB MAHAPATRA (Project Guide)**.

The matter embodied in this project has not been submitted by me or anybody else for the award of any other degree in any other University/Institution.

Date:

Priyanshu Jha (Roll No.:26000119032)
Nandalal Gorai (Roll No.:26000119016)
Mayank Singh (Roll No.:26000119034)
Kunal Kr. Roy (Roll No.:26000119027)

CERTIFICATE

Certified that **PRIYANSHU JHA** (Roll No. - 26000119032), **NANDALAL GORAI** (Roll No. – 26000119016), **KUNAL KR. ROY** (Roll No. – 26000119027) **MAYANK SINGH** (Roll No. - 26000119034) has carried out the project work presented in this report entitled “**Facial Recognition Attendance System**” for the award of **Bachelor of Technology in Computer Science & Engineering** from **Bengal College of Engineering (Formerly Bengal College of Engineering & Technology for Women), Durgapur** under my supervision. The report embodies results of original work, and studies are carried out by the student herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

MR. BIPLAB MAHAPATRA

Assistant Professor, CSE Deptt

(Project Guide)

(Bengal College of Engineering)

RAVI SINGH

(HOD, CSE Deptt.)

(Bengal College of Engineering)

ACKNOWLEDGEMENT

We are grateful to many people who have helped us throughout our B.Tech. First, we would like to thank my supervisor, Biplab Mahapatra and Head of Dept., for their intimacy and for their great support during all the steps of this report. We would also like to thank our faculty members and our colleagues for their valuable suggestions. Besides our colleagues, we appreciate the care and support of our good friends, who always encouraged us towards this research work.

We would like to thank the entire staff member for motivation, guidance, and support. We are indebted to a few friends and well -wishers who have extended their co-operation and help in the preparation of the project.

Finally, it goes without saying that we are deeply indebted to our parents for their support and their patient guidance.

Thanking You

Priyanshu Jha (Roll No.:26000119032)

Nandalal Gorai (Roll No.:26000119016)

Mayank Singh (Roll No.:26000119034)

Kunal Kr. Roy (Roll No.:26000119027)

CONTENTS

S no.	Topic	Pageno.
1.1	Abstract	1
1.2	Introduction	2
1.3	Objective	3
1.4	Literature/Survey Methodology	4-5
1.5	Proposed System	6-7
2	Feasibility Studies	8
3	Software Requirements	9
4	Flowchart	10-11
5	E-R Diagram	12-13
6	What is DFD? How to Prepare a DFD?	14
7	DFD's	15-16
8	Implementation/Result	17-19
9	Coding Section	20-61
10	Snapshots	62-70
11	Application of Facial Recognition System	71
12	Future Scope	72
13	Conclusion	73
14	Bibliography / References	74

List of all figures

S.no	Topic	Page. no.
1	Software Preconceives	7
2	Flowchart	11
3	Use Case Diagram	12
4	Activity Diagram	13
5	DFD 1	15
6	DFD 2	16
7	Implementation	17
8	Face Recognition Approach	18

1.1. ABSTRACT

The conventional attendance system consists of registers marked by teachers which leads to human error and a lot of maintenance. Time consumption is an important point of concern in this system. We have thought of revolutionizing it using available digital tools in the modern era i.e., FACE RECOGNITION. Our project will ensure more precision and negligible manual work. The project is revolutionized to overcome the problems of conventional systems. Face recognition and then marking the attendance is our project all about.

The database of all the students in the class is stored in a folder and when the face of the individual student matches with one of the faces stored image, attendance is marked else the face is ignored and attendance not marked. In our project, face recognition (Machine Learning) technology is used. Inside this Histogram of Oriented Gradient for face detection and SVM Classifier for name recognition is used. The model has an accuracy of 85.38% on the Labelled Faces in the Wild benchmark.

Keywords- Face Detection, Face Recognition, OpenCV, Tkinter etc.

1.2. INTRODUCTION

Attendance plays an important role in any organization whether it be educational institutions or companies. So, it is very important to keep a record of the attendance. The problem arises when one has to manually take the attendance which is not only time-consuming but exhausting as well.

So, an automatic attendance system can solve such a problem. Basically, there are two kinds of system:

- a) Manual Attendance System (MAS)
- b) Automated Attendance System (AAS)

One of AAS system is biometric technique using fingerprints, though it is automatic and a step ahead of traditional method it fails to meet the time and hygiene constraint. But using the biometric features of face solves such problem. Our projects emphasize the features of the face like ears, nose etc.

We used a method invented in 2005 called Histogram of Oriented Gradients (HOG) for face detection.

For identifying the name of a person, a simple linear SVM Classifier is used. All we need to do is train a classifier that can take in the measurements from a new test image and tell which known person is the closest match. The result is the name of person which is used to mark attendance.

Reduce manual process errors by providing automated and reliable attendance system that uses face recognition technology.

Admin can manage students (CRUD operations).

Produce monthly reports for students (Attendance Sheet).

Flexibility, Lectures capability of editing attendance records.

Send Email to students regarding important information. (If wanted)

1.3. OBJECTIVE

The main objective of a facial recognition attendance system is to provide a modern, efficient, and accurate method of tracking and recording attendance in various settings such as schools, workplaces, events, and other organizations.

One of the primary goals of such a system is to eliminate the need for traditional methods of attendance taking, such as manual sign-in sheets or swipe cards, which can be time-consuming, prone to errors, and susceptible to fraudulent activities like buddy punching (where one person clocks in or out on behalf of another). Facial recognition attendance systems aim to enhance security and prevent attendance fraud by ensuring that the person physically present is the same as the registered individual.

By utilizing advanced facial recognition algorithms, the system can capture and analyze unique facial features and patterns of individuals. When a person approaches the system, their face is detected, and the system matches it with the pre-registered database of authorized individuals. If a match is found, the attendance is recorded automatically, providing a quick and seamless process.

This technology offers several advantages. It can significantly reduce administrative tasks associated with attendance management, save time for both employees and supervisors, and minimize errors in attendance records. It also enhances security by preventing unauthorized access and providing a reliable audit trail. Additionally, it can generate real-time reports and analytics, enabling organizations to have better insights into attendance patterns and trends.

While privacy concerns and ethical considerations should be addressed, the objective of facial recognition attendance systems is to streamline attendance management, increase accuracy, and improve overall efficiency in various sectors where attendance tracking is crucial.

1.4. LITERATURE SURVEY/METHODOLOGY

Approach for Face Detection and Attendance Using OpenCV and Machine learning.

The Face detection has been implemented Using a Method Called Histogram of Oriented Gradients In this system students' images.

are stored in database folder With Students name. when Any person comes in front of camera it captures the image of person and compares the captured image with images present in database Folder if images match with any of the image in database folder, then the attendance of the student will be marked and stored in CSV file.

Marking attendance using face recognition

Automated Attendance system using Face recognition proposes that the system is based on face detection and recognition algorithm

which is used to detect the student's face when he/she come in front of camera and then compare the face with the images present in the folder if the match is found it will mark the attendance. This system has advantage over the traditional system as it saves time and there is no chance of proxy (that is no other student will mark the attendance of his/her friends).

Email notification for any Information

Attendance System proposes a feature of email notification by which users can get details about their attendance through email

on their respective google accounts.

No.	Existing System	Features	Benefits	Limitations
1.	Automated attendance management system using face recognition.	Use Eigen faces for Recognition	High accuracy	Multiple faces were not recognized.
2.	Face recognition attendance system	Stores the faces that are detected and automatically marks attendance	Used for security purposes in organizations	Don't recognize properly in poor light.
3.	Smart Attendance System using OPENCV based on Facial Recognition	Takes pictures through the webcam and create a dataset for users using mimages. Takes real-time images and mark attendance	Used for marking attendance in schools and colleges.	Cannot mark attendance of the student on a remote server database.

4.	Smart Attendance Management System Using Face Recognition	Student Registration Face Recognition Addition of subject with their corresponding time. Attendance sheet generation and import to Excel. (xlsx) format.	In this the data is stored in sorted manner so that it can easily accessible	Required high-definition camera
5.	Face Recognition - A Tool for Automated Attendance	Face detection, Pre-processing, Feature extraction, and Classification stages	High accuracy	Camera should be attached at a specific position
6.	Smart Application for AMS Using Face Recognition	Uses CCTV and Android mobile	3D face recognition algorithm is used	Android phone is expensive and detects one face at a time
7.	Student Attendance System in Classroom Using Face Recognition Technique	Use of Discrete Wavelet Transform and Discrete Cosine Transform.	Multiple face detection was possible	Success rate is only 82%
8.	Attendance System based on Face Recognition using Eigenface and PCA Algorithms	In this Illumination invariant algorithm is used	The problem of light intensity problem and head pose was overcome.	Masked faces were not recognized.
9.	Attendance System Using Face Recognition and Class Monitoring System	Open CV python library is used and MySQL is used for database	This method is fast and secure and has a low false positive rate.	Recognition rate is lower
10.	Algorithm for Efficient Attendance Management: Face Recognition based approach	Median filter and skin classification is used	Multiple faces can be detected at a time and no special hardware is needed	Accuracy is low only 50% faces were recognized

1.5. PROPOSED SYSTEM

The main task of our proposed system is to detect and recognize the image of the student and mark the attendance accordingly in the excel file. Also, can capture the new entries if needed. Further our system can perform all the basic operations like create, read, delete, edit, search etc. The proposed system is divided into major 3 modules which are as follows:

Admin Module

In this module, one has to provide the login credentials which involves id and password which will be matched with the one that is stored in database.

Student Detail Module

Student details like enrollment, name etc. can be edited, added, update, delete and can search student based on details.

Attendance Module

This will mark the attendance if the face of student match with the database else not.

Algorithm used is **LBPH**, here are the steps are as follows:

Step 1: Load the input image containing the face you want to process.

Step 2: Convert the image to grayscale to simplify the processing.

Step 3: Divide the grayscale image into small, equally sized cells or blocks. The size of the cells can vary depending on the application.

Step 4: For each cell, calculate the Local Binary Pattern (LBP) value. LBP compares the intensity of the central pixel with its neighboring pixels and encodes the result into a binary code. The most common way to calculate LBP is as follows:

- a. Select a central pixel in the cell.
- b. Compare the intensity of the central pixel with its neighboring pixels in a circular pattern.
- c. For each neighboring pixel, if its intensity is greater or equal to the central pixel's intensity, assign a binary value of 1; otherwise, assign a binary value of 0.

d. Combine the binary values for all neighboring pixels in a clockwise order to obtain the LBP code for the central pixel.

Step 5: After obtaining the LBP code for each central pixel in the cell, convert it to decimal representation. This decimal value represents the texture pattern within the cell.

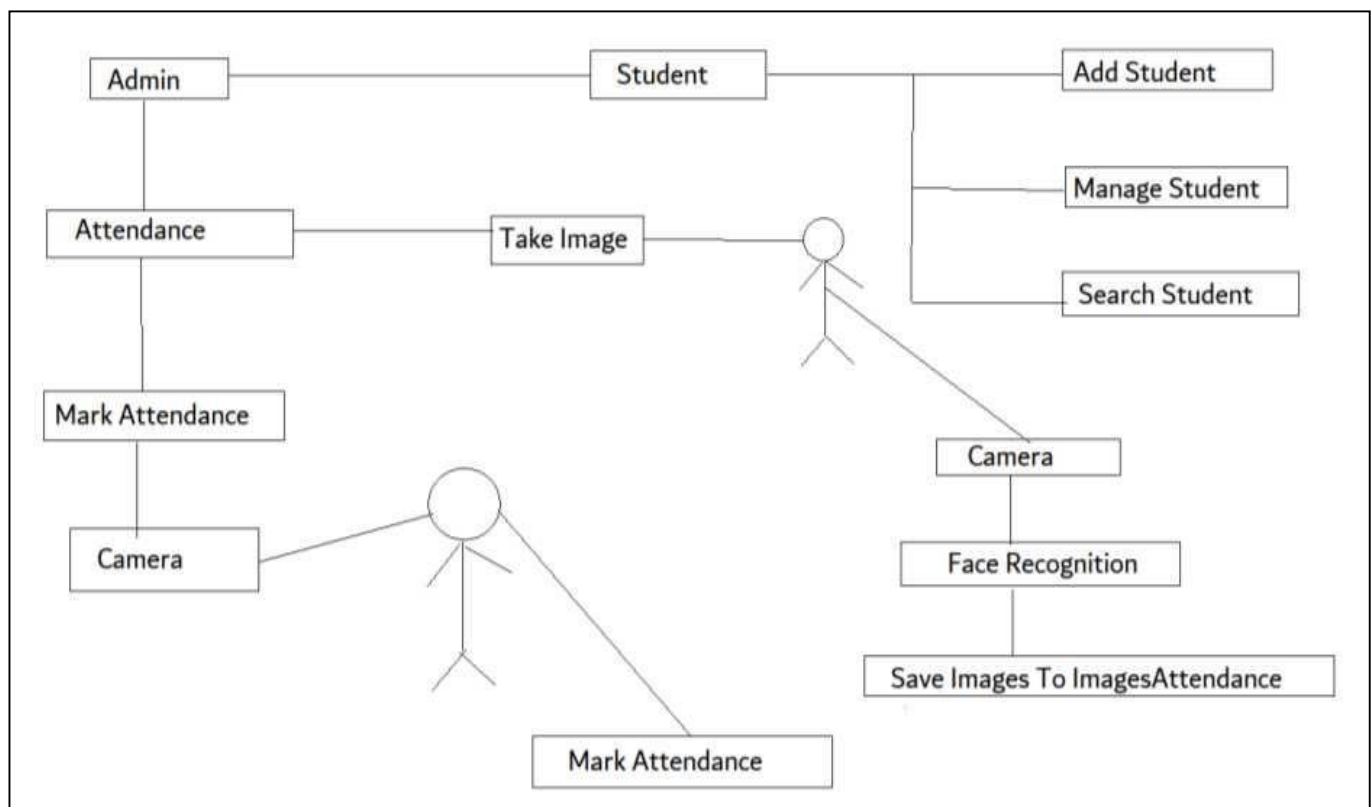
Step 6: Concatenate the decimal values from all cells to create a feature vector that represents the entire face image.

Step 7: Optionally, normalize the feature vector to reduce the effect of illumination variations.

Step 8: Use the feature vector for further processing, such as face recognition or classification.

Step 9: End

Software Preconceives



2. FEASIBILITY STUDY

The feasibility study of a facial recognition attendance system involves evaluating its practicality, viability, and potential success. Here are some key aspects typically considered in a feasibility study:

Technical Feasibility: Assessing the technological aspects of implementing a facial recognition attendance system, including the availability of reliable facial recognition algorithms, hardware requirements (such as cameras and processing power), compatibility with existing infrastructure, and the system's ability to handle the expected user load.

Economic Feasibility: Evaluating the financial viability of the system, including the initial investment required for hardware, software, and implementation, as well as ongoing maintenance costs. The study considers the potential cost savings and return on investment (ROI) in terms of reduced administrative efforts, improved accuracy, and prevention of attendance fraud.

Legal and Ethical Feasibility: Addressing legal and ethical considerations associated with facial recognition technology, such as compliance with data protection and privacy regulations, consent requirements, and potential concerns related to biases, discrimination, or misuse of personal data. Evaluating the system's ability to adhere to legal frameworks and ethical guidelines is crucial for its feasibility.

Operational Feasibility: Assessing the practicality of implementing and operating the system within the intended environment. This includes considering factors such as user acceptance and adoption, training requirements for administrators and users, potential disruptions to existing workflows, and the system's ability to integrate with other relevant systems and processes.

Security Feasibility: Evaluating the system's ability to ensure data security and prevent unauthorized access. This includes considering measures such as encryption, secure storage of facial data, access controls, and protection against potential hacking or misuse.

Scalability and Future Expansion: Assessing the system's scalability to accommodate growth in the number of users and locations, as well as its potential for future expansion or integration with other systems or applications.

By conducting a comprehensive feasibility study encompassing these aspects, organizations can make informed decisions regarding the implementation of a facial recognition attendance system, considering its technical, economic, legal, ethical, operational, and security aspects.

3. Software Requirement

Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below:

Hardware

- 32-bit, x86 Processing system,64-bit Computing Core at least 4 Compute Cores
- Intel i3 and above
- Internet connection
- High- definition Camera

Software

- Windows 10 or 11 later operating system or digital device for showing page
- Face Optimization Engine 3.0
- LBPH Algorithm
- DLSS
- Xtreme Shaders
- MySQL server

4. FLOWCHART

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan.

Symbols

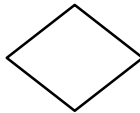
1. **Start/Stop:**



2. **Input:**



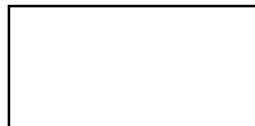
3. **Decision:**



4. **Connectore:**



5. **Process:**



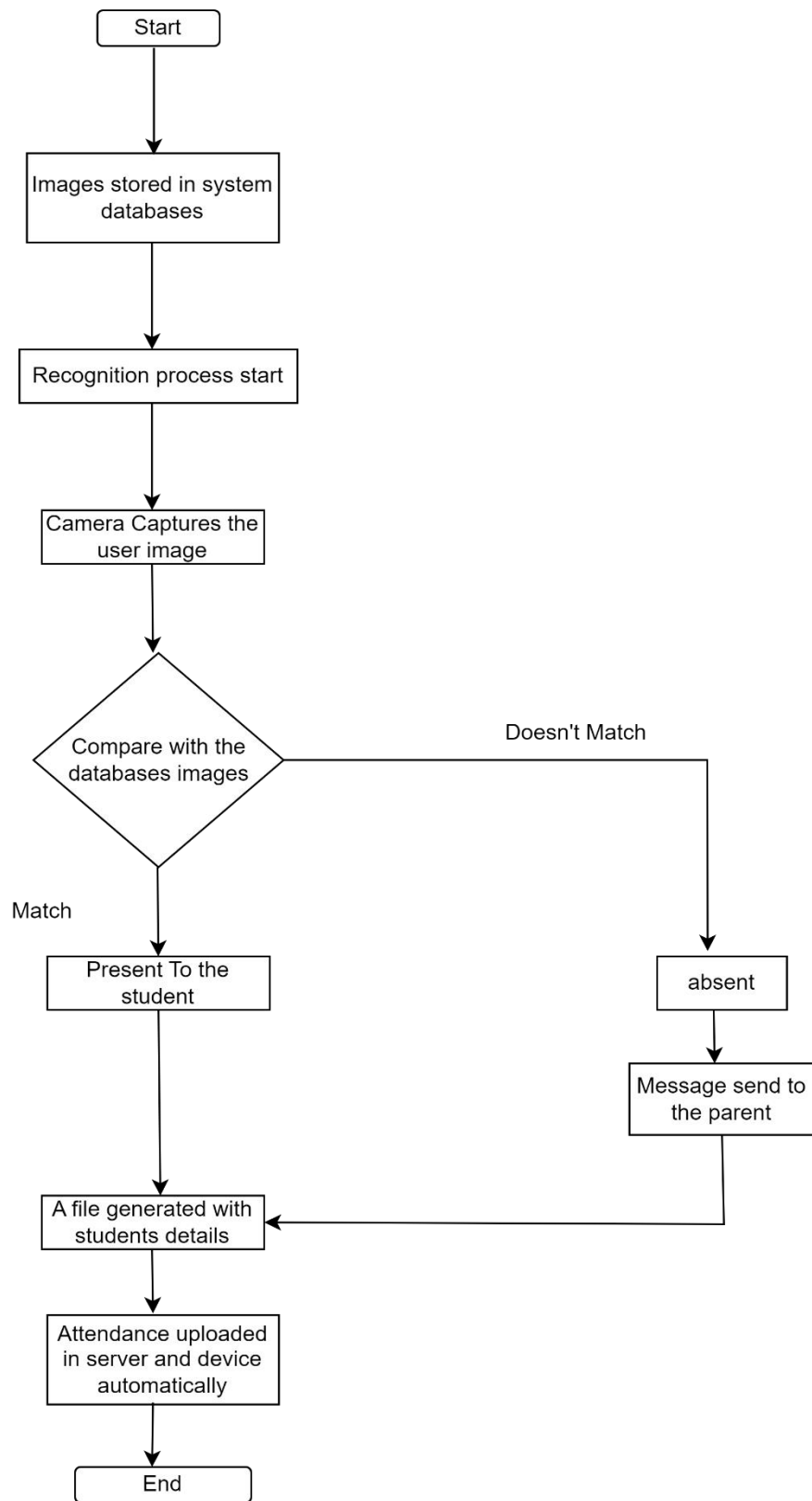


Fig :1

5. E-R DIAGRAM

USE-CASE DIAGRAM

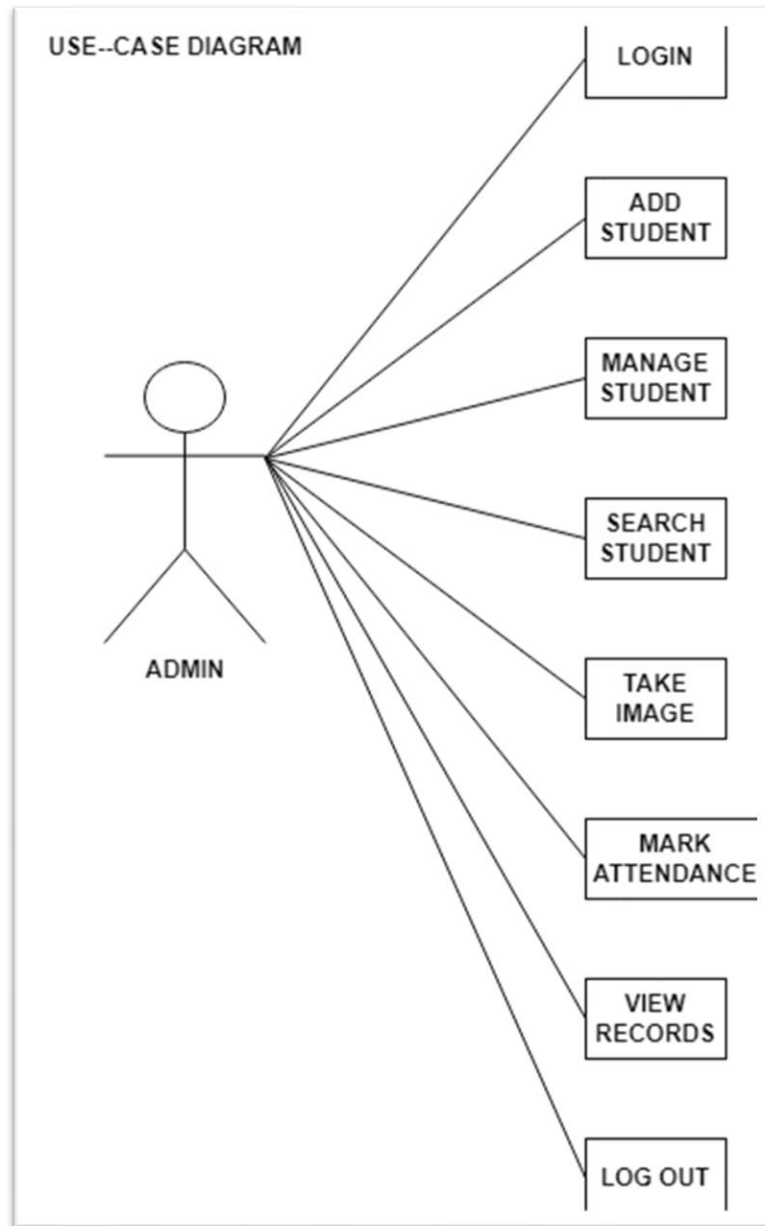


Fig. 2: Use Case Diagram

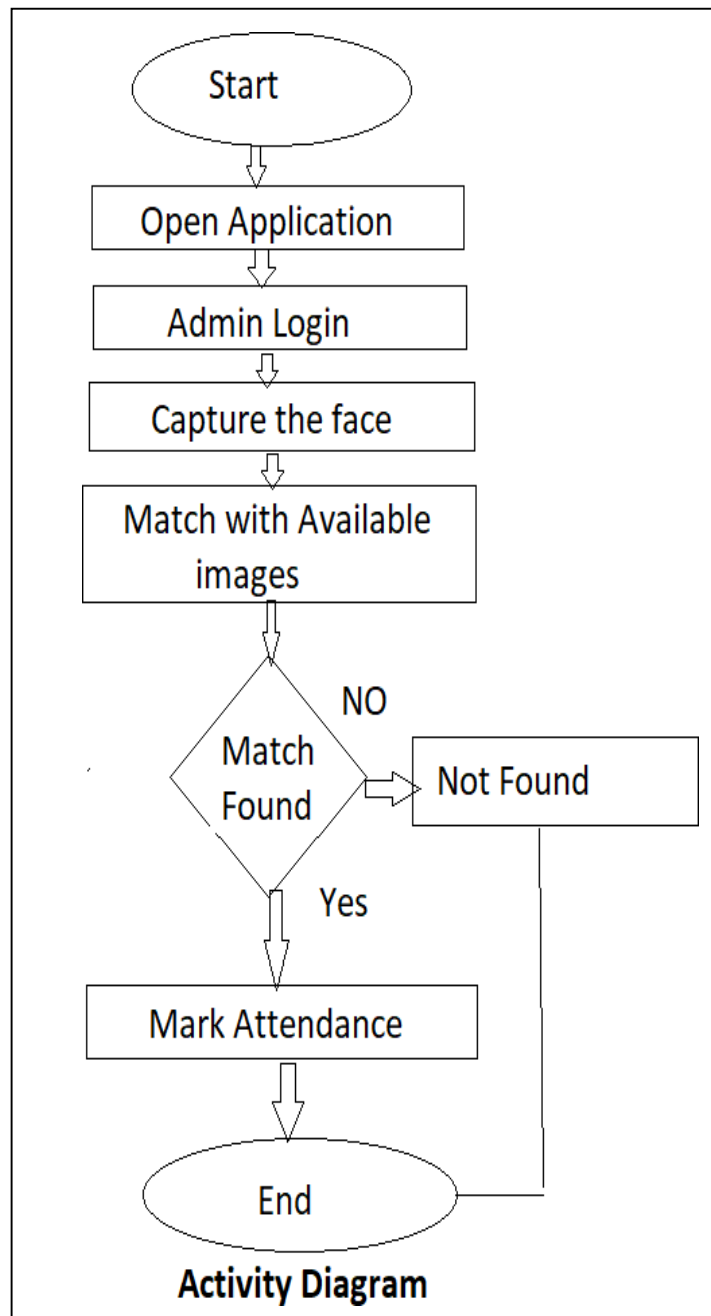


Fig :3

6. What is DFD/ How to prepare a DFD

DFD:

DFD is the abbreviation for Data Flow Diagram. The flow of data of a system or a process is represented by DFD. It also gives insight into inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decisions rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured analysis modeling tools.

How to prepare a DFD:

Rules for creating DFD :

- The name of the entity should be easy and understandable without any extra assistance (like comments).
- The processes should be numbered or put in ordered list to be referred easily.
- The DFD should maintain consistency across all the DFD levels.
- A single DFD can have maximum processes up to 9 and minimum 3 processes.

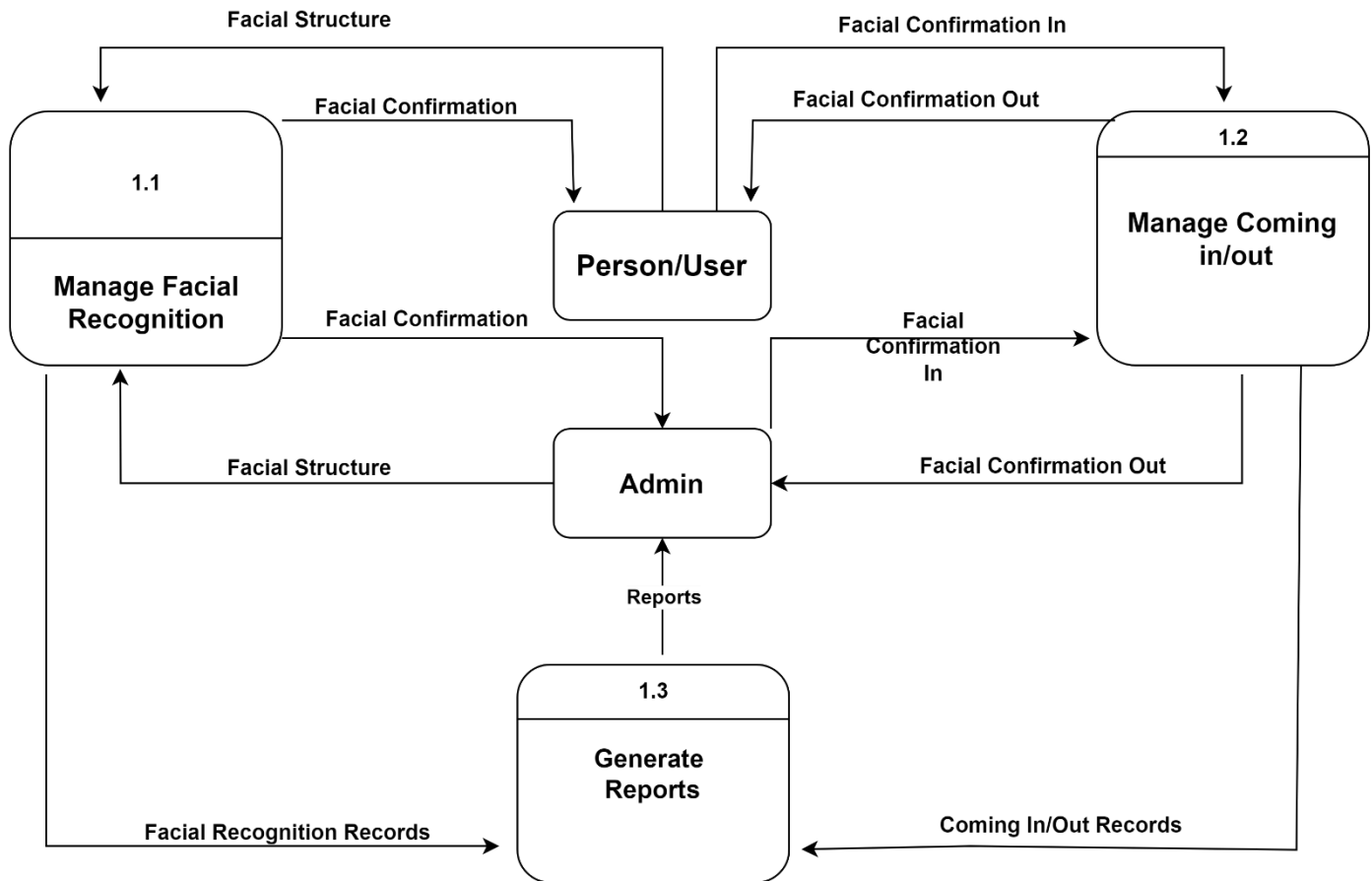
Levels of DFD:

DFD uses hierarchy to maintain transparency thus multilevel DFD's can be created

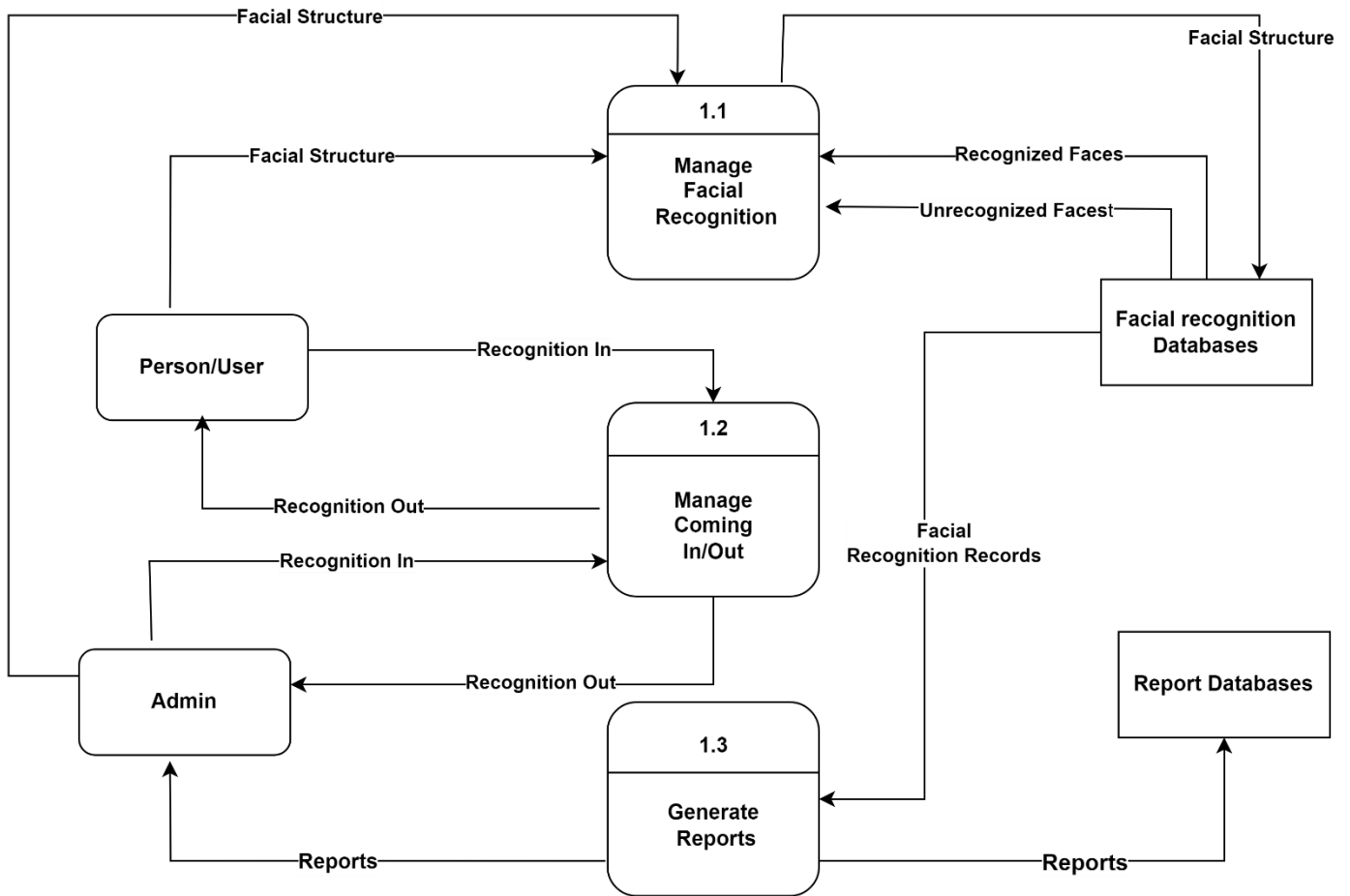
Levels of DFD are as follows:

- **0-level DFD**
- **1-level DFD**
- **2-level DFD**

7. DATA FLOW DIAGRAM (DFD)



DATA FLOW DIAGRAM LEVEL 1



DATA FLOW DIAGRAM LEVEL 2

8. IMPLEMENTATION AND RESULTS

One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build sophisticated vision applications quickly. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal faces are extracted from the image then saved to the file.

Technique Used: -

Data Science

Face recognition is a computer technology being used in a variety of applications that identifies human faces in digital images. The most advanced face recognition method, which is also employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.

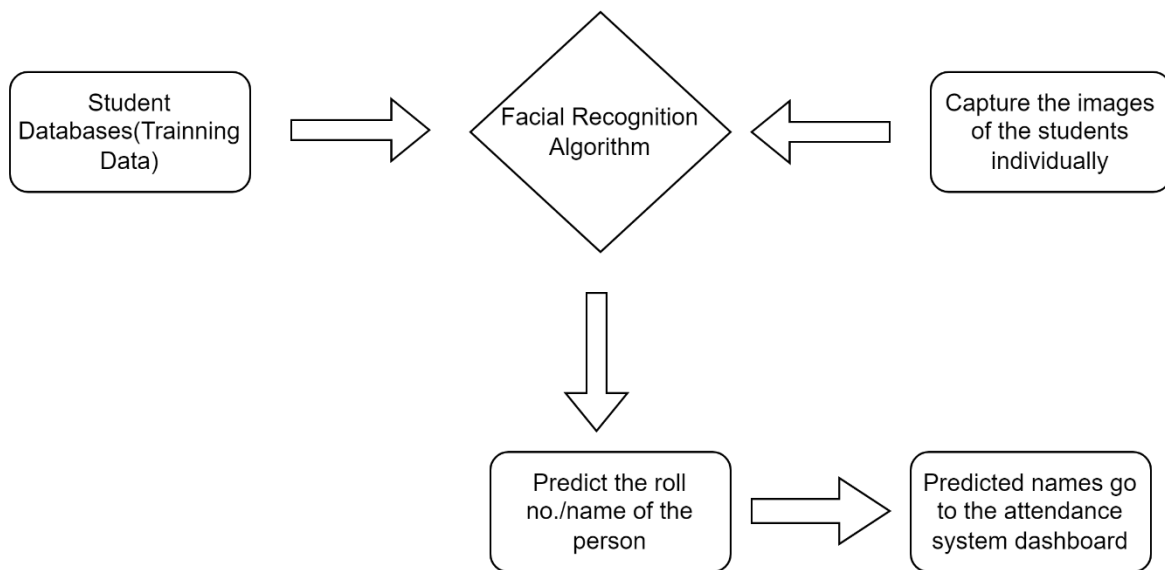


Fig. 4: Implementation

Approach

- Take an image from the webcam using Open CV.
- The face recognition model detects the face and saves it into the folder.
- At the time of attendance, it compares the current faces with the faces saved in the folder.
- If the match is found attendance was marked.

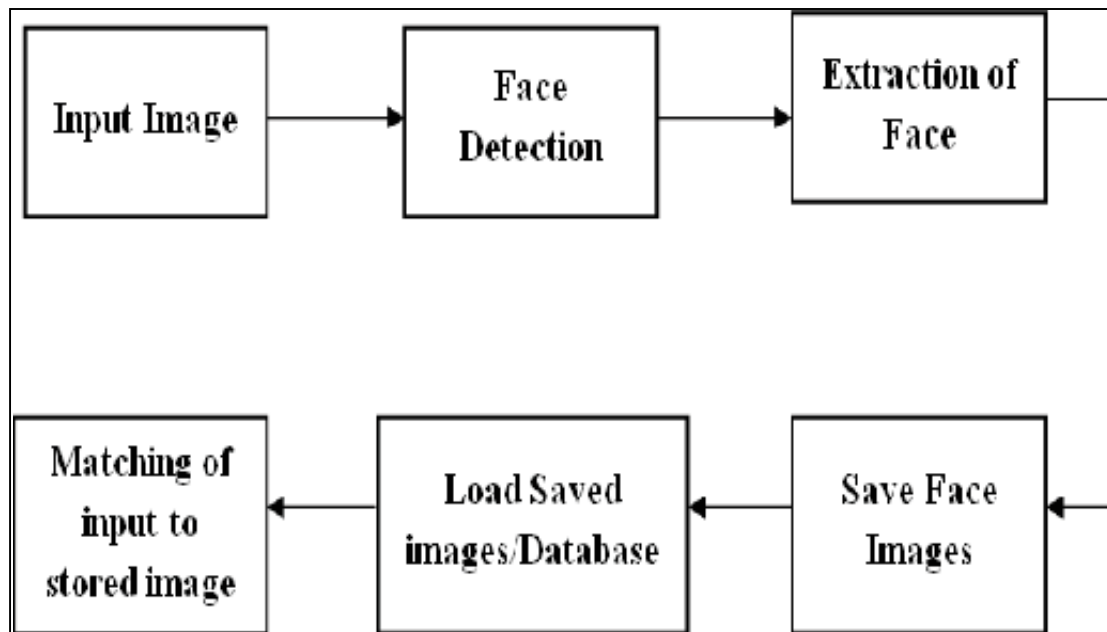


Fig. 5: Face Recognition Approach

Image Folder Name

Studentname.jpg

Tools Used

OpenCV Python

OpenCV (Open-Source Computer Vision Library) Used for Face Recognition and detection

Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

MySQL Server

MySQL is a free and opensource webserver developed by Oracle Corporation .It is one of the widely used platforms which helpsdevelopers to develop and test applications.

NumPy

NumPy is a python library used for working with arrays. It also has functions for working in domain of linear algebra, Fouriertransform, and matrices.

Steps for Face Recognition/Coding

First, look at a picture and find all the faces in it.

Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it isstill the same person.

Third, be able to pick out unique features of the face that you can use to tell it apart from other people - like how big the eyesare, how long the face is, etc.

Finally, compare the unique features of that face to all the people you already know to determine the person's name.

Mathematical Model

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

Where, Ψ = average image,

M = number of images, and

Γ_i = image vector. $\Phi_i = \Gamma_i - \Psi$

Where, $i = 1, 2, 3, \dots, M$. $A = [\Phi_1, \Phi_2, \Phi_3 \dots \Phi_M]$

9.CODING SECTION

Register.py

```
from tkinter import *
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector

class Register:
    def __init__(self,root):
        self.root = root
        self.root.title("Register")
        self.root.geometry("1350x800+0+0")

        #variables
        self.var_fname=StringVar()
        self.var_lname=StringVar()
        self.var_contact=StringVar()
        self.var_email=StringVar()
        self.var_securityQ=StringVar()
        self.var_securityA=StringVar()
        self.var_pass=StringVar()
        self.var_confpass=StringVar()

        #background image
        img = Image.open(r"C:\Users\ADMIN\OneDrive\Desktop\CLASS\FACE RECOG\Images\register_bg.jpg")
```

```

img=img.resize((1350,800),Image.Resampling.LANCZOS)
self.bg = ImageTk.PhotoImage(img)
bg_lbl=Label(self.root,image=self.bg)
bg_lbl.place(x=0,y=0,relwidth=1,relheight=1)

#left image
img1 = Image.open(r"C:\Users\ADMIN\OneDrive\Desktop\CLASS\FACE
RECOG\Images\register_left_img.jpg")
img1 = img1.resize((600,800),Image.Resampling.LANCZOS)
self.bg1 = ImageTk.PhotoImage(img1)
bg_lbl1=Label(self.root,image=self.bg1)
bg_lbl1.place(x=10,y=10,width=500,height=675)

frame = Frame(self.root,bg="white")
frame.place(x=500,y=10,width=800,height=675)

register_lbl = Label(frame,text="REGISTER",font=("times new
roman",20,"bold"),fg="green",bg="white")
register_lbl.place(x=20,y=20)

#Labels and entries
#first name
fname = Label(frame,text="First Name",font=("times new roman",15,"bold"),bg="white")
fname.place(x=50,y=100)
fname_entry = ttk.Entry(frame,textvariable=self.var_fname,font=("times new
roman",15,"bold"))
fname_entry.place(x=50,y=130,width=250)

#last name

```

```

lname = Label(frame,text="Last Name",font=("times new roman",15,"bold"),bg="white")
lname.place(x=370,y=100)

self.txt_lname = ttk.Entry(frame,textvariable=self.var_lname,font=("times new
roman",15,"bold"))
self.txt_lname.place(x=370,y=130,width=250)


#contact
contact = Label(frame,text="Contact No",font=("times new roman",15,"bold"),bg="white")
contact.place(x=50,y=170)

self.contact_entry = ttk.Entry(frame,textvariable=self.var_contact,font=("times new
roman",15,"bold"))
self.contact_entry.place(x=50,y=200,width=250)


#email
email = Label(frame,text="Email",font=("times new roman",15,"bold"),bg="white")
email.place(x=370,y=170)

self.email_entry = ttk.Entry(frame,textvariable=self.var_email,font=("times new
roman",15,"bold"))
self.email_entry.place(x=370,y=200,width=250)


#security questions
sec_ques = Label(frame,text="Select Security Question",font=("times new
roman",15,"bold"),bg="white",fg="black")
sec_ques.place(x=50,y=240)

self.combo_sec_ques=ttk.Combobox(frame,textvariable=self.var_securityQ,font=("times
new roman",15,"bold"),state="readonly")

self.combo_sec_ques["values"]=("Select","Your Birth Place","Your Pet Name","Your First
School")

self.combo_sec_ques.place(x=50,y=270,width=250)

self.combo_sec_ques.current(0)

```

```

#security answer

sec_ans = Label(frame,text="Security Answer",font=("times new
roman",15,"bold"),bg="white")

sec_ans.place(x=370,y=240)

self.sec_ans_entry = ttk.Entry(frame,textvariable=self.var_securityA,font=("times new
roman",15,"bold"))

self.sec_ans_entry.place(x=370,y=270,width=250)


#password

password = Label(frame,text="Password",font=("times new roman",15,"bold"),bg="white")

password.place(x=50,y=310)

self.password_entry = ttk.Entry(frame,textvariable=self.var_pass,font=("times new
roman",15,"bold"))

self.password_entry.place(x=50,y=340,width=250)


#conf_password

conf_pass = Label(frame,text="Confirm Password",font=("times new
roman",15,"bold"),bg="white")

conf_pass.place(x=370,y=310)

self.conf_pass_entry = ttk.Entry(frame,textvariable=self.var_confpass,font=("times new
roman",15,"bold"))

self.conf_pass_entry.place(x=370,y=340,width=250)


#buttons

img=Image.open(r"C:\Users\ADMIN\OneDrive\Desktop\CLASS\FACE
RECOG\Images\register_btn.png")

img=img.resize((400,200),Image.Resampling.LANCZOS)

self.photoimage=ImageTk.PhotoImage(img)

```

```
b1=Button(frame,image=self.photoimage,command=self.register_data,borderwidth=0,cursor="hand2",bg="white",activeforeground="white",activebackground="white")
```

```
b1.place(x=10,y=400,width=300)
```

```
img1=Image.open(r"C:\Users\ADMIN\OneDrive\Desktop\CLASS\FACE RECOG\Images\login_btn.png")
```

```
img1=img1.resize((300,150),Image.Resampling.LANCZOS)
```

```
self.photoimage1=ImageTk.PhotoImage(img1)
```

```
b2=Button(frame,image=self.photoimage1,borderwidth=0,cursor="hand2",bg="white",activeforeground="white",activebackground="white")
```

```
b2.place(x=350,y=420,width=350)
```

```
#Functions
```

```
def register_data(self):
```

```
    if self.var_fname.get()==" or self.var_email.get()==" or self.var_securityQ.get()=="Select":
```

```
        messagebox.showerror("Error","All fields are required")
```

```
    elif self.var_pass.get()!=self.var_confpass.get():
```

```
        messagebox.showerror("Error","Password & Confirm Password must be the same")
```

```
    else:
```

```
        conn = mysql.connector.connect(host="localhost",username="root",password="123456789",database="face_recognizer")
```

```
        cur = conn.cursor()
```

```
        query = ("select * from register where email = %s")
```

```
        value = (self.var_email.get(),)
```

```
        cur.execute(query, value)
```

```
        row = cur.fetchone()
```

```
        if row!=None:
```

```

        messagebox.showerror("Error", "User already registered")
    else:
        cur.execute("insert into register values(%s,%s,%s,%s,%s,%s,%s,%s)",(
            self.var_fname.get(),
            self.var_lname.get(),
            self.var_contact.get(),
            self.var_email.get(),
            self.var_securityQ.get(),
            self.var_securityA.get(),
            self.var_pass.get()
        ))
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Registered Successfully")

if __name__ == "__main__":
    root = Tk()
    app = Register(root)
    root.mainloop()

```

Login.py

```
from tkinter import *
from tkinter import ttk
from PIL import Image,ImageTk
from tkinter import messagebox

class Login_Window:

    def __init__(self,root):

        self.root = root

        self.root.title("Login")

        self.root.geometry("1350x800+0+0")


        img=Image.open(r"C:\Users\ADMIN\OneDrive\Desktop\CLASS\FACE
RECOG\Images\login_bg.jpg")

        img=img.resize((1350,800),Image.Resampling.LANCZOS)

        self.bg=ImageTk.PhotoImage(img)

        lbl_bg=Label(self.root,image=self.bg)

        lbl_bg.place(x=0,y=0,relwidth=1,relheight=1)


        frame=Frame(self.root,bg="black")

        frame.place(x=500,y=170,width=340,height=450)


        img1 = Image.open(r"C:\Users\ADMIN\OneDrive\Desktop\CLASS\FACE
RECOG\Images\login_icon1.png")

        img1=img1.resize((100,100),Image.Resampling.LANCZOS)

        self.photoimage1=ImageTk.PhotoImage(img1)

        lblimg1 = Label(image=self.photoimage1,bg="black",borderwidth=0)

        lblimg1.place(x=625,y=175,width=100,height=100)
```



```
get_str=Label(frame,text="Get Started",font=("times new  
roman",20,"bold"),fg="white",bg="black")
```

```
get_str.place(x=100,y=100)
```

```
username_lbl=Label(frame,text="Username",font=("times new  
roman",15,"bold"),fg="white",bg="black")
```

```
username_lbl.place(x=40,y=155)
```

```
self.txtuser=ttk.Entry(frame,font=("times new roman",15,"bold"))
```

```
self.txtuser.place(x=40,y=190,width=270)
```

```
password_lbl=Label(frame,text="Password",font=("times new  
roman",15,"bold"),fg="white",bg="black")
```

```
password_lbl.place(x=40,y=220)
```

```
self.txtpass=ttk.Entry(frame,font=("times new roman",15,"bold"))
```

```
self.txtpass.place(x=40,y=255,width=270)
```

```
#login button
```

```
loginbtn=Button(frame,text="Login",font=("times new  
roman",15,"bold"),command=self.login,bd=3,relief=RIDGE,fg="white",bg="orange",activeforeg  
round="white",activebackground="orange")
```

```
loginbtn.place(x=110,y=300,width=120,height=35)
```

```
#register button
```

```
registerbtn=Button(frame,text="New User",font=("times new  
roman",15,"bold"),borderwidth=0,fg="white",bg="lime  
green",activeforeground="white",activebackground="lime green")
```

```
registerbtn.place(x=40,y=350,width=120,height=35)
```

```
#forgot button
```

```
forgotbtn=Button(frame,text="Forgot Password",font=("times new  
roman",15,"bold"),borderwidth=0,fg="white",bg="red",activeforeground="white",activebackgro  
und="red")
```

```
forgotbtn.place(x=40,y=390,width=200,height=35)
```

```
def login(self):
```

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

```
        messagebox.showerror("Error","all fields are required")
```

```
    elif self.txtuser.get()=="abcd" or self.txtpass.get()=="abcd":
```

```
        messagebox.showinfo("Success","Login Successful")
```

```
    else:
```

```
        messagebox.showerror("Error","Invalid Data")
```

```
if __name__ == "__main__":
```

```
    root=Tk()
```

```
    app = Login_Window(root)
```

```
    root.mainloop()
```

Main.py

```
from tkinter import*
from tkinter import ttk
import tkinter
import webbrowser
from PIL import Image,ImageTk
from student import Student
from train import Train
from Attendance import attendance
from face_recognition import Face_Recognition
from developer import Developer
import os

class Face_Recognition_System:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1500x1080+0+0")
        self.root.title("Face Recognition System")
        #first image
        img=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
        img=img.resize((500,130),Image.Resampling.LANCZOS)
        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 image
        img1=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
```

```

img1=img1.resize((500,130),Image.Resampling.LANCZOS)
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 image
img2=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
img2=img2.resize((500,130),Image.Resampling.LANCZOS)
self.photoimg2=ImageTk.PhotoImage(img2)

f_lbl=Label(self.root,image=self.photoimg2)
f_lbl.place(x=1000,y=0,width=500,height=130)

#bg image
img3=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\main_background.jpg")
img3=img3.resize((1530,710),Image.Resampling.LANCZOS)
self.photoimg3=ImageTk.PhotoImage(img3)

bg_img=Label(self.root,image=self.photoimg3)
bg_img.place(x=0,y=130,width=1530,height=710)

#label title
title_lbl=Label(bg_img, text="FACE RECOGNITION ATTENDANCE SYSTEM",
font=("times new roman",35,"bold"),bg="white",fg="red")
title_lbl.place(x=0,y=0,width=1400,height=45)

#student button

```

```

img4=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\student_icon.png")

img4=img4.resize((220,220),Image.Resampling.LANCZOS)

self.photoimg4=ImageTk.PhotoImage(img4)


b1=Button(bg_img,image=self.photoimg4,command=self.student_details,cursor="hand2")

b1.place(x=100, y=60, width=220,height=220)


b1_1=Button(bg_img,text="Student
Details",command=self.student_details,cursor="hand2", font=("times new
roman",20,"bold"),bg="darkblue",fg="white")

b1_1.place(x=100, y=270, width=220,height=40)


#Detect Face

img5=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\detect_face.png")

img5=img5.resize((220,220),Image.Resampling.LANCZOS)

self.photoimg5=ImageTk.PhotoImage(img5)


b2=Button(bg_img,image=self.photoimg5,command=self.face_data,cursor="hand2")

b2.place(x=400, y=60, width=220,height=220)


b2_2=Button(bg_img,text="Detect Face",command=self.face_data,cursor="hand2",
font=("times new roman",20,"bold"),bg="darkblue",fg="white")

b2_2.place(x=400, y=270, width=220,height=40)


#Attendance Button

img6=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\attendance_button.jpg")

img6=img6.resize((220,220),Image.Resampling.LANCZOS)

```

```

self.photoimg6=ImageTk.PhotoImage(img6)

b3=Button(bg_img,image=self.photoimg6,command=self.attendance_data,cursor="hand2")
b3.place(x=700, y=60, width=220,height=220)

b3_3=Button(bg_img,text="Attendance",command=self.attendance_data,cursor="hand2",
font=("times new roman",20,"bold"),bg="darkblue",fg="white")
b3_3.place(x=700, y=270, width=220,height=40)

#Help Desk Button
img7=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\help_desk.png")
img7=img7.resize((220,220),Image.Resampling.LANCZOS)
self.photoimg7=ImageTk.PhotoImage(img7)

def openlink():
    webbrowser.open("https://forms.gle/UTwRxMJfZiDsWctLA")

b4=Button(bg_img,image=self.photoimg7,cursor="hand2",command= openlink)
b4.place(x=1000, y=60, width=220,height=220)

b4_4=Button(bg_img,text="Help Desk",cursor="hand2", font=("times new
roman",20,"bold"),bg="darkblue",fg="white", command = openlink)
b4_4.place(x=1000, y=270, width=220,height=40)

#second row Button

#Train Face Button
img8=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\train_face.jpg")
img8=img8.resize((220,220),Image.Resampling.LANCZOS)

```

```

self.photoimg8=ImageTk.PhotoImage(img8)

b5=Button(bg_img,image=self.photoimg8,command=self.training,cursor="hand2")
b5.place(x=100, y=320, width=220,height=220)

b5_5=Button(bg_img,text="Train Data",command=self.training,cursor="hand2",
font=("times new roman",20,"bold"),bg="darkblue",fg="white")
b5_5.place(x=100, y=520, width=220,height=40)

#Photos
img9=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\photos.png")
img9=img9.resize((220,220),Image.Resampling.LANCZOS)
self.photoimg9=ImageTk.PhotoImage(img9)

b6=Button(bg_img,image=self.photoimg9,cursor="hand2",command=self.open_img)
b6.place(x=400, y=320, width=220,height=220)

b6_6=Button(bg_img,text="Photos",cursor="hand2",command=self.open_img,
font=("times new roman",20,"bold"),bg="darkblue",fg="white")
b6_6.place(x=400, y=520, width=220,height=40)

#Developer
img10=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\developer_icon.png")
img10=img10.resize((220,220),Image.Resampling.LANCZOS)
self.photoimg10=ImageTk.PhotoImage(img10)

b7=Button(bg_img,image=self.photoimg10,cursor="hand2",command=self.developer_data)
b7.place(x=700, y=320, width=220,height=220)

```

```
b7_7=Button(bg_img,text="Developer",cursor="hand2",command=self.developer_data,
font=("times new roman",20,"bold"),bg="darkblue",fg="white")
```

```
b7_7.place(x=700, y=520, width=220,height=40)
```

```
#Exit
```

```
img11=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\exit_icon.png")
```

```
img11=img11.resize((220,220),Image.Resampling.LANCZOS)
```

```
self.photoimg11=ImageTk.PhotoImage(img11)
```

```
b8=Button(bg_img,image=self.photoimg11,cursor="hand2", command = self.isExit)
```

```
b8.place(x=1000, y=320, width=220,height=220)
```

```
b8_8=Button(bg_img,text="Exit",cursor="hand2", font=("times new
roman",20,"bold"),bg="darkblue",fg="white", command= self.isExit)
```

```
b8_8.place(x=1000, y=520, width=220,height=40)
```

```
def open_img(self):
```

```
    os.startfile("data")
```

```
#Function Buttons
```

```
def student_details(self):
```

```
    self.new_window=Toplevel(self.root)
```

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

```
def training(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 isExit(self):
        self.isExit=tkinter.messagebox.askyesno("Face recognition system", "Are you sure want to
exit?")

        if self.isExit>0:
                self.root.destroy()

        else:
                return
if __name__ == "__main__":
        root=Tk()

        obj=Face_Recognition_System(root)

        root.mainloop()

```

Attendance.py

```
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

mydata = []
class attendance:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1357x790+0+0")
        self.root.title("Face Recognition System")

        #variables
        self.var_attend_id = StringVar()
        self.var_attend_roll = StringVar()
        self.var_attend_name = StringVar()
        self.var_attend_dep = StringVar()
        self.var_attend_time = StringVar()
        self.var_attend_date = StringVar()
        self.var_attend_attendance = StringVar()

        #first image
```



```

img=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
img=img.resize((500,130),Image.Resampling.LANCZOS)
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 image
img1=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
img1=img1.resize((500,130),Image.Resampling.LANCZOS)
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 image
img2=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
img2=img2.resize((500,130),Image.Resampling.LANCZOS)
self.photoimg2=ImageTk.PhotoImage(img2)

f_lbl=Label(self.root,image=self.photoimg2)
f_lbl.place(x=1000,y=0,width=500,height=130)

#bg image
img3=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\main_background.jpg")
img3=img3.resize((1530,710),Image.Resampling.LANCZOS)
self.photoimg3=ImageTk.PhotoImage(img3)

```

```

bg_img=Label(self.root,image=self.photoimg3)
bg_img.place(x=0,y=130,width=1530,height=710)

#label title
title_lbl=Label(bg_img, text="ATTENDANCE", font=("times new
roman",35,"bold"),bg="white",fg="red")
title_lbl.place(x=0,y=0,width=1400,height=45)

main_frame = Frame(bg_img,bd=2,bg="lightblue")
main_frame.place(x=0,y=50,width=1357,height=650)

#left_label_frame
Left_frame = LabelFrame(main_frame,bd=2,relief=RIDGE,text="Student Attendance
Details",font=("times new roman",12,"bold"),bg="lightblue",fg="black")
Left_frame.place(x=0,y=0,width=650,height=500)

left_inside_frame = LabelFrame(Left_frame,bd=2,relief=RIDGE,bg="lightblue")
left_inside_frame.place(x=0,y=0,width=630,height=400)

#labels and entries
#attendance ID
attendance_label = Label(left_inside_frame,text="Attendance ID:",font=("times new
roman",12,"bold"),bg="lightblue")
attendance_label.grid(row=0,column=0,padx=10,pady=5,sticky=W)

attendance_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_id,font=("times new
roman",12,"bold"))
attendance_entry.grid(row=0,column=1,padx=10,sticky=W)

```

```

#roll

roll_label =
Label(left_inside_frame,text="Roll:",font=("comicsansns",12,"bold"),bg="lightblue")

roll_label.grid(row=0,column=2,padx=10,pady=5,sticky=W)

roll_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_roll,font=("times new
roman",12,"bold"))

roll_entry.grid(row=0,column=3,padx=10,sticky=W)


#name

name_label = Label(left_inside_frame,text="Name:",font=("times new
roman",12,"bold"),bg="lightblue")

name_label.grid(row=1,column=0,padx=10,pady=5,sticky=W)

name_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_name,font=("times new
roman",12,"bold"))

name_entry.grid(row=1,column=1,padx=10,sticky=W)


#department

department_label = Label(left_inside_frame,text="Department:",font=("times new
roman",12,"bold"),bg="lightblue")

department_label.grid(row=1,column=2,padx=10,pady=5,sticky=W)

department_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_dep,font=("times new
roman",12,"bold"))

department_entry.grid(row=1,column=3,padx=10,sticky=W)


#time

time_label = Label(left_inside_frame,text="Time:",font=("times new
roman",12,"bold"),bg="lightblue")

time_label.grid(row=2,column=0,padx=10,pady=5,sticky=W)

```

```

time_entry =
tkk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_time,font=("times new
roman",12,"bold"))

time_entry.grid(row=2,column=1,padx=10,sticky=W)

#date

date_label = Label(left_inside_frame,text="Date:",font=("times new
roman",12,"bold"),bg="lightblue")

date_label.grid(row=2,column=2,padx=10,pady=5,sticky=W)

date_entry =
tkk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_date,font=("times new
roman",12,"bold"))

date_entry.grid(row=2,column=3,padx=10,sticky=W)

#attendance

attendanceLabel = Label(left_inside_frame,text="Attendance Status:",font=("times new
roman",12,"bold"),bg="lightblue")

attendanceLabel.grid(row=3,column=0)

self.atten_status=tkk.Combobox(left_inside_frame,width=20,textvariable=self.var_attend_attend
ance,font="comicsansns 11 bold",state="readonly")

self.atten_status["values"] = ("Status","Present","Absent")

self.atten_status.grid(row=3,column=1,pady=8)

self.atten_status.current(0)

#button frame

btn_frame = Frame(left_inside_frame,bd=2,relief=RIDGE,bg="lightblue")

btn_frame.place(x=0,y=250,width=700,height=40)

#import csv button

import_csv_btn = Button(btn_frame,text="Import
CSV",command=self.importCSV,width=13,font=("times new
roman",13,"bold"),bg="blue",fg="white")

```

```

import_csv_btn.grid(row=0,column=0)

#export csv button

export_csv_btn = Button(btn_frame,text="Export
CSV",command=self.exportCSV,width=13,font=("times new
roman",13,"bold"),bg="blue",fg="white")

export_csv_btn.grid(row=0,column=1)

#update button

update_btn = Button(btn_frame,text="Update",width=13,font=("times new
roman",13,"bold"),bg="green",fg="white")

update_btn.grid(row=0,column=2)

#reset button

reset_btn =
Button(btn_frame,text="Reset",command=self.reset_data,width=13,font=("times new
roman",13,"bold"),bg="red",fg="white")

reset_btn.grid(row=0,column=4)


#right_label_frame

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

Right_frame.place(x=660,y=0,width=680,height=500)


table_frame = Frame(Right_frame,bd=2,relief=RIDGE,bg="lightblue")

table_frame.place(x=0,y=0,width=680,height=480)


#scroll bar and tables

scroll_x = ttk.Scrollbar(table_frame,orient=HORIZONTAL)

scroll_y = ttk.Scrollbar(table_frame,orient=VERTICAL)


self.AttendanceReportTable =
ttk.Treeview(table_frame,column=("id","roll","name","department","time","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)

self.AttendanceReportTable.heading("id",text="Attendance ID")
self.AttendanceReportTable.heading("roll",text="Roll")
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="Attendance")

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)

def fetchData(self,rows):
    self.AttendanceReportTable.delete(*self.AttendanceReportTable.get_children())

```



```

for i in rows:

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


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)


def exportCSV(self):

    try:

        if len(mydata)<1:

            messagebox.showerror("Error","No Data Found",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","Data Exported 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_attend_id.set(rows[0])
    self.var_attend_roll.set(rows[1])
    self.var_attend_name.set(rows[2])
    self.var_attend_dep.set(rows[3])
    self.var_attend_time.set(rows[4])
    self.var_attend_date.set(rows[5])
    self.var_attend_attendance.set(rows[6])

def reset_data(self):
    self.var_attend_id.set("")
    self.var_attend_roll.set("")
    self.var_attend_name.set("")
    self.var_attend_dep.set("")
    self.var_attend_time.set("")
    self.var_attend_date.set("")
    self.var_attend_attendance.set("")
from tkinter import *
from PIL import Image,ImageTk
from tkinter import messagebox
import mysql.connector
import cv2
import os
import csv
from tkinter import filedialog

```

```

mydata = []

class attendance:

    def __init__(self,root):

        self.root=root

        self.root.geometry("1357x790+0+0")

        self.root.title("Face Recognition System")


    #variables

    self.var_attend_id = StringVar()

    self.var_attend_roll = StringVar()

    self.var_attend_name = StringVar()

    self.var_attend_dep = StringVar()

    self.var_attend_time = StringVar()

    self.var_attend_date = StringVar()

    self.var_attend_attendance = StringVar()


    #first image

    img=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")

    img=img.resize((500,130),Image.Resampling.LANCZOS)

    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 image

    img1=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")

    img1=img1.resize((500,130),Image.Resampling.LANCZOS)

```

```

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 image
img2=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\bg_img.jpg")
img2=img2.resize((500,130),Image.Resampling.LANCZOS)
self.photoimg2=ImageTk.PhotoImage(img2)

f_lbl=Label(self.root,image=self.photoimg2)
f_lbl.place(x=1000,y=0,width=500,height=130)

#bg image
img3=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\main_background.jpg")
img3=img3.resize((1530,710),Image.Resampling.LANCZOS)
self.photoimg3=ImageTk.PhotoImage(img3)

bg_img=Label(self.root,image=self.photoimg3)
bg_img.place(x=0,y=130,width=1530,height=710)

#label title
title_lbl=Label(bg_img, text="ATTENDANCE", font=("times new
roman",35,"bold"),bg="white",fg="red")
title_lbl.place(x=0,y=0,width=1400,height=45)

main_frame = Frame(bg_img,bd=2,bg="lightblue")
main_frame.place(x=0,y=50,width=1357,height=650)

```

```

#left_label_frame

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

Left_frame.place(x=0,y=0,width=650,height=500)


left_inside_frame = LabelFrame(Left_frame,bd=2,relief=RIDGE,bg="lightblue")
left_inside_frame.place(x=0,y=0,width=630,height=400)


#labels and entries

#attendance ID

attendance_label = Label(left_inside_frame,text="Attendance ID:",font=("times new
roman",12,"bold"),bg="lightblue")

attendance_label.grid(row=0,column=0,padx=10,pady=5,sticky=W)

attendance_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_id,font=("times new
roman",12,"bold"))

attendance_entry.grid(row=0,column=1,padx=10,sticky=W)


#roll

roll_label =
Label(left_inside_frame,text="Roll:",font=("comicansns",12,"bold"),bg="lightblue")

roll_label.grid(row=0,column=2,padx=10,pady=5,sticky=W)

roll_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_roll,font=("times new
roman",12,"bold"))

roll_entry.grid(row=0,column=3,padx=10,sticky=W)


#name

```

```

name_label = Label(left_inside_frame,text="Name:",font=("times new
roman",12,"bold"),bg="lightblue")

name_label.grid(row=1,column=0,padx=10,pady=5,sticky=W)

name_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_name,font=("times new
roman",12,"bold"))

name_entry.grid(row=1,column=1,padx=10,sticky=W)


#department

department_label = Label(left_inside_frame,text="Department:",font=("times new
roman",12,"bold"),bg="lightblue")

department_label.grid(row=1,column=2,padx=10,pady=5,sticky=W)

department_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_dep,font=("times new
roman",12,"bold"))

department_entry.grid(row=1,column=3,padx=10,sticky=W)


#time

time_label = Label(left_inside_frame,text="Time:",font=("times new
roman",12,"bold"),bg="lightblue")

time_label.grid(row=2,column=0,padx=10,pady=5,sticky=W)

time_entry =
ttk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_time,font=("times new
roman",12,"bold"))

time_entry.grid(row=2,column=1,padx=10,sticky=W)


#date

date_label = Label(left_inside_frame,text="Date:",font=("times new
roman",12,"bold"),bg="lightblue")

date_label.grid(row=2,column=2,padx=10,pady=5,sticky=W)

```

```

date_entry =
tkk.Entry(left_inside_frame,width=20,textvariable=self.var_attend_date,font=("times new
roman",12,"bold"))

date_entry.grid(row=2,column=3,padx=10,sticky=W)

#attendance

attendanceLabel = Label(left_inside_frame,text="Attendance Status:",font=("times new
roman",12,"bold"),bg="lightblue")

attendanceLabel.grid(row=3,column=0)

self.atten_status=tkk.Combobox(left_inside_frame,width=20,textvariable=self.var_attend_attend
ance,font="comicsansns 11 bold",state="readonly")

self.atten_status["values"] = ("Status","Present","Absent")

self.atten_status.grid(row=3,column=1,pady=8)

self.atten_status.current(0)

#button frame

btn_frame = Frame(left_inside_frame,bd=2,relief=RIDGE,bg="lightblue")

btn_frame.place(x=0,y=250,width=700,height=40)

#import csv button

import_csv_btn = Button(btn_frame,text="Import
CSV",command=self.importCSV,width=13,font=("times new
roman",13,"bold"),bg="blue",fg="white")

import_csv_btn.grid(row=0,column=0)

#export csv button

export_csv_btn = Button(btn_frame,text="Export
CSV",command=self.exportCSV,width=13,font=("times new
roman",13,"bold"),bg="blue",fg="white")

export_csv_btn.grid(row=0,column=1)

#update button

update_btn = Button(btn_frame,text="Update",width=13,font=("times new
roman",13,"bold"),bg="green",fg="white")

```

```

update_btn.grid(row=0,column=2)

#reset button

reset_btn =
Button(btn_frame,text="Reset",command=self.reset_data,width=13,font=("times new
roman",13,"bold"),bg="red",fg="white")

reset_btn.grid(row=0,column=4)


#right_label_frame

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

Right_frame.place(x=660,y=0,width=680,height=500)


table_frame = Frame(Right_frame,bd=2,relief=RIDGE,bg="lightblue")

table_frame.place(x=0,y=0,width=680,height=480)


#scroll bar and tables

scroll_x = ttk.Scrollbar(table_frame,orient=HORIZONTAL)

scroll_y = ttk.Scrollbar(table_frame,orient=VERTICAL)


self.AttendanceReportTable =
ttk.Treeview(table_frame,column=("id","roll","name","department","time","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)


self.AttendanceReportTable.heading("id",text="Attendance ID")

self.AttendanceReportTable.heading("roll",text="Roll")

```



```

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="Attendance")


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)


def fetchData(self,rows):
    self.AttendanceReportTable.delete(*self.AttendanceReportTable.get_children())
    for i in rows:
        self.AttendanceReportTable.insert("",END,values=i)


def importCSV(self):
    global mydata
    mydata.clear()

    fln=filedialog.askopenfilename(initialdir=os.getcwd(),title="Open CSV",filetypes=(("CSV
File","*.csv"),("All File","*.txt")),parent=self.root)

    with open(fln) as myfile:

```

```

        csvread = csv.reader(myfile,delimiter=",")
        for i in csvread:
            mydata.append(i)
        self.fetchData(mydata)

def exportCSV(self):
    try:
        if len(mydata)<1:
            messagebox.showerror("Error","No Data Found",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","Data Exported 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_attend_id.set(rows[0])
    self.var_attend_roll.set(rows[1])
    self.var_attend_name.set(rows[2])
    self.var_attend_dep.set(rows[3])
    self.var_attend_time.set(rows[4])

```

```

        self.var_attend_date.set(rows[5])
        self.var_attend_attendance.set(rows[6])
def reset_data(self):
    self.var_attend_id.set("")
    self.var_attend_roll.set("")
    self.var_attend_name.set("")
    self.var_attend_dep.set("")
    self.var_attend_time.set("")
    self.var_attend_date.set("")
    self.var_attend_attendance.set("")

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

```

Face Recognize.py

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



```
class Face_Recognition:
    def __init__(self,root):
        self.root = root
        self.root.geometry("1357x790+0+0")
        self.root.title("Face Recognition System")

        title_lbl = Label(self.root,text="FACE DETECTION",font=("times new
roman",35,"bold"),bg="white",fg="red")
        title_lbl.place(x=0,y=0,width=1400,height=45)

        img_top = Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\face_detection.png")
        img_top = img_top.resize((650,700),Image.Resampling.LANCZOS)
        self.photoimg_top = ImageTk.PhotoImage(img_top)

        f_lbl = Label(self.root,image=self.photoimg_top)
        f_lbl.place(x=0,y=55,width=650,height=700)
```

```

img_top1 = Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\face_detection2.jpg")

img_top1 = img_top1.resize((950,700),Image.Resampling.LANCZOS)

self.photoimg_top1 = ImageTk.PhotoImage(img_top1)


f_lbl1 = Label(self.root,image=self.photoimg_top1)

f_lbl1.place(x=650,y=55,width=950,height=700)


b1_1=Button(self.root,text="DETECT
FACE",command=self.face_recog,cursor="hand2",font=("times new
roman",30,"bold"),bg="red",fg="white")

b1_1.place(x=425,y=600,width=500,height=40)


#ATTENDANCE

def mark_attendance(self,i,r,n,d):

    with open ("attendance.csv","r+",newline='\n') as f:

        myDataList = f.readlines()

        name_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},Present")

```

```

#face recognition
def face_recog(self):
    def draw_boundary(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)))

            conn =
mysql.connector.connect(host="localhost",username="root",password="123456789",database="face_recognizer")

            my_cursor = conn.cursor()
            my_cursor.execute("select Student_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 Dept 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,y-
75),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)

    cv2.putText(img,f"Roll:{r}",(x,y-
55),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)

    cv2.putText(img,f"Name:{n}",(x,y-
30),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)

    cv2.putText(img,f"Department:{d}",(x,y-
5),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)

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

else:

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

    cv2.putText(img,"Unknown face",(x,y-
5),cv2.FONT_HERSHEY_COMPLEX,0.8,(255,255,255),3)

    coord = [x,y,w,h]

    return coord

def recognize(img,clf,faceCascade):

    coord = draw_boundary(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()

```


Train Data.py

```
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

class Train:
    def __init__(self,root):
        self.root=root
        self.root.geometry("1357x790+0+0")
        self.root.title("Face Recognition System")

        title_lbl=Label(self.root, text="TRAIN DATA", font=("times new
roman",35,"bold"),bg="white",fg="red")
        title_lbl.place(x=0,y=0,width=1400,height=45)

        img_top=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\train_data1.jpg")
        img_top=img_top.resize((1400,325),Image.Resampling.LANCZOS)
        self.photoimg_top=ImageTk.PhotoImage(img_top)

        f_lbl=Label(self.root,image=self.photoimg_top)
        f_lbl.place(x=0,y=55,width=1400,height=325)

        #training button

        b1_1=Button(self.root,text="Train Data
Set",command=self.train_classifier,cursor="hand2", font=("times new
roman",20,"bold"),bg="orange",fg="black")
        b1_1.place(x=0, y=380, width=1400,height=60)
```



```

img_bottom=Image.open(r"D:\FINAL YEAR PROJECT\FACE
RECOG\Images\train_data2.jpg")

img_bottom=img_bottom.resize((1400,325),Image.Resampling.LANCZOS)

self.photoimg_bottom=ImageTk.PhotoImage(img_bottom)

f_lbl=Label(self.root,image=self.photoimg_bottom)

f_lbl.place(x=0,y=440,width=1400,height=325)
def train_classifier(self):
    data_dir = ("data")
    path = [os.path.join(data_dir,file) for file in os.listdir(data_dir)]
    faces=[]
    ids=[]
    for image in path:
        img=Image.open(image).convert('L')      #converting to grayscale
        imageNp=np.array(img,'uint8')
        id=int(os.path.split(image)[1].split('.')[1])
        faces.append(imageNp)
        ids.append(id)
        cv2.imshow("Training",imageNp)
        cv2.waitKey(1)==13
    ids=np.array(ids)
    #training classifiers
    clf = cv2.face.LBPHFaceRecognizer_create()
    clf.train(faces,ids)
    clf.write("classifier.xml")
    cv2.destroyAllWindows()

    messagebox.showinfo("Result","Training Datasets Completed")
if __name__ == "__main__":
    root=Tk()
    obj=Train(root)
    root.mainloop()

```



Exit.py

#Exit

```
img11=Image.open(r"D:\FINAL YEAR PROJECT\FACE RECOG\Images\exit_icon.png")
```

```
img11=img11.resize((220,220),Image.Resampling.LANCZOS)
```

```
self.photoimg11=ImageTk.PhotoImage(img11)
```

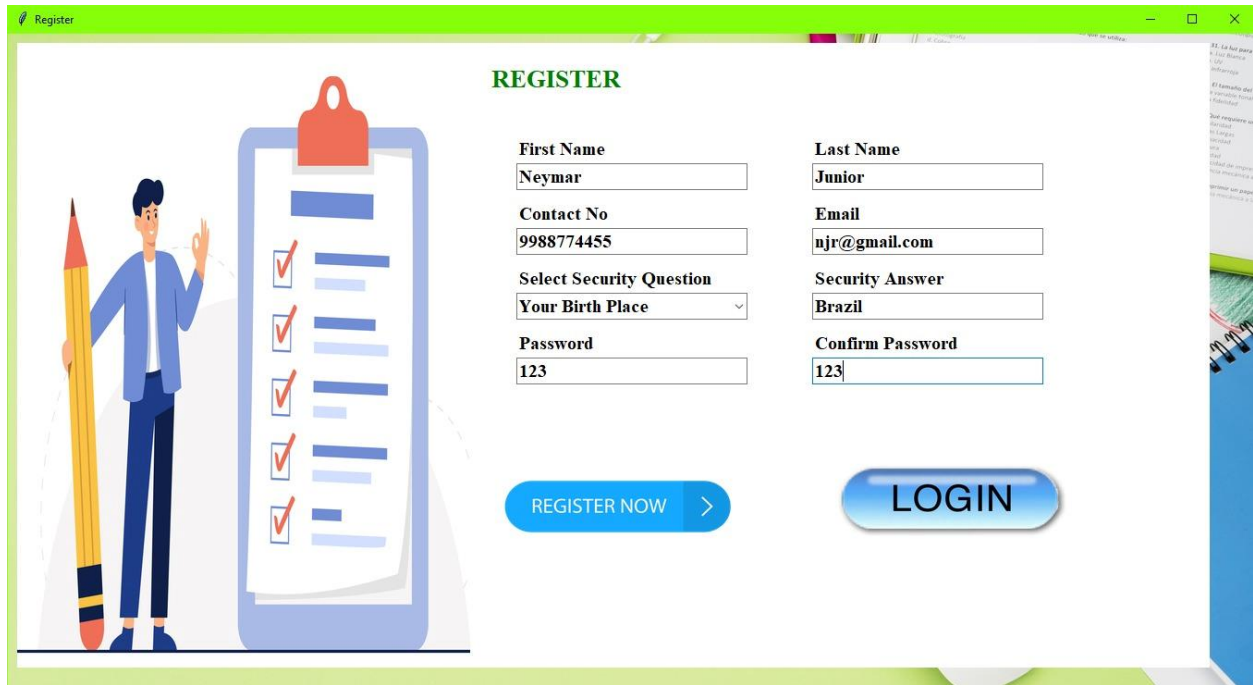
```
b8=Button(bg_img,image=self.photoimg11,cursor="hand2",  
command = self.isExit)
```

```
b8.place(x=1000, y=320, width=220,height=220)
```

```
b8_8=Button(bg_img,text="Exit",cursor="hand2", font=("times new  
roman",20,"bold"),bg="darkblue",fg="white", command= self.isExit)
```

```
b8_8.place(x=1000, y=520, width=220,height=40)
```

10. SNAPSHOTS



The screenshot shows a web browser window titled "Register". The page has a light green header and a white main area. On the left, there is an illustration of a man in a blue suit holding a large yellow pencil, standing next to a large blue clipboard with a red clip at the top. The clipboard has a checklist with five items, each marked with a red checkmark. The title "REGISTER" is displayed in green text. Below the illustration, there are two columns of form fields. The first column contains: "First Name" (text input with "Neymar"), "Contact No" (text input with "9988774455"), "Select Security Question" (dropdown menu with "Your Birth Place" selected), and "Password" (text input with "123"). The second column contains: "Last Name" (text input with "Junior"), "Email" (text input with "njr@gmail.com"), "Security Answer" (text input with "Brazil"), and "Confirm Password" (text input with "123"). At the bottom of the form, there are two buttons: a blue "REGISTER NOW" button with a right arrow and a blue "LOGIN" button.

REGISTER

First Name: Neymar

Last Name: Junior

Contact No: 9988774455

Email: njr@gmail.com

Select Security Question: Your Birth Place

Security Answer: Brazil

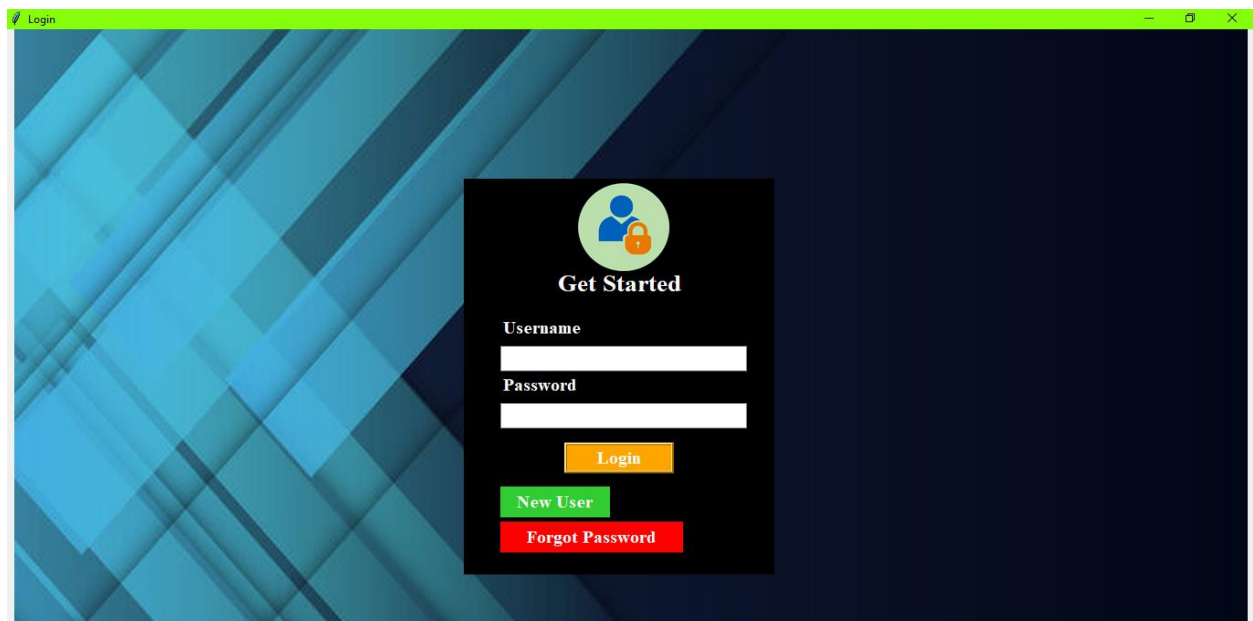
Password: 123

Confirm Password: 123

REGISTER NOW >

LOGIN

Register Module



The screenshot shows a web browser window titled "Login". The background is a dark blue gradient with a pattern of overlapping light blue squares. In the center, there is a black rectangular box with a white border. At the top of the box is a circular icon containing a blue person silhouette and a red padlock. Below the icon, the text "Get Started" is displayed in white. Underneath, there are two white text input fields labeled "Username" and "Password". Below the "Password" field is an orange "Login" button. Below the "Login" button are two more buttons: a green "New User" button and a red "Forgot Password" button.

Get Started

Username

Password

Login

New User

Forgot Password

Login

```
1 from tkinter import*
2 from tkinter import ttk
3 from PIL import Image,ImageTk
4
5 class Face_Recognition_System:
6     def __init__(self,root):
7         self.root=root
8         self.root.geometry("1530x790+0+0")
9         self.root.title("Face Recognition System")
10
11         img=Image.open(r"D:\Final Year project\Facial Recognition attendance\images\face_recog.jpg")
12         img=img.resize((510,130),Image.ANTIALIAS)
13         self.photoimg=ImageTk.PhotoImage(img)
14
15         f_lbl=Label(self.root,image=self.photoimg)
16         f_lbl.place(x=0,y=0,width=510,height=130)
17
18         img1=Image.open(r"D:\Final Year project\Facial Recognition attendance\images\download.jpg")
19         img1=img1.resize((530,130),Image.ANTIALIAS)
20         self.photoimg1=ImageTk.PhotoImage(img1)
21
22         f_lbl=Label(self.root,image=self.photoimg1)
23         f_lbl.place(x=510,y=0,width=530,height=130)
24
25         img2=Image.open(r"D:\Final Year project\Facial Recognition attendance\images\images.jpg")
26         img2=img2.resize((540,130),Image.ANTIALIAS)
27         self.photoimg2=ImageTk.PhotoImage(img2)
28
29         f_lbl=Label(self.root,image=self.photoimg2)
30         f_lbl.place(x=1000,y=0,width=500,height=130)
31
32
33         #bg image
34
35         img3=Image.open(r"D:\Final Year project\Facial Recognition attendance\images\bg.jpg")
36         img3=img3.resize((1530,710),Image.ANTIALIAS)
37         self.photoimg3=ImageTk.PhotoImage(img3)
```

Facial Recognition Dashboard Module Code



Dashboard Module

Face Recognition System

STUDENT REGISTRATION

Student Details

Current Course Info

Department
Select Department
Course
Select Course
Year
Select Year
Semester
Select Semester

Class Student Info

Student ID:
Student Name:

Class Division:
Select Division
Roll no:

Gender:
Select Gender
Date Of Birth:

Email:
Phone No:

Address:
Teacher Name:

☐ Take Photo Sample
☐ No Photo Sample

Save
Update
Delete
Reset
Take Photo
Update Photo

Student Details

Search System

Search By:
Select
Search
Show All

Department	Course Name	Year	Semester	Student ID	Student Name	
Computer	SE	2021-22	Sem-2	03	Biplab Mahapatra	A
Computer	BE	Select Year	Sem-2	1	Nandalal	A
Computer	BE	2022-23	Sem-3	2	Priyanshu	A
Computer	SE	2020-21	Sem-1	4	Ravi Sir	B

28°C
Haze

Search

1: 16.1 KB/s
1: 298.9 KB/s

ENG
IN

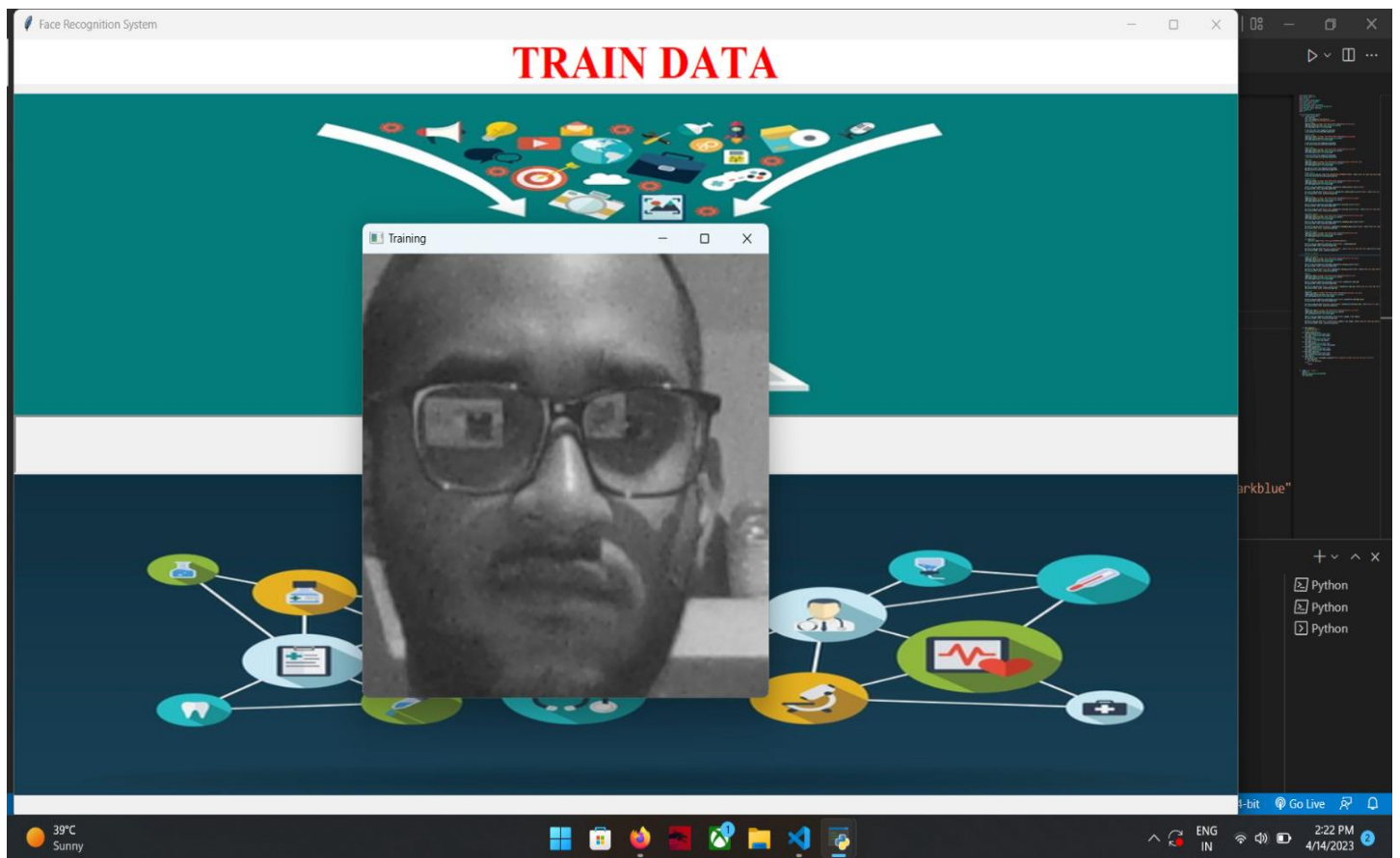
01:38 AM
15-05-2023

Student Management Detail Module

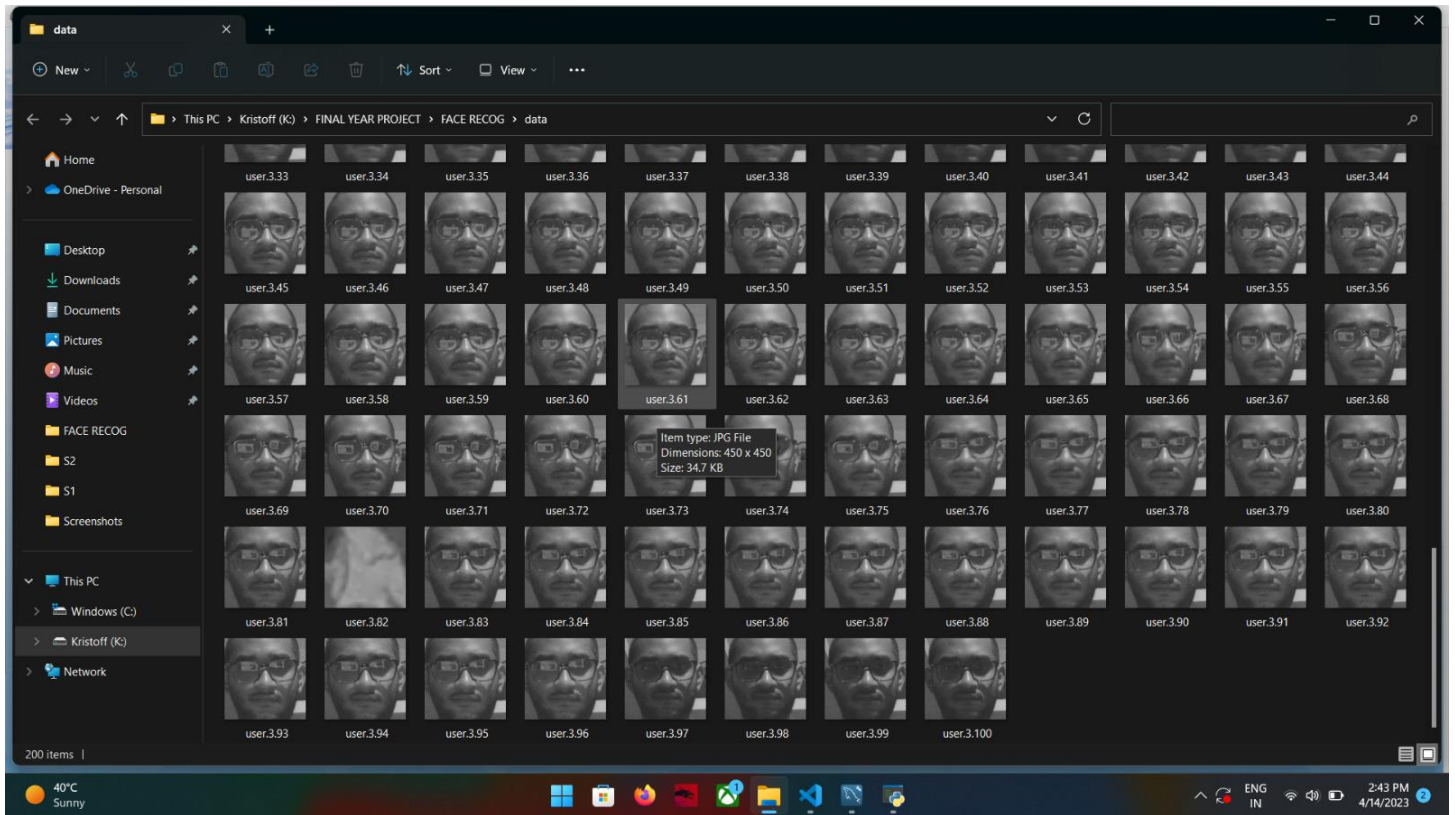
65



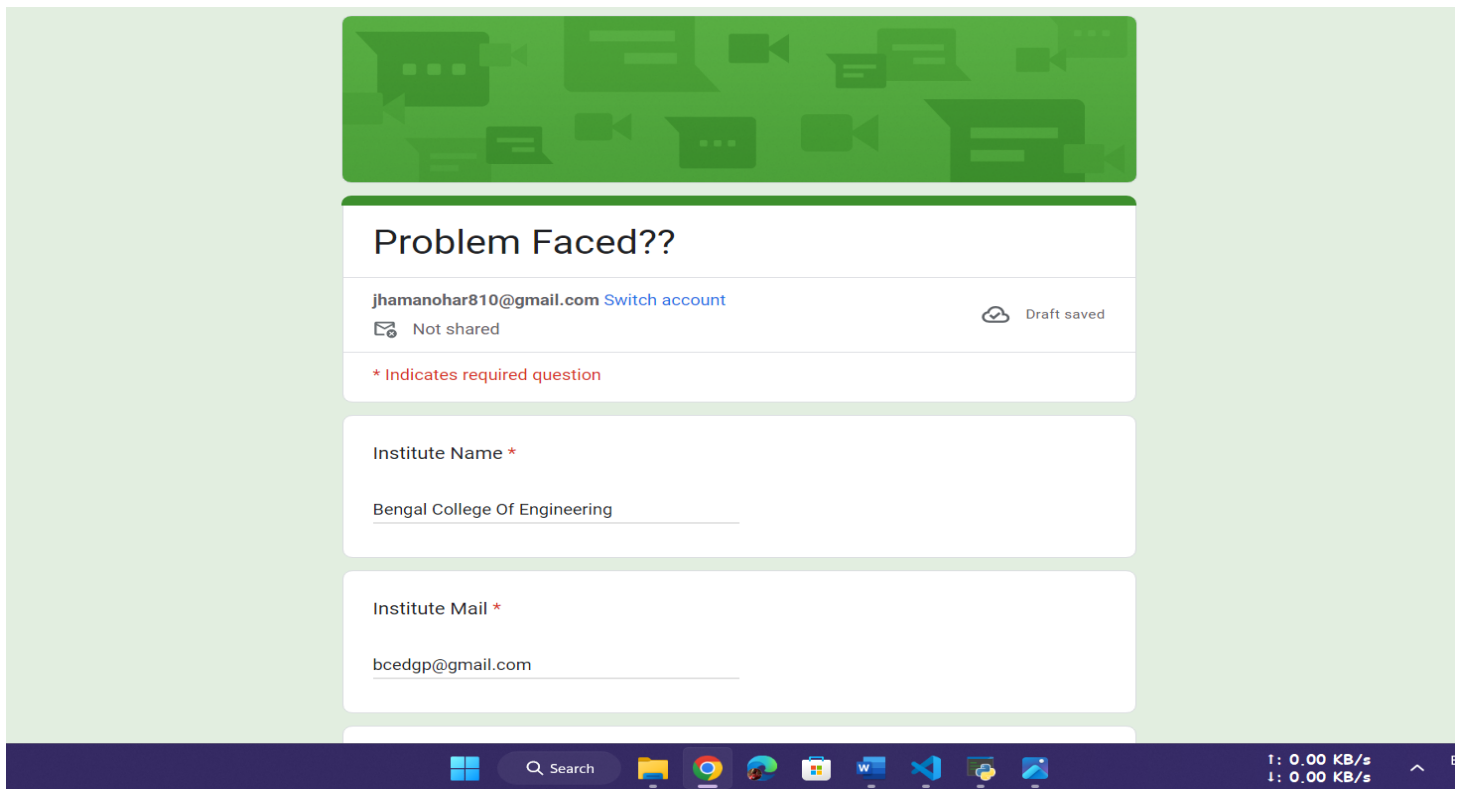
Face Recognition



Train Data



Here our system takes 100 sample photos that can be stored in a database.



The screenshot shows a web application interface with a green header featuring icons of speech bubbles and video cameras. Below the header is a form titled "Problem Faced??". The form includes a user profile section with the email "jhamanohar810@gmail.com", a "Switch account" link, a "Not shared" status, and a "Draft saved" indicator. A red asterisk note states "* Indicates required question". The form has two main input fields: "Institute Name *" with the value "Bengal College Of Engineering" and "Institute Mail *" with the value "bcdg@gmail.com". The Windows taskbar is visible at the bottom, showing the Start button, Search bar, and various application icons.

Problem Faced??

jhamanohar810@gmail.com [Switch account](#)

Not shared Draft saved

* Indicates required question

Institute Name *

Bengal College Of Engineering

Institute Mail *

bcdg@gmail.com

Help Desk/Support

11. Application of Facial Recognition Attendance System

Facial recognition technology is one of the emerging innovative technologies that help companies, organizations, and government agencies to improve their business productivity. It helps in identifying and recognizing people.

- It is an analytical tool that identifies a person by matching their face with the stored image in the database. Then, an algorithm compares a person's facial features against the facial features of the stored images on the database. This technology has gained rapid acceptance globally among global brands.
- It is a fast, high-accuracy system that accurately identifies and recognizes faces for identity verification, customer identification, security, access control, and other uses.
- Technology is used in various sectors, including finance, retail, government, and industry. For instance, it can be used in automated teller machines, retail banking, airport check-in, customer identification, and credit/debit card payments.

Time Dynamo can be integrated into workplace collaboration applications like Troop Messenger, helping company users track and manage their attendance, leave requests, and other crucial information.

Advancements in face recognition system:

Facial recognition technology promises many advantages over existing biometric technologies, including reduced error rates and significantly less cost and training needs than those associated with fingerprinting. The benefits promised by the vendors include faster screening times with increased accuracy, better usability, speed, reliability, ease of implementation, improved security, convenience, etc.

Facial recognition technology is a very efficient technology. It is based on how closely you can match a photograph to a person in real time. Some face recognition software uses a 3D scan of the face or retinal scan to identify the person.

12. FUTURE SCOPE

The face recognition model would be done more precisely so that maximum accuracy can be achieved. The marked attendance is stored in only one excel, rather different excels for different date can be created. More Tweaks to be done and add extra features like a biometric finger print scanning System.

Facial Recognition Dynamics:

The future scope of facial recognition attendance systems is promising, with several potential advancements and applications on the horizon. Some key areas of future development include:

Improved Accuracy and Performance: Continued advancements in facial recognition algorithms and technology will likely lead to even higher accuracy rates and faster processing times. This can result in more reliable attendance tracking and reduced false positives or negatives.

Enhanced Security Features: Future systems may incorporate additional security features such as liveness detection to ensure that the detected face is from a live person and not a photograph or video. This can further enhance the system's resistance to spoofing or fraudulent attempts.

Integration with Other Systems: Facial recognition attendance systems can be integrated with other existing systems, such as access control systems, time and attendance management software, and HR databases. This integration allows for seamless data exchange, providing a more comprehensive solution for organizations.

Mobile and Cloud-based Solutions: The development of mobile facial recognition applications can enable employees or students to check-in or check-out using their smartphones, providing flexibility and convenience. Cloud-based solutions can also offer scalability, remote accessibility, and centralized management for multi-site organizations.

CONCLUSION

In this approach, a face recognition based automated attendance system is thoroughly described. The proposed approach provides a method to identify the individuals by comparing their input image obtained from image in folder. From this model we can recognize the faces of students and can mark their attendance automatically in real time without human intervention. The feasibility of the model can be increased if a cloud can be hired to store details.

Limitation of our Work

- The working of this project would become a tedious task for the working on identical twins or we can say that the proposed idea will not work for two identical twins.
- In the system, scanning of genuine person is done using camera, so sometimes it may take large amount of time for configuring the genuine identity due to the lack of server issue or the failure of the database.
- Unable to recognize face with different angles, image quality, size, and light intensity.

REFERENCES

- [1] M. Karunakar, C. A. Sai, K. Chandra, and K. A. Kumar, “Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment,” International Journal for Recent Developments in Science and Technology, vol. 4, no. 5, pp. 194-201, 2020.
- [2] S. Bhattacharya, G. S. Nainala, P. Das and A. Routray, “Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment,” in 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), 2018.
- [3] GitHub Link: - <https://github.com/NANDALALGORAI/Face-Recognition-System>